

UNIVERSITY OF MINES AND TECHNOLOGY

TARKWA

**FACULTY OF COMPUTING AND MATHEMATICAL
SCIENCES**

DEPARTMENT OF MATHEMATICAL SCIENCES

A PhD THESIS ENTITLED

**A HYBRID MODEL FOR RECURRENT HEAD AND NECK
SQUAMOUS CELL CARCINOMA PROGNOSIS IN GHANA**

SUBMITTED IN FULFILMENT OF THE REQUIREMENT FOR THE
AWARD OF THE DEGREE OF DOCTOR OF PHILOSOPHY IN
MATHEMATICS

BY

DAMIANUS KOFI OWUSU

TARKWA, GHANA
OCTOBER 2023

UNIVERSITY OF MINES AND TECHNOLOGY

TARKWA

**FACULTY OF COMPUTING AND MATHEMATICAL
SCIENCES**

DEPARTMENT OF MATHEMATICAL SCIENCES

A PhD THESIS ENTITLED

**A HYBRID MODEL FOR RECURRENT HEAD AND NECK
SQUAMOUS CELL CARCINOMA PROGNOSIS IN GHANA**

BY

DAMIANUS KOFI OWUSU

**SUBMITTED IN FULFILMENT OF THE REQUIREMENT FOR THE
AWARD OF THE DEGREE OF DOCTOR OF PHILOSOPHY IN
MATHEMATICS**

THESIS SUPERVISORS

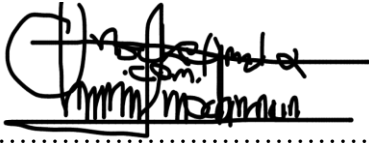
.....
ASSOC PROF CHRISTIANA CYNTHIA NYARKO

.....
DR JOSEPH ACQUAH

**TARKWA, GHANA
OCTOBER 2023**

DECLARATION

I declare that this thesis work is my own. It is being submitted for the degree of Doctor of Philosophy in Mathematics in the University of Mines and Technology (UMaT), Tarkwa. It has not been submitted for any degree or examination in any other University.



.....

(Signature of candidate)

.....day of(year)



ABSTRACT

Despite the rapid advancement in the development of hybrid ensemble Machine Learning (ML) techniques in malignancy management, recurrence and mortality from Head and Neck Squamous Cell Carcinoma (HNSCC) subtypes have not significantly improved in recent decades due to poor prognosis. Moreover, the recurrent HNSCC prognoses increase in patients with HNSCC due to the metastatic stage of the tumor at diagnosis, but studies providing promising prognostic models as a supporting tool for recurrence classification and prediction in HNSCC are lacking. As a supporting tool for identifying the most accurate prognosis and a robust prognostic classification model for classifying HNSCC recurrence patterns, this study presents a hybrid stacked ensemble classifier model when the same ML classifiers for; feature selectors, base classifiers, and meta classifiers are used, that could accurately predict recurrence outcomes and identify the most newly accurate prognostic features in HNSCC recurrence. Retrospective data of 125 HNSCC patients treated with curative intent between 2016 and 2020 at KBTH and who had a follow-up within this calendar period are collected. Data is pre-processed using mode imputation and one-hot encoding. The proposed Hybrid Ensemble Super Classification Algorithm (HESCA) model uses the ML classifier models including Gradient Boosting Machine (GBM), Distributed Random Forest (DRF), Deep Neural Network (DNN), Generalised Linear Model (GLM), and Naïve Bayes (NB) for stacked ensemble learning. These classifier models are employed in constructing feature subsets, base classifiers, and with each as a meta-classifier in a stacking ensemble. The performances of the HESCA model on various feature subsets are compared. Next, the performance of the HESCA model on 8-input features is compared with the HESCA model on full-input features. Then, based on gradient-boosted features, the performance of the HESCA model is compared with the established stacked ensembles. Thus, the two baseline stacked ensemble models, and one state-of-the-art stacked ensemble model. The results show that when the GBM classifier is used as a meta-classifier in a stacking ensemble consisting of five base classifiers on gradient-boosted features (GBM-input features) including concurrent chemoradiotherapy treatment, age at diagnosis, *p63*, cervical lymph/neck nodes, tumor size, smoking habit, pathological tumor staging at T4, and stage IV of tumor at diagnosis achieves higher accuracy (90.63%) with the least log loss (0.2959) compared to that achieved by base models and the established stacked ensemble models on the same gradient boosted features of recurrent HNSCC prognostic data. This gives a hybrid stacked ensemble model termed the HESCA model, which consists of five base models under study

and a GBM meta-model. It is also observed that this HESCA model on GBM-input features achieves better classification evaluation measures than that achieved on any other input feature subsets as well as the full-input feature subset considered in this study. The study shows that using the GBM classifier as a meta-classifier model in a stacking ensemble having five base classifiers with its gradient-boosted features results in better performance than base models and any other established stacked ensemble model used in this study; and using the HESCA model with gradient boosted features is clinically appropriate as a supporting tool for identifying, classifying and predicting patients' recurrent HNSCC prognostic data.



DEDICATION

I dedicate this thesis to my loving father and mother

Mr Andrews Owusu and Mrs Hilaria Kofi



ACKNOWLEDGEMENTS

I wish to gratefully acknowledge the Almighty God who by His grace and mercies granted me with wisdom, knowledge, and strength to undertake this academic thesis. I again express my sincere gratitude to my lovely family, the Owusu family, for their immense contributions to my formal education and general upbringing.

I would like to express my deepest appreciation and gratitude to my supervisors, Christiana Cynthia Nyarko (Assoc. Prof), Joseph Acquah (PhD) and Joel Yarney (MD), for their invaluable guidance, assistance and efficient supervision and timely intervention during the course of preparing this thesis.

Special thanks to Dr Joel Yarney, the Head of National Centre for Radiotherapy and Nuclear Medicine, Department of Radiation and Oncology at Korle Bu Teaching Hospital (KBTH), Accra, Ghana, for his support and contributions towards the data collection and the accomplishment of this thesis.

My appreciation also goes to the lecturers in the Department of Mathematical Sciences and all the staff of the University.

Finally, I wish to record my indebtedness and appreciation to everyone that has been so helpful and supportive in this thesis and brought it to success.

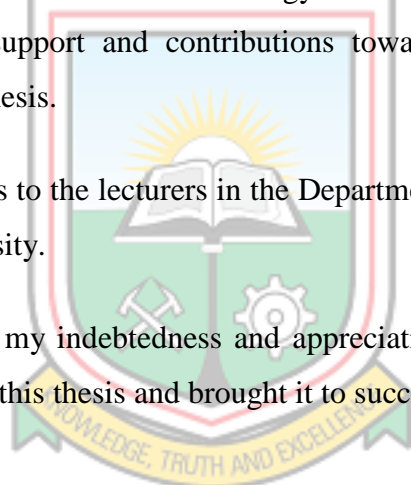
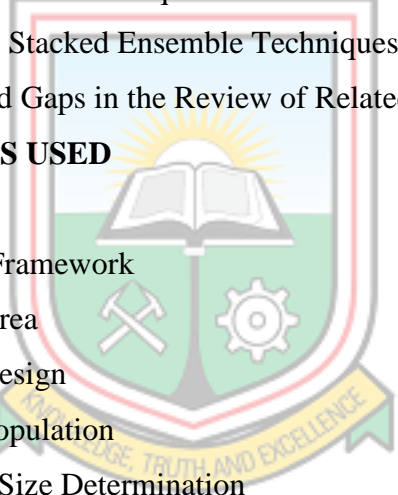


TABLE OF CONTENTS

Content	Page
DECLARATION	i
ABSTRACT	ii
DEDICATION	iv
ACKNOWLEDGEMENTS	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	x
LIST OF TABLES	xii
CHAPTER 1 INTRODUCTION	1
1.1 Background of the Study	1
1.2 Statement of the Problem	4
1.3 Aim of Study	6
1.4 Research Objectives	6
1.5 Methods Used	6
1.6 Facilities and Resources used for the Study	7
1.7 Scope of the Study	7
1.8 Organisation of the Thesis	7
CHAPTER 2 REVIEW OF RELEVANT LITERATURE	8
2.1 Overview	8
2.2 History of Cancer and Machine Learning	8
2.2.1 Early History of Cancer	8
2.2.2 History of Machine Learning	8
2.3 Prediction and Prognosis of Cancer	9
2.4 Squamous Cell Carcinoma of Head and Neck Subtypes	10
2.4.1 Oral Cavity	11
2.4.2 Pharyngeal Cancer	11
2.4.3 Laryngeal Cancer	11
2.4.4 Paranasal Sinuses and Nasal Cavity	11
2.4.5 Salivary Glands	11
2.5 Head and Neck Squamous Cell Carcinoma Statistics	12
2.6 Risk Factors of Head and Neck Squamous Cell Carcinoma	16
2.6.1 Gender and Age	17

2.6.2	Smoking, Tobacco and Alcohol	17
2.6.3	Virus Infection	17
2.6.4	Genes Mutation	18
2.7	Clinical, Pathological and Genomic Markers	18
2.7.1	Clinical Markers of HNSCC	19
2.7.2	Histopathological (Pathological) Markers of HNSCC	19
2.7.3	Genomic Markers of HNSCC	19
2.8	Management of Head and Neck Cancers	19
2.8.1	Diagnosis	20
2.8.2	Treatment	20
2.8.3	Prognosis	22
2.8.4	Follow Up/Recurrence	22
2.9	Works Related to ML Techniques in Recurrent HNSCC Prognosis	22
2.10	Works Related to Stacked Ensemble Techniques in Cancer Study	25
2.10.1	Identified Gaps in the Review of Related Literature	31
CHAPTER 3 METHODS USED		32
3.1	Overview	32
3.2	Methodological Framework	32
2.2.1	Study Area	32
2.2.2	Study Design	33
2.2.3	Study Population	33
3.2.4	Sample Size Determination	34
3.2.5	Data Collection Instruments	35
3.2.6	Data Collection Procedures	35
3.2.7	Data Handling and Quality Control	36
3.2.8	Data Pre-processing	37
3.3	Theoretical Background of Classifiers	41
3.3.1	Ensemble Learning	42
3.3.2	Base Learners (Classifiers)	45
3.4	Feature Selection Stanzas Ensemble Feature Selection	57
3.4.1	The Feature Selection Technique	57
3.5	V-Fold Cross-Validation Technique	58
3.6	Detecting Multicollinearity using Variance Inflation Factor	59



3.7	Good Fit Learning Curves	59
3.8	Model Evaluation Measurements, Validation and Comparison	59
3.8.1	Binary Cross-Entropy/Logarithmic Loss	61
CHAPTER 4	DEVELOPMENT OF HYBRID RECURRENT HEAD AND NECK SQUAMOUS CELL CARCINOMA PROGNOSTIC MODEL	63
4.1	Overview	63
4.2	Development of Recurrent HNSCC Prognostic (HESCA) Model	63
4.2.1	Proposed Ensemble Feature Selection Technique	63
4.2.2	The Proposed Stacked Ensemble Classification Model for HESCA	65
CHAPTER 5	RESULTS AND DISCUSSIONS	79
5.1	Overview	79
5.2	Multicollinearity check using Variance Inflation Factor Technique	80
5.3	Feature Selection Techniques	82
5.4	HESCA Model	85
5.4.1	HESCA Model Training on Training Data Set	88
5.4.2	HESCA Model Evaluation on Test Set	92
5.5	Baseline Stacked Ensemble Classification Techniques	98
5.5.1	Stacked Ensemble Model (GBM & DRF) with GLM Meta-Classifier	99
5.5.2	Stacked Ensemble Model (GBM, DRF & DNN) with GLM Meta-Classifier	100
5.5.3	State-of-the-Art (SA) Stacked Ensemble Model	102
5.6	Comparative Analysis of the Results	105
5.7	HESCA Classification Model Prediction	112
5.7.1	Partial Dependence Plot and Individual Conditional Expectations	115
CHAPTER 6	CONCLUSIONS AND RECOMMENDATIONS	121
6.1	Conclusion	121
6.2	Research Contributions to Knowledge	122
6.3	Recommendations	124
6.4	Limitation of the Study	125
REFERENCES		126

APPENDICES	138
APPENDIX A GRAPHS FOR PROGNOSTIC FEATURES R CODE	138
APPENDIX B DATA CLEANSING R CODE	149
APPENDIX C BASE MODELS SELECTION R CODE	151
APPENDIX D MODELS LEARNING R CODE	152
APPENDIX E GRAPHS OF MODELS PERFORMOMANCE R CODE	163
APPENDIX F PDP AND ICE R CODES	169
APPENDIX G HNSCC PROGNOSIS DATASET	172
APPENDIX H MEAN ACCURACY FOR BASE MODELS SELECTION	176
APPENDIX I FRAMEWORK OF SAMPLE SIZE DETERMINATION	177
APPENDIX J LIST OF ACRONYMS AND ABBREVIATIONS	178
APPENDIX K LIST OF PUBLICATIONS	181
APPENDIX L ETHICAL ISSUES	184
APPENDIX M INDEX	185
APPENDIX N SIMILARITY INDEX	194



LIST OF FIGURES

Figure	Title	Page
2.1	Head and Neck Cancer Regions	12
2.2	Ten Most Frequent Cancers in Ghana 2012; Both Sexes, All Ages	14
2.3	Ten Most Frequent Cancers in Ghana 2018; Both Sexes, All Ages	14
2.4	Cancer Cases Recorded Among Ghanaian Males	15
2.5	Cancer Cases Recorded Among Ghanaian Females	15
2.6	Estimated Number of HNCs Incidences in Ghana from 2018 to 2040	16
2.7	Estimated Number of HNCs Deaths in Ghana from 2018 to 2040	16
3.1	Study Location	33
3.2	Graphs for Prognostic Features	39
3.3	Plot of Linearity or Nonlinearity of the HNSCC Dataset	41
3.4	Biologically inspired Neural Network (Karparthy, 2016)	53
3.5	An ANN Model Architecture	53
3.6	A Multi-layer Perceptron Feedforward Model	55
3.7	Algorithm for V-Fold Cross-Validation	58
4.1	Architecture of Feature Selection Techniques	65
4.2	Architecture of HESCA Model for Recurrent HNSCC Prognosis	74
4.3	Architecture of HESCA Model with Full-Input Features	78
5.1	Variance Inflation Factor Plot for Multicollinearity	81
5.2	Boxplots for Features	82
5.3	Ranks of Features	83
5.4	Plot of Performance of Feature Selection Techniques on Train Sets	87
5.5	Plot of Performance of Feature Selection Techniques on Test Sets	87
5.6	ROC Curve Analysis of Base Classifiers on Training Set	89
5.7	ROC Curve Analysis of Meta Classifiers on Training Set	92
5.8	ROC Curve Analysis of Base Classifiers on Test set	95
5.9	ROC Curve Analysis of Meta Classifiers on Test set	98
5.10	Graph of Base Models versus HESCA Model on Training Set	107
5.11	Plot of Base Models compared with HESCA Model on Test Set	108
5.12	Graph of Stacked Ensemble Models compared with HESCA Model on Training Set	112

5.13	Graph of Stacked Ensemble Models compared with HESCA Model on Test Set	112
5.14	A Plot of Good Fit Learning Curves	114
5.15	Individual Conditional Expectations on Feature Nodes	115
5.16	Individual Conditional Expectations on Feature Age	116
5.17	Individual Conditional Expectations on Feature Smoke	116
5.18	Individual Conditional Expectations on Feature StageIV	117
5.19	Individual Conditional Expectations on p63 Feature	117
5.20	Individual Conditional Expectations on TreatCCRT Feature	118
5.21	Individual Conditional Expectations on Feature PaTT4	118
5.22	Individual Conditional Expectations on Feature Size	119



LIST OF TABLES

Table	Title	Page
2.1	Summary of Stacked Ensemble Techniques in the Cancer (Subtypes) Studies	27
2.2	Summary of ML Techniques in the Prognosis for Recurrent HNSCC (Subtypes)	28
3.1	Demographic, Clinical, Pathological and Genomic Features	35
3.2	Description of Features for 125 Instances	38
3.3	Stacking Algorithm	45
3.4	Random Forest Algorithm	46
3.5	Gradient Boosting Algorithm	47
3.6	Confusion Matrix for Prognosis HNSCC Recurrence	60
4.1	Baseline Stacked Ensemble Algorithm with V-fold Cross Validation	67
4.2	State-of-the-Art Stacked Ensemble Algorithm with V-fold Cross Validation	68
4.3	Proposed Hybrid Ensemble Super Classification Algorithm for HESCA	72
4.4	The HESCA Model Hyperparameters for Recurrent HNSCC Prognosis	76
5.1	Variance Inflation Factor (VIF) of Features	80
5.2	Performance Metrics of HESCA Model with Full-Input Features	81
5.3	Top 20 Most Important Features Selection	84
5.4	Optimal Feature Subsets by various Feature Selection Techniques	85
5.5	Performance Metrics of HESCA Model on various Feature Subsets	86
5.6	Performance of Base Classifiers on Training Set based on GBM-FS Optimal Feature Subset	88
5.7	Performance Metrics of Base Classifiers on 10-Fold Cross-Validation Set	90
5.8	Performance Metrics of Meta Classifiers on Level-one Training Set	91
5.9	Performance Matrix of Base Classifiers on Test Set	93
5.10	Performance Metrics of Base Classifiers on Test Set	94
5.11	Performance Matrix of Meta Classifiers on Test Set	96
5.12	Performance Metrics of Meta Classifiers on Test Set	97
5.13	Classification Matrix for Stacked Ensemble-GLM1 on Training and Test Sets: Stack GBM and DRF using GLM	99
5.14	Performance Metrics of Stacked Ensemble-GLM1 on Training and	

	Test Set: Stack GBM and DRF using GLM	100
5.15	Classification Matrix for Stacked Ensemble-GLM2 on Training and Test Sets: Stack GBM, DRF and DNN using GLM	101
5.16	Performance Metrics of Stacked Ensemble-GLM2 on Training and Test Sets: Stack GBM, DRF and DNN using GLM	101
5.17	Classification Matrix for State-of-the-Art (SA) Stacked Ensemble Model on Test Set	102
5.18	Performance Metrics of State-of-the-Art (SA) Stacked Ensemble Model on Training Set	103
5.19	Performance Metrics of State-of-the-Art (SA) Stacked Ensemble Model on Test Set	104
5.20	Comparison of HESCA model with full input features and HESCA model with GBM-FS features	105
5.21	Comparison of Base Models and HESCA Model Performance on Training Data based on GBM-FS	106
5.22	Comparison of Base Models and HESCA Model Performance on Test Data based on GBM-FS	107
5.23	Comparison of Baseline Stacked Ensemble Models and HESCA Model Performance on Training and Test Sets	108
5.24	Comparison of State-of-the-art (SA) Stacked Ensemble Model and HESCA Model Performance on Training and Test Sets	110
5.25	Summary of the comparison of Baseline Stacked Ensemble Models and State-of-the-Art Model with HESCA Model on Test Set	111
5.26	HESCA Classification Model Prediction	113

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

The problem of Head and Neck Squamous Cell Carcinoma (HNSCC) and its associated relapse is continuously increasing for patients with locally metastatic stage tumors and for years now, has globally increased medical attention; particularly in the developing countries without the exclusion of Ghana. Increased knowledge in ensemble Machine Learning (ML) techniques that are predisposed to provide the most protuberant prognostic features that are associated with the progressions and treatments of cancer under study, the prognosis deemed most accurate for early primary cancer detection and treatment outcomes can be generated to improve the recurrence patterns of locally advanced stage patients that minimises HNSCC recurrences.

On the one hand, machine learning (ML) is a subfield of artificial intelligence (AI) that enables algorithms to automatically learn from a prior set of training data and improve upon it by utilising statistical, probabilistic, and optimisation tools that categorise new data, identify novel patterns, and/or prognosticate novel trends without being explicitly programmed. Mitchell (2006). Ensemble ML is a technique that combines multiple of either homogeneous or heterogeneous base learners into a strong learner. ML techniques are becoming versatile as an alternate approach to conventional statistical methods in medical diagnosis and prognosis as the algorithms can handle noisy and incomplete data and despite the small sample size, significant results can be obtained. The vital goal of ML techniques in modelling cancer prognosis focuses on producing a well optimally prognostic model for classification, prediction, estimation, and/or related tasks for the progression and treatment of cancer subtypes. ML techniques can be divided into two main categories. According to Mitchell (1997) and Duda *et al.* (2001): supervised learning, where a set of training data is labeled to produce a function that maps input data to the desired output, and unsupervised learning, where a set of examples is given but no labels are provided and the notion of output during the learning process is unknown. Surprisingly, the majority of machine learning (ML) techniques used to predict and diagnose cancer typically involve supervised learning, including Deep Neural Networks (DNN) or Artificial Neural Networks (ANNs), Bayesian Neural Networks (BNNs), Naïve Bayes (NB), Gradient Boosting Machine (GBM), Support

Vector Machines (SVMs), Distributed Random Forest (DRF), Decision Trees (DTs), K-Nearest Neighbours (KNNs), and others.

On the other hand, a variety of conditions can change the Deoxyribonucleic acid (DNA) in cells, which ultimately leads to the sickness known as cancer. Oncogene activation and tumor suppressor gene inactivation can cause unchecked cell division and rapid aberrant cell development, resulting in the formation of a mass of tissue known as a tumor. According to Anon (2016), this tumor may be benign (non-cancerous) or malignant (cancerous). According to this, cancer is a heterogeneous group of connected diseases rather than a single illness, with hereditary and environmental factors both contributing to the development and progression of the disease. Head and Neck Cancer (HNC) develops from functional areas such as the pharynx, mouth cavity, and larynx, among others (Stewart and Wild, 2016; Argiris *et al.*, 2008; Boyle and Levin, 2008; Jemal *et al.*, 2007). Squamous cell carcinomas make up about 90% of these HNCs. The term Squamous Cell Carcinoma (SCC) refers to cancer that develops from squamous cells. These squamous cells are moist tissues that line bodily cavities and are present in both the mucous membranes and the epidermis of the skin. It turns out that HNC, which develops from these squamous cells, is simply referred to as Head and Neck Squamous Cell Carcinoma (HNSCC) or Squamous Cell Carcinoma of Head and Neck (SCCHN). In a nutshell, SCC prognosis is simply the estimate of the likely course and outcome of the disease; the chance of recovery or recurrence (Anon, 2020). Cancer susceptibility prediction (the likelihood of developing a cancer type before the disease occurs), cancer recurrence prediction (the likelihood of redeveloping a cancer type following the complete remission of the disease), and cancer survivability prediction (the outcome; life expectancy, survivability, progression, and tumor-drug sensitivity following cancer) are the three main predictions that Cruz and Wishart (2006) state are the focus of cancer prediction and prognosis.

Specifically, when the primary tumor is successfully treated and reaches its state of remission; a patient is well thought-out cancer-free, such a patient in remission state turns out to have about a 25%-48% probability of cancer recurrence as a result of the metastatic stage at diagnosis of the primary tumor (Mucke *et al.*, 2009). According to Worsham (2011), HNSCC recurrences are strongly linked to the tumor stage. Tumors staged at I and II have about 60-95% probability of being treated successfully unlike those with advanced stages (III and IV) of tumor at diagnosis. Nonetheless, World Health Organisation (WHO)

anticipates a future wide-reaching increase in HNSCC recurrence as a result of poor prognosis.

In Ghana, Commeh (2019), in an interview; specified and inveterate that, “*cancer cases in Ghana are not decreasing including that of HNSCC.*” She indicated that, 16 000 cancer cases were registered at the Korle-Bu Teaching Hospital (KBTH) and Komfo Anokye Teaching Hospital (KATH) between the years 2016 and 2018, and it is projected to increase to around 22 000. It was further indicated that more than 90 percent of cancer subtypes diagnosed at cancer centres are at the stage of metastasis because patients wait to see the cancer symptoms before they are presented to the hospital. As a result of this, the recurrent rate is high and its associated death tends to be one of the highest among other deaths in the country and even worldwide. Yarney *et al.* (2017) observed HNC to be the third most common cancer diagnosed at KBTH, Accra, with nasopharyngeal cancer being the most common in this regard.

Cancer prognoses typically involve multiple surgeons from various specialties, using a variety of subsets of clinical and histopathological parameters, such as tumor type, size, grade, location of the malignant tumor, and metastatic lymph nodes. Risk factors including alcohol consumption, and smoking; staging from I to IV; and treatments including chemotherapy, surgery, radiotherapy or a combination of these treatments (Catto *et al.*, 2006; Reichart, 2001; Fielding *et al.*, 1992). Medically, these factors exclusively might not offer enough information on a patient to make robust prognostications. Colozza *et al.* (2005) had theoretically demonstrated that the combination of some specific molecular information, either about the tumor or the genetic markers, would yield enough information. Typically, in cancer prognosis research, socio-demographic information about the patient is combined with clinical data (patient-based), histopathological data (tissue-based), genomic or microarray data (molecular-based), or any combination of these. In the meantime, numerous research (Alabi *et al.*, 2019; Ribeiro *et al.*, 2017; Chang *et al.*, 2013) on HNSCC have been carried out in various domains, including both theoretical medicine and medical statistics, taking into account the integration of genetic, clinical, and pathological indicators.

Now, given the rapid development of these medical pieces of information (clinical, pathologic, and genomic) and the growing trend and reliance on the application of ML techniques, it is worthwhile that, if the genomic markers of patients are combined with clinicopathological information (Chang *et al.*, 2013; Exarchos *et al.*, 2011a) under some

variant studies such as ensemble Machine Learning (ML) techniques (Akinbohun, 2021; Akinbohun *et al.*, 2020; Adeyemi *et al.*, 2019), which are associated with the evolvments and treatments of the primary tumor, the prognosis deemed the most newly accurate that aid early detection and treatment results may be enhanced (Lavanya & Chandra, 2019; Colozza *et al.*, 2005; Mitchell, 2006). But according to Yaliang *et al.* (2015), if by ensemble learning, a robust prognostic model that generates the most accurate prognosis can be obtained for the improvement of treatment outcomes of cancer, then compared with the ensembles; bagging and boosting, stacking or stacked generalisation is the most effective and efficient technique. This is a technique that combines multiple different base learners into a strong learner in their combination using a meta-learning algorithm. This assertion is supported by, Ragunthar and Selvakumar (2019); Warsinske *et al.* (2019) that; indeed, this technique (stacked ensemble learning) has been observed to yield the most accurate outcomes in several studies for which it has been employed.

The study seeks to investigate how stacked ensemble learning technique of base classifiers can be employed in the prognosis of HNSCC recurrence by formulating a prognostic model termed a Hybrid Ensemble Super Classification Algorithm (HESCA) model that learns the recurrent HNSCC data based on genomic and clinicopathological markers.

1.2 Statement of the Problem

The HNSCC subtypes with their relapses globally pose clinical challenges to all clinicians, especially in the developing countries such as Ghana. The country Ghana records a high rate of recurrence and mortality, as a result of inaccurate identification of prognostic markers (Commeah, 2019; Bray *et al.*, 2018; Yarney *et al.*, 2017).

In cancer diagnosis and prognosis including HNSCC, traditional statistical methods (Log-rank test, Kaplan-Meier, Cox PH, etc.) are the most extensively used methods for feature selection and training of prognostic models. However, existing studies (Kourou *et al.*, 2015; Chang *et al.*, 2013; Cruz and Wishart, 2006) have shown that most of these methods are not suitable for complex and noisy cancer data. Thus, when used for a prognostic model where biological markers (genomic) are usually nonlinear, and some features are conditionally dependent, leads to model overfitting thereby yielding unstable prognostic results. As a result, ML techniques are currently the most extensively used technique in cancer studies. Meanwhile, existing studies (Kabir and Ludwig, 2019; Kwon *et al.*, 2019; Yaliang *et al.*,

2015) have shown that it is not that easy to obtain a single (standalone) classification model that has a good generalisation ability to be considered a robust classifier model, whereas stacking that combines different weak classifiers may well transform such classifiers into a robust one. According to Yaliang *et al.* (2015), analysis had theoretically and practically proven that the error expectation of stacking (heterogeneous ensemble) models is usually less compared to the error expectation of a single (or homogeneous ensemble) model; thereby yielding more accurate, reliable, and stable results.

To address the weakness of statistical and standalone ML models, previous studies have focused on stacked ensemble classification models in the prognosis of HNC subtypes; relating to HNC susceptibility, and HNC survivability. Akinbohun (2021); Akinbohun *et al.* (2020) in HNC susceptibility, and Chang *et al.* (2013); Chi-Chang *et al.* (2021) in HNC survivability. Information gathered from these studies shows that stacked ensemble ML techniques can produce more stable, unbiased, and reliable prognosis and prediction results at a higher level of accuracy compared to statistical and standalone ML techniques (Akinbohun, 2021; Chi-Chang *et al.*, 2021; Akinbohun *et al.*, 2020). There is currently no much exploration on the stacked ensemble models for recurrent HNSCC prognosis leaving a gap in recurrent HNC predictive foci to be filled (Chi-Chang *et al.*, 2021; Kwon *et al.*, 2019; Yarney *et al.*, 2017).

In Ghana, there are currently limited studies on any form of the implementation of stacked ensemble ML techniques on the combination of genomic and clinicopathologic makers for recurrent HNSCC prognosis that is prone to provide unbiased, reliable, and stable prognostic outcomes (Commeh, 2019; Yarney *et al.*, 2017). Therefore, due to this medical gap in the domain of HNC prediction, this study seeks to adapt baseline stacked ensemble ML techniques and state-of-the-art stacked ensemble technique used in breast cancer by Kabir and Ludwig (2019) and Kwon *et al.* (2019) respectively for recurrent HNSCC prognosis in Ghana to address the issue of poor and contradictory prognostic results produced by biased, unreliable, and unstable prognostic models that are in existence. This technique is believed to identify, classify and prognosticate the most stable, reliable, and accurate prognosis for recurrent HNSCC patients being the first study ever.

The study however differs from the previous studies in that, it falls in the domain of recurrent HNSCC prognosis, extending a stacked ensemble model having a maximum of four base classifiers to that having five base classifiers (Kabir and Ludwig, 2019) with (gradient

boosted features) GBM ensemble feature selection technique (Xu *et al.*, 2019), with the regularisation technique that improves the prognosis accuracy as well as the stability and generalisation ability of a classification model. The gradient boosting feature selection learns an ensemble of classification trees that can reliably extract relevant features, naturally discover nonlinear interactions between features and labels, scale linearly with the number of features and dimensions, and allow the incorporation of known sparsity structure (Xu *et al.*, 2019).

1.3 Aim of Study

The main aim of the study is to develop a hybrid stacked ensemble-based model on the combination of clinicopathologic and genomic optimal features for recurrent HNSCC prognosis that is prone to generate unbiased, reliable, and stable prognostic outcomes.

1.4 Research Objectives

The objectives of the study are to;

- i. identify the most accurate prognosis associated with recurrent HNSCC in Ghana.
- ii. develop a hybrid stacked ensemble classification model for recurrent HNSCC prognosis.
- iii. validate the developed model with existing data and compare it with three existing models in a stacking ensemble learning.
- iv. investigate that the prognosis for recurrent HNSCC is more robust when gradient-boosted features are used.

1.5 Methods Used

The methods employed in achieving research objectives include;

- i. GBM, DRF, DNN, GLM, and NB to develop a hybrid stacked ensemble classification model for recurrent HNSCC prognosis.
- ii. GBM ensemble feature selection to identify the optimum subset of prognostic features having significant main effects on the target feature.
- iii. V-fold Cross Validation (CV) technique to validate the developed model.
- iv. GBM as an ensemble feature selection technique to prove its robustness based on the first objective.

1.6 Facilities and Resources used for the Study

The facilities and resources used for the study include;

- i. Secondary data on patients with recurrent HNSCC at KATH, Kumasi, and KBTH, Accra.
- ii. Library and internet facilities at the University of Mines and Technology (UMaT), Tarkwa, and the University of Cape Coast (UCC), Cape Coast.
- iii. H2O package in R programming language. Technical advice from supervisors and experts in the field of the study.

1.7 Scope of the Study

The study is delimited to patients with recurrent HNSCC and nonrecurrent HNSCC only, that were diagnosed of any of the HNSCC subtypes specifically the laryngeal, hypopharyngeal, oropharyngeal, or nasopharyngeal cancer and were treated with curative intent only. Though there are several types of HNC cancer cases as well as their treatment intents. Also, the study focuses on the identification of the most accurate combination of prognostic markers, and the development, and validation of the developed hybrid stacked ensemble classification model for recurrent HNSCC prognosis based on the GBM feature selection technique using multiple ML techniques for comparative analysis.

1.8 Organisation of the Thesis

The thesis is organised into six (6) chapters as follows. Chapter 1 presents the background of the study, the statement of the problem, research objectives, methods used, facilities and resources to be used for the study, and the scope of the Study. Chapter 2 presents the general review of relevant literature pertinent to ML techniques, feature selection methods, and ensemble learning in recurrent HNSCC prognosis research, a review of HNSCC cases in Ghana, risk factors of HNSCC, and cancer management. Chapter 3 presents the methodological framework of the study. This chapter has two sections. The first section specifies the study area, research design, study population, sample size, data collection instrument and procedures, and data handling/control. And the second section presents the theoretical background of the base classifiers. Chapter 4 presents the mathematical formulation of a hybrid stacked ensemble classification model. Chapter 5 presents the data analysis and discussion of results. Chapter 6 presents the conclusions and recommendations of the study.

CHAPTER 2

REVIEW OF RELEVANT LITERATURE

2.1 Overview

This chapter discusses the general review of relevant literature pertinent to ML techniques and feature selection techniques in recurrent HNSCC prognosis research, a review of HNSCC cases in Ghana, risk prognostic factors of HNSCC, and cancer management.

2.2 History of Cancer and Machine Learning

2.2.1 Early History of Cancer

Throughout recorded history, both humans and other animals have experienced cancer. Ancient Egyptian mummies and fossilized bone tumors were where the disease's first signs of existence were found. Both the destruction of the bony skull of the head and neck and evocative growths of the malignant bone known as osteosarcoma (osteogenic sarcoma) have been observed in mummies. In a nutshell, cancer was discovered in Egypt in about 3 000 BC, which was called the "*Edwin Smith Papyrus*." The Greek physician Hippocrates, often referred to as the "Father of Medicine," lived from 460 to 470 BC and is credited with coining the term "cancer." Hippocrates used the terms carcinoma and carcinomas to denote both tumors that cause ulcers and those that do not. Because the finger-like progression of cancer reminded people of the shape of a crab, these two phrases are used in Greek to mean a crab. Well, the Roman physician Celsus (28–50 BC) changed the Latin word for cancer from the Greek word for crab. Also, the word *oncos* in Greek which means swelling (the growth nature of the malignant tumor) was used by another Greek physician, Galen (130–200 AD), to describe tumors. In as much as the crab analogy of Hippocrates and Celsus is still applied to describe malignant tumors, the term Galen is now mostly used to describe cancer-causing genes (as oncogenes), cancer management (as oncology), and as part of the name for cancer specialists (as oncologists).

2.2.2 History of Machine Learning

In a word, ML employs mathematical algorithms to learn from and analyze data in order to make future predictions and judgments. Today, ML algorithms let computers to interact with people, drive themselves, forecast natural disasters, and identify terrorist suspects. One of the terms that has been used a lot recently is ML. When pioneering computer scientist

Alan Turing wrote a paper in 1950 addressing the subject "Can Machines Think?" the idea of ML first came to mind. He put up the idea that machines with Artificial Intelligence (AI) might be able to persuade people that they are not actually machines. "Turing Test" was the name given to this.

The first computer learning software, which was the game of checkers, was created in 1952 by IBM's Arthur Samuel. The longer it played, the IBM computer improved the game by observing the plays that comprised winning tactics and adding such moves to its program.

Frank Rosenblatt created the Perceptron model, or the first neural network for computers, in 1957. The perceptron algorithm was created to activate the brain's classification abilities so that individuals may be divided into one of two groups based on the visual inputs they were given.

The "nearest neighbour" technique, created in 1967, later made it possible for computers to do fundamental pattern recognition. A robot called the Stanford Cart was created in 1979 by Stanford University students and can autonomously move around obstacles in a space. In 1981, Gerald Dejong established the idea of explanation-based learning, which uses data analysis and general rules to weed out irrelevant knowledge. The work of machine learning (ML) changed from a knowledge-driven method to a more data-driven approach during the 1990s (Baiju, 2019). Scientists started developing computer algorithms that could analyze massive amounts of data and make inferences or "learn" from the outcomes. Since the dawn of the twenty-first century, numerous academic disciplines, including the study of medicine, have dabbled in the use of inventive ML approaches. In other words, as data production increases, so does the computers' capacity to process and analyse it.

2.3 Prediction and Prognosis of Cancer

The prediction of cancer susceptibility, cancer recurrence, and cancer survivability are the three predictive foci that are the focus of cancer prognosis and prediction. Cancer susceptibility makes an effort to predict the likelihood that cancer will develop based on some risk factors. Recurrence of cancer refers to the attempt to predict whether a certain cancer type will return after an apparent or complete remission. After then, cancer survival prediction attempts to forecast the course of events, including life expectancy, survival, progression, and tumor-drug sensitivity following cancer diagnosis and therapy (Cruz and Wishart 2006). The accuracy of the diagnosis affects the prognosis of cancer survival and

recurrence (Hagerty *et al.*, 2005). Cancer recurrence can be classified by its location: local recurrence; when cancer occurs in the same place as the original cancer, regional recurrence; when cancer grows into lymph nodes (tissues) near original cancer, or distant or metastatic recurrence; when cancer spread to tissues or organs far from original cancer (Anon, 2016).

With the speedy growth of molecular-scale information (genomic, proteomic, and imaging technologies) about tumors or patients, this information can now be readily acquired. That, combining any of such molecular information with clinical and pathologic markers, a robust prognostic accuracy of cancer may be enhanced (Cruz and Wishart, 2006).

There is also a growing trend and reliance on the applications of ML techniques in medical studies to make prognostications. This is the fact that ML techniques perform well in several domains of research even where data has a small number of instances, for which the conventional statistics oppose but require a large number of instances to perform well for significant outcomes (Mitchell, 1997). Sampling a large amount of medical data is hard as it is costly and requires a longer time, and the samples when obtained are usually either noisy or incomplete. It is here that ML techniques are needed to still achieve more accurate diagnostic or prognostic results.

With ML introduction to medical research, there has been a design of several techniques that have been implemented in this field for cancer cases diagnostic or prognostic results. Meanwhile, these techniques are usually supervised learning including ANN, BNN, SVM, DT, DRF, and other hybrid methods. Most of the studies focused on the comparison between ML methods and statistical ones by proving how effective and efficient are their methods given specific data and this will further be discussed in Section 2.9. But this present study focuses only on a comparison between some selected supervised ML techniques for study and not with statistical ones,

2.4 Squamous Cell Carcinoma of Head and Neck Subtypes

Squamous cells, which are present in the epidermis of the skin and the mucous membranes of the nose, mouth, and throat, are the source of the cancer known as head and neck squamous cell carcinoma (HNSCC). According to its location or region, HNSCC can be categorised as follows:

2.4.1 Oral Cavity

The hard plate, which is the bony top of the mouth, the gingiva (gums), the buccal mucosa, the floor of the mouth beneath the tongue, and the small area of gum behind the wisdom teeth (retromolar trigone) are all places where cancer can develop (Anon, 2017).

2.4.2 Pharyngeal Cancer

if a pharyngeal (throat) cancer begins to grow. The pharynx is a short, hollow tube that extends from behind the nose to the esophagus, measuring little under 5 inches. Pharyngeal cancer describes cancer that develops in the pharynx. Based on the area of the pharynx that has developed, there are three different types of pharyngeal cancer: nasopharyngeal carcinoma (cancer that develops in the upper region of the pharynx, behind the nose); oropharyngeal carcinoma (cancer that develops in the middle region of the pharynx, which includes the base of the tongue, the soft palate [the back of the mouth], and the tonsils); and *hypopharyngeal carcinoma* (when cancer evolves in the hypopharynx, the lower region of the pharynx) (Anon, 2017).

2.4.3 Laryngeal Cancer

Laryngeal cancer refers to cancer that has spread to the larynx. The larynx, often known as the voice box, is a small neck tunnel formed by cartilage directly below the pharynx. The vocal cords are located in the larynx, along with the epiglottis, a little piece of tissue that moves to cover the larynx's food route in order to prevent food from entering the airways. Laryngeal carcinoma is a type of cancer that develops in the larynx (Anon, 2017).

2.4.4 Paranasal Sinuses and Nasal Cavity

Cancer can also emerge in the paranasal sinuses, the bones of the head containing small hollow spaces; and surrounding the nose, and this cancer is called sinus carcinoma. Inside the nose, there is a hollow space called the nasal cavity (Anon, 2017).

2.4.5 Salivary Glands

Cancer can also develop in the salivary glands, which produce saliva. Salivary glands are the on the floor of the mouth and close to the jawbone (Anon, 2017).

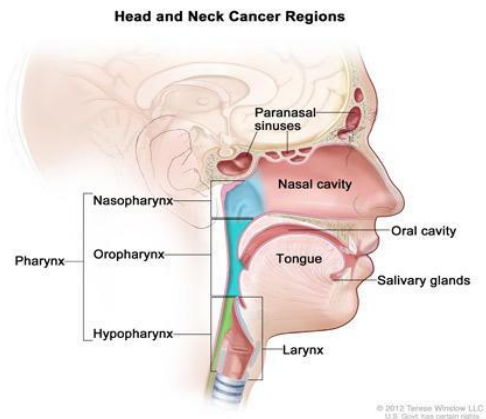


Figure 2.1 Head and Neck Cancer Regions

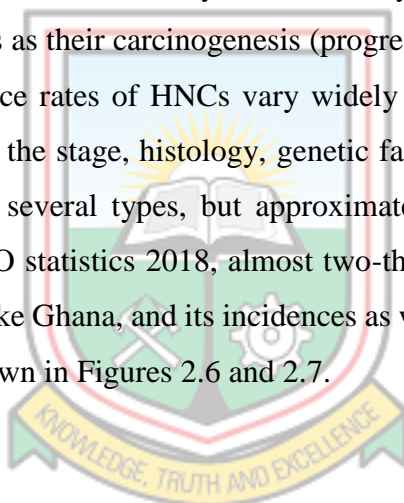
2.5 Head and Neck Squamous Cell Carcinoma Statistics

Based on the Ghanaian Cancer Statistics; National Strategy for Cancer Control in Ghana, HNC at the KBTH Oncology department in 2009 is listed as the third top most common leading cancer cases, with 14% of total cases in Ghana. Data from the Oncology Directorate of KATH and KBTH indicated that sinuses and laryngeal cancers were the commonest male cancers with 5.2% and 3.7% respectively of total cancers reported at KATH. According to Owusu-Afriyie *et al.* (2020), pharyngeal cancer (7.4%) and laryngeal cancer (3.5%) of total malignancies seen at the National Hospital respectively represent the 2nd and 7th most common cancers in Ghana. The estimated number of new HNC cases in Ghana based on GLOBOCAN data for the year 2012, was 889 (589 in males and 300 in females) and 901 in the year 2018 indicating that there had been an increase in the number of cases, as respectively shown in Figure 2.2 and Figure 2.3. In Ghana, males are more susceptible to HNCs as compared to their counterpart females as shown in Figure 2.4 and Figure 2.5. Even though HNC has not been viewed to be the foremost case of cancer in Ghana, its rate of mortality continues to rise as the number of incident cases increases. This is evident from the estimated number of incidences and deaths by GLOBOCAN, 2018 as respectively shown in Figure 2.6 and Figure 2.7.

There are over 40 000 and 800 000 new HNC cases reported annually in Africa and worldwide respectively. The rate of incidence of HNC subtypes differs from region to region. In Africa, based on GLOBOCAN, 2018 data, the annual age-standardized incidence rates per 100 00 for lip, and oral cavity in Western Africa was 1.1, in Middle Africa (1.5), North Africa (1.5), Eastern Africa (2.2), and Southern Africa (2.9). Also, Western Africa

(0.57), Middle Africa (0.88), North Africa (0.36), Eastern Africa (0.62), and Southern Africa (0.35) for salivary gland cancer. Again, Western Africa (0.25), Middle Africa (0.42), North Africa (0.20), Eastern Africa (0.34), and Southern Africa (0.79) for oropharyngeal cancer. More so, Western Africa (0.55), Middle Africa (0.75), North Africa (1.6), Eastern Africa (1.3), and Southern Africa (0.34) for nasopharyngeal cancer. Furthermore, Western Africa (0.12), Middle Africa (0.33), North Africa (0.32), Eastern Africa (0.28), and Southern Africa (0.29) for hypopharyngeal cancer.

HNC subtypes have percentages of about 80 to 90 of complete remission if their diagnoses are at the early stage (Foundation, 2010). HNCs have 17% and 22% locoregional and distant recurrence respectively when diagnosed at stage IV (Brockstein *et al.*, 2004). This cancer simply has a high recurrent mortality rate not that its diagnosis is impossible, but because the development of the disease is normally detected lately when it is in its advanced stage. HNCs are most dangerous as their carcinogenesis (progression) may not inform the patient earlier. As such, recurrence rates of HNCs vary widely between individuals, and within cancer types according to the stage, histology, genetic factors, patient-related factors, and treatments. HNCs are of several types, but approximately 90% are SCCs (Foundation, 2010). According to WHO statistics 2018, almost two-thirds of HNCs are experienced in underdeveloped nations like Ghana, and its incidences as well as deaths are expected to rise in the next decades as shown in Figures 2.6 and 2.7.



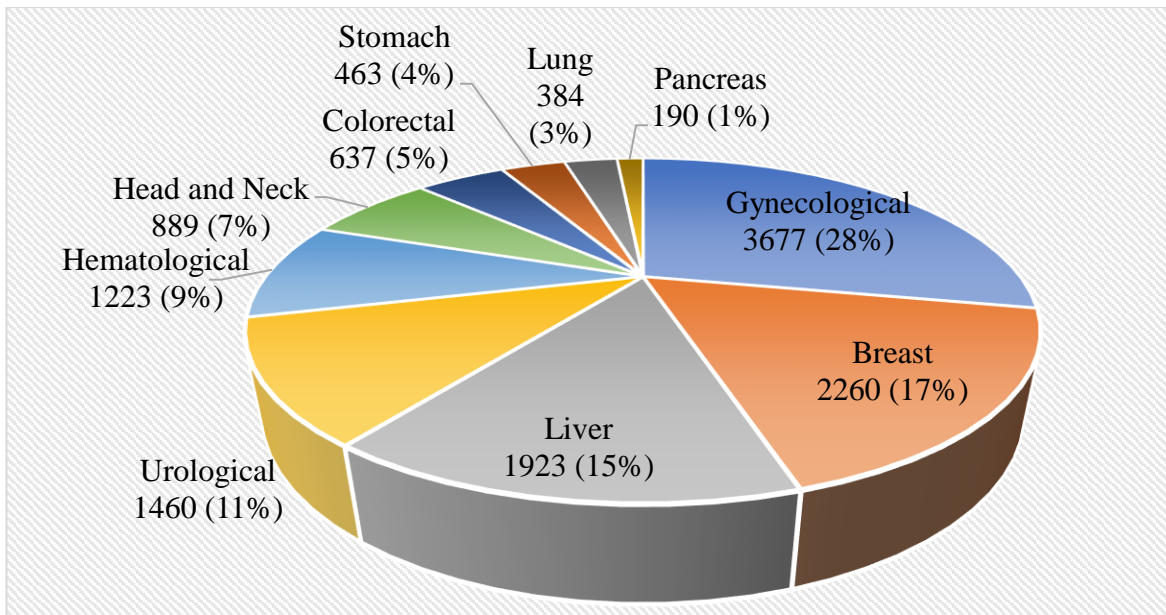


Figure 2.2 Ten Most Frequent Cancers in Ghana 2012; Both Sexes, All Ages
 Source: GLOBOCAN (2012)

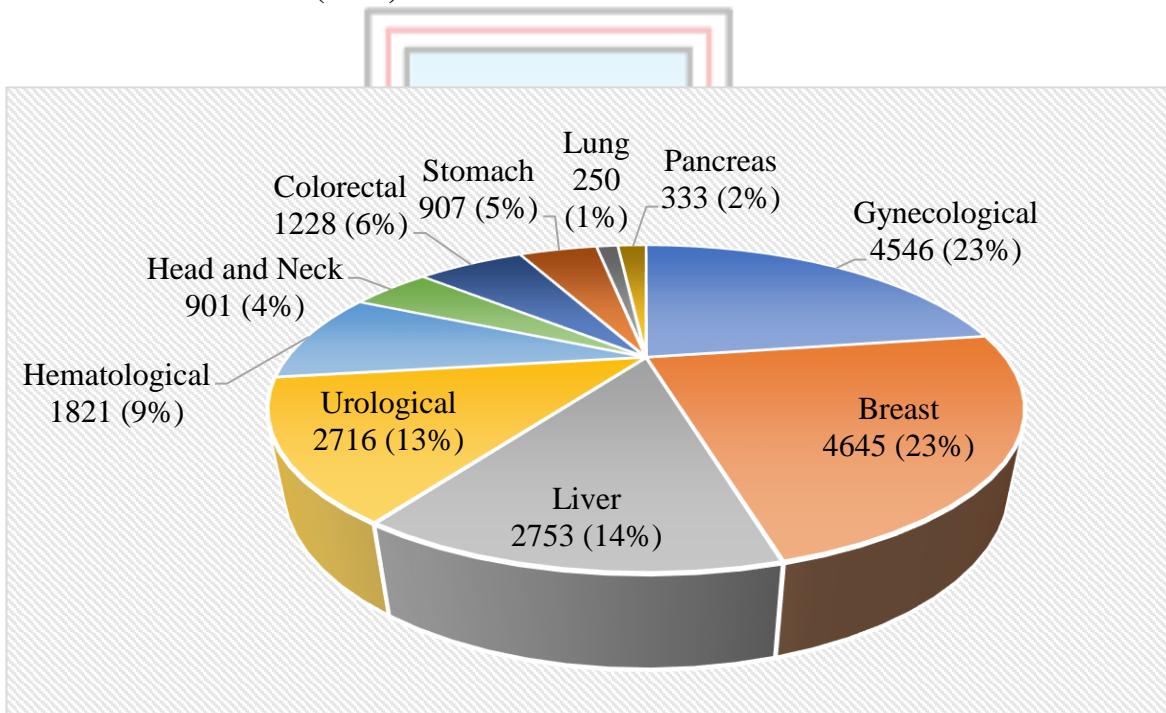


Figure 2.3 Ten Most Frequent Cancers in Ghana 2018; Both Sexes, All Ages
 Source: GLOBOCAN (2018)

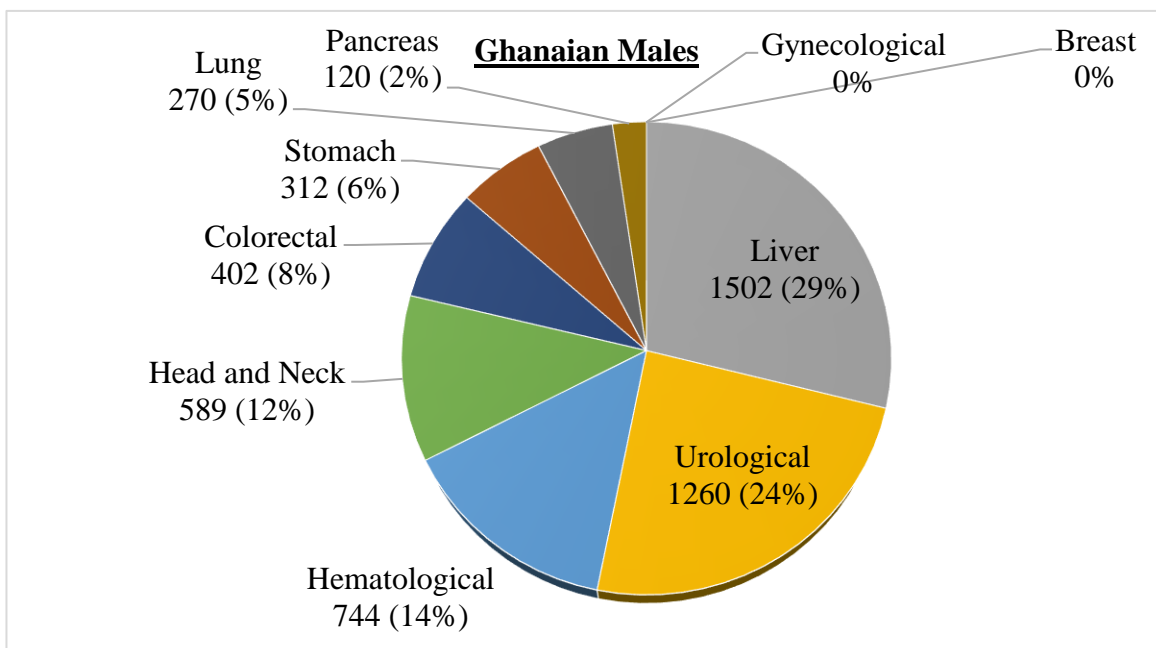


Figure 2.4 Cancer Cases Recorded Among Ghanaian Males
 Source: GLOBOCAN (2012)

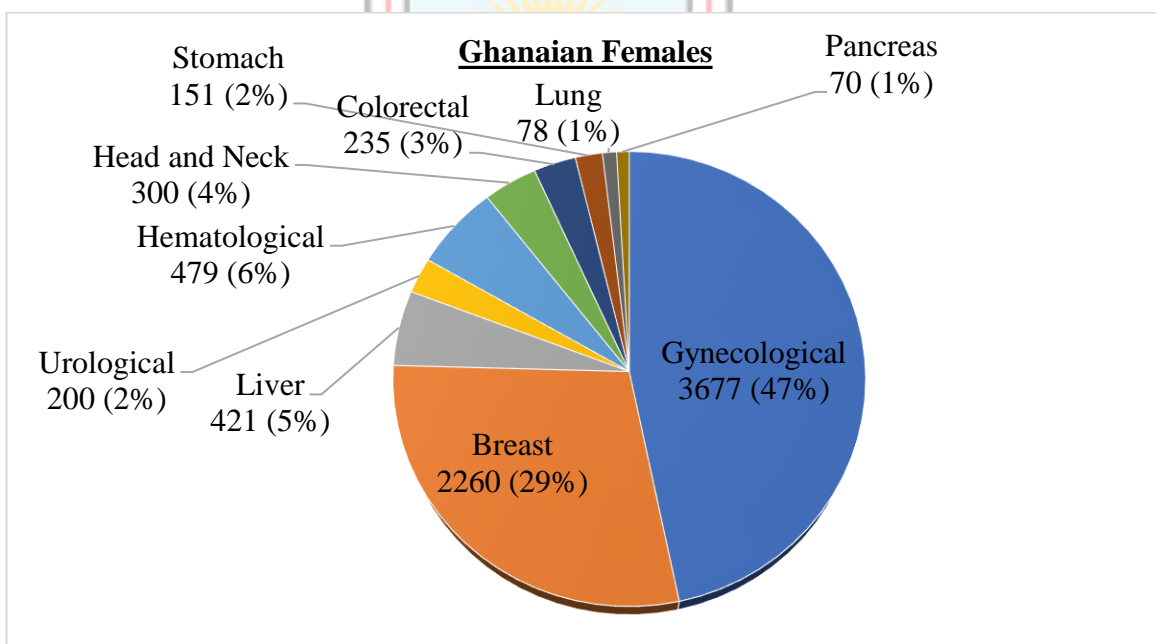


Figure 2.5 Cancer Cases Recorded Among Ghanaian Females
 Source: GLOBOCAN (2012)

***Figures 2.2-2.5** Gynecological (*cervix uteri, corpus uteri, vulva, vaginal, and ovarian*); Urological (*bladder, kidney, prostate, testes, and testis*), Hematological malignancies (*Hodgkin lymphoma, non-Hodgkin lymphoma, multiple myeloma, and leukemia*), and Head and Neck (*lip, oral cavity, larynx, salivary glands, nasopharynx, oropharynx, hypopharynx*).

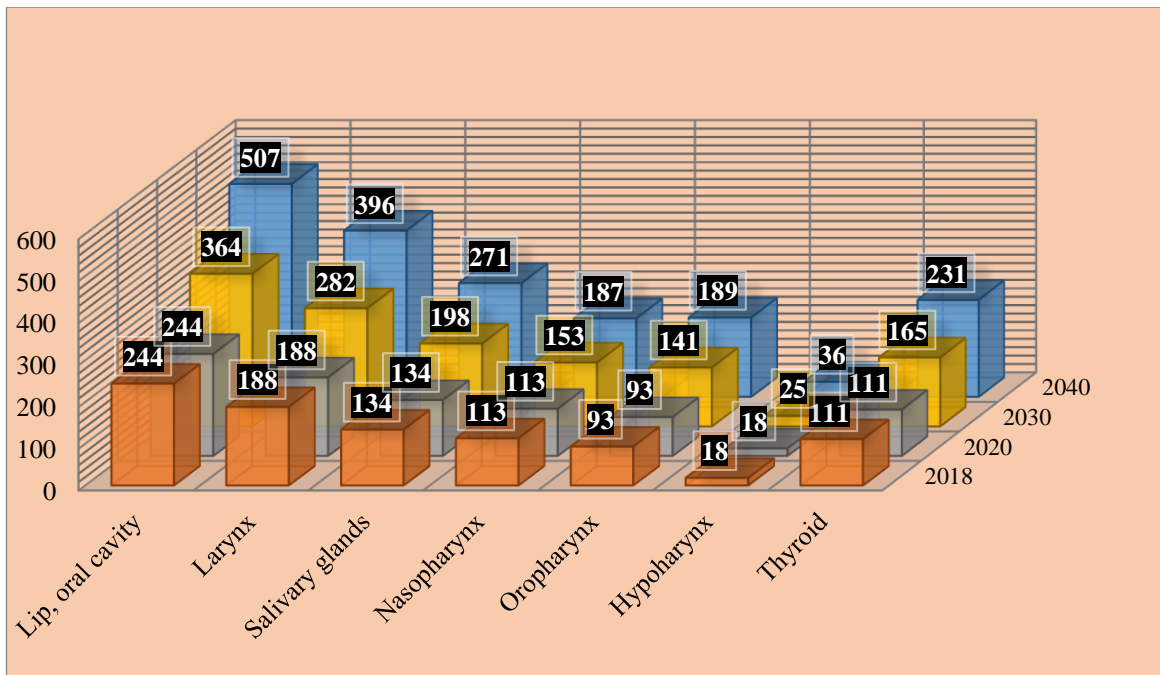


Figure 2.6 Estimated Number of HNCs Incidences in Ghana from 2018 to 2040
 Source: GLOBOCAN (2018)

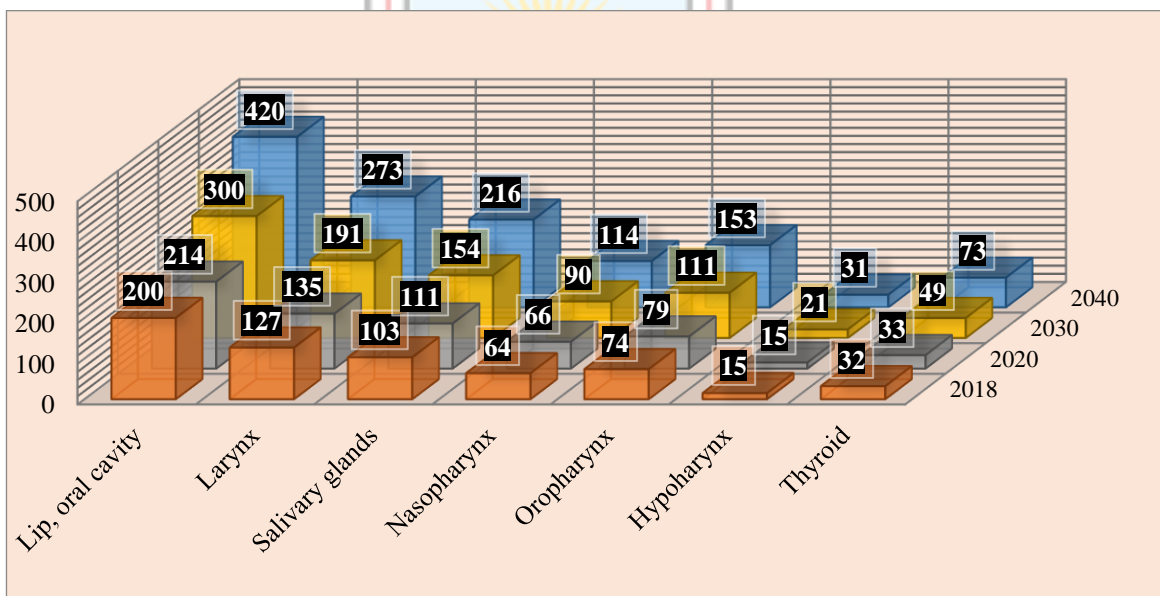


Figure 2.7 Estimated Number of HNCs Deaths in Ghana from 2018 to 2040
 Source: GLOBOCAN (2018)

2.6 Risk Factors of Head and Neck Squamous Cell Carcinoma

HNCs have low incidences as well as recurrences relative to other cancers but are possibly dangerous if identified lately. Therefore, identifying its possible associated risk factors of recurrence is doubtlessly very essential for early diagnosis, to combat or reduce the rate of

recurrence. Several factors have been identified as possible risks associated with HNCs recurrence as listed below.

2.6.1 Gender and Age

Previous studies (Razak *et al.*, 2010; Oliveira *et al.*, 2008; Chen *et al.*, 2007) have shown that increasing age is associated with HNCs incidences. About 90% of HNC cases are diagnosed more often among persons aged 50 years and above than they are among those less than age 50 (Anon, 2017). From a gender viewpoint, twice as males are susceptible to these cancer cases as females (Siegel *et al.*, 2017). In Ghana, males are still more susceptible to HNCs than females, as this might be a result of smoking. From the adapted Globocan 2012, Ghana Cancer Statistics, as shown in Figures 2.4 and 2.5, HNC is ranked as the 4th top most cancer in Ghanaian males (12%) and is ranked the 5th in Ghanaian females (4%). Among the HNC subtypes based on Globocan 2018, the lip-oral cavity has the highest incident rate and hypopharyngeal cancer has the least incident rate as shown in Figure 2.6.

2.6.2 Smoking, Tobacco, and Alcohol

The use of alcohol and tobacco accounts for most HNCs. Around 90% of HNC cases, particularly, oral and pharyngeal cancers have been attributed to smoking in the United States ((Reichart, 2001). HNCs are Alcohol and cigarettes or pipe smoking (including smokeless tobacco called snuff or chewing tobacco) contribute to the majority of risk factors of HNCs, particularly cancers of the hypopharynx, oropharynx, oral cavity, and larynx (Exarchos *et al.*, 2012b; Boffetta *et al.*, 2008; Gandini *et al.*, 2008; Hashibe *et al.*, 2007). In spite of the fact that alcohol and tobacco use are risk factors for salivary gland cancer, people who use both have a higher risk of acquiring HNCs than people who only use one of the two (at least 75% of HNCs are linked to both, according to Hasibe *et al.*, 2009). This risk factor's contribution to the development of cancer is categorized by the International Agency for Research on Cancer (IARC). Smoking is the leading preventable cause of laryngeal cancer (64%), pharyngeal cancer (37%), nasopharyngeal cancer (25%) and oral cavity cancer (17%), according to Brown et al. (2018).

2.6.3 Virus Infection

Oncoviruses (cancer-causing viruses) play a key role in some human cancers. These viruses are known to induce cancers by way of causing proliferative mutations/alterations of the

DNA in cells as well as cells of chromosomal structures. Human Papilloma Virus (HPV) is a small circular DNA virus that infects basal cells in the squamous epithelium. More than 40 HPV types are termed "genital type", which sometimes can infect the genital area of a man or woman, and can also infect the areas of the head and neck. HPV types can be cancerous or noncancerous. HPV-6 or HPV-11 strains which are noncancerous are called low-risk HPV, whereas HPV-16 or HPV-18 strains which are cancerous are called high-risk HPV. HPV causes cancers through viral oncoproteins *E6* and *E7*, which inactivate two tumor suppressor genes, *p53* and *RB* (Kennedy *et al.*, 2014). Studies have shown that certain strains of HPV infection; HPV-16 and HPV-18, which are well known to be 70% potential oncoviruses for cervical cancer are also 80% linked to the progression or redevelopment of HNSCC (Anon, 2020; Reichart, 2001).

2.6.4 Genes Mutation

HNSCC has been linked to a number of oncogenes and tumor suppressor genes through research (Oliveira *et al.*, 2008; Mehrotra and Yadav, 2006; Reichart, 2001). Therefore, it is now understood that a number of different groups of cellular genes, including tumor suppressor genes and mismatch repair genes, may be implicated in the multi-step process that might result in human cancer. The most often altered tumor suppressor genes are *p16* and *p53*, which are found on chromosomes 9p and 17p respectively and have been regularly seen in HNSCC (Liu *et al.*, 2013), with *p53* likely being the gene that has been investigated the most. In contrast to other human-related malignancies, HNSCC instances have been shown to be associated with *p53* over-expression or mutation in 40% to 90% of examined cases. Another tumor suppressor gene, known as *p63*, has also been researched and may offer a better outlook for HNSCC (Oliveira *et al.*, 2008).

2.7 Clinical, Pathological, and Genomic Markers

Several markers can be considered when determining the cancer prognosis. Here, three types; clinical markers, pathological markers, and genomic markers are considered. Conventionally, clinicians use clinical and pathological markers in determining the prognosis of a patient with cancer. Meanwhile, even the most skillful clinician may not find it easy in coming out with an accurate prognosis where only clinical and pathological markers are used. It is this that makes it necessary to combine biomarkers or genomic markers to obtain an improved prognostic accuracy result.

Investigations on tumor markers, being oncogenes as well as mutations of tumor suppressor genes have been conducted as prognosis outcomes of such molecular alterations to clinical and pathologic information. The study of these markers could induce proper treatment that is highly tailored to the patient's tumor.

2.7.1 Clinical Markers of HNSCC

The clinical staging of HNCs is a core part and necessary in cancer studies as it enables the clinicians to; determine a treatment plan, assess treatment modalities, and internationally compare cancers of various aspects. The staging system looks at the size and depth of the malignant tumor and whether it has metastases (Mehrotra and Yadav, 2006).

2.7.2 Histopathological (Pathological) Markers of HNSCC

Histopathological data defines the results the histopathologists obtain from the microscopic examination of tissues and/or cells removed termed a biopsy, and parameters being; tumor grade, size, depth of invasion, and other post-surgical pathologic margins (Mehrotra and Yadav, 2006).

2.7.3 Genomic Markers of HNSCC

The core challenge with the Tumor, Node, Metastasis (TNM) staging system is its failure to consider the biological and molecular characteristics of the tumor, and as such, might not offer accurate outcomes for the patient (Oliveira *et al.*, 2008). Cancer evolves via multiple stages, so that, the clonal expansion which follows the chronological simulation of additional genetic defects characterizes each stage for which it occurs (Mehrotra and Yadav, 2006). For one to better identify genetic alterations to HNSCC subtypes, a comparison between progressing and non-progressing subtypes of HNC lesions, which are genetically different from each other be taken into consideration (Mehrotra and Yadav, 2006). Oncogene activation and tumor suppressor genes (example; *p53*, *p63*) inactivation (impairment) are the main cause of genetic alterations (example *cyclin*, *ras*) observed in HNSCC, leading to cell proliferation in an uncontrolled manner (Anon, 2015; Mehrotra and Yadav, 2006).

2.8 Management of Head and Neck Cancers

The classification of cancer management focuses on three main foci as discussed below.

2.8.1 Diagnosis

A cancer diagnosis is a process where the cancer is recognized by its signs and symptoms. The doctors then need to perform a number of medical examinations, which may include a biopsy, laboratory testing (blood tests, urine tests), and imaging techniques (micro-CT scan, X-rays, ultrasound). Most often, doctors use the biopsy to identify malignancy. This is done by taking a sample tissue with a needle/endoscope following surgery, which is taken to the pathologists to be viewed and examined under a microscope. The patient then has to go through treatment and procedures of prognosis upon cancer confirmation (Foundation, 2010).

2.8.2 Treatment

The most effective treatment options for head and neck malignancies include surgery, radiation, chemotherapy, targeted therapy, or a combination of these (Gangil *et al.*, 2022). The choice of a treatment plan for this cancer varies depending on the stage, site, tumor grade, and age of the patient including baseline medical condition.

Surgery

The usual treatment for head and neck cancer for curative intent is mostly surgery, which is normally performed on less invasive tumors. The surgery can be very complex depending on where the tumor is located, its size, and what other structures it invades. If surgery is an option, the goal of the surgeon is usually to remove the entire tumor including some healthy tissue around it (this process is termed as achieving negative or clean margins). In effect, surgery can remove the entire tumor if that tumor is localised, however, complete surgical excision is somewhat not possible if cancer has metastasised to other sites. Sometimes, lymph nodes can be removed for the pathologist to perform a further evaluation. Surgery may be used in combination with chemotherapy or radiotherapy as a cancer treatment option (Foundation, 2010).

Radiation Therapy (Radiotherapy)

Treatment of cancer by radiotherapy is the use of ionizing radiation (high-energy x-rays) to kill the tumor. Two different methods of radiation can be given; external beam or brachytherapy (internal radiation). A machine called a linear accelerator is used to administer external beam radiation, which points beams of radiation from many angles

toward the tumor. In an attempt to target all cancer, the nearby healthy cells are hit by radiation, leading to damaging the DNA in those cells. Brachytherapy (also known as internal radiation) on the other hand, involves internally implanting radioactive material/source in the tumor and/or around the site of the tumor. The radiation is released slowly by this source over time, delivering it to a small tissue or tumor area. There is a decrease or fewer side effects as only a small area is being treated, but this can also increase the likelihood of cancer recurrence. Thus, ionising radiation kills both healthy (noncancerous) and cancerous cells in the treated area in which the DNA in such healthy cells are damaged, restricting them from continuously growing. Meanwhile, these healthy cells can repair themselves. Localised tumors (oral cancer) and lymphatic cancer (lymphoma cancer and leukemia) are suitably treated with radiotherapy (Anon, 2019; Foundation, 2010).

Chemotherapy

Chemotherapy is a medication that is used to destroy cancer cells in the body. Typically, chemotherapy is used in more advanced tumors of the head and neck. Sometimes, chemotherapy is targeted to shrink a tumor to make its removal very easy. Chemotherapy as the best in treating metastatic cancer (widespread cancer) can be given either before or after surgery. Chemotherapy that is given before and after surgery is respectively called neoadjuvant and adjuvant chemotherapy. Depending on the subtype of HNC a patient has chemotherapy medications to be given. The most common chemotherapy medications used for the treatment of HNC include cisplatin, carboplatin, paclitaxel, docetaxel, epirubicin, gemcitabine, and methotrexate. Chemotherapy is usually given in conjunction with radiation therapy, referred to as chemoradiation therapy. In this situation, the chemotherapy performs two functions: to induce the radiation to work better and to treat/kill cancer cells. This is termed radio-sensitization, where the cells are made more sensitive to radiation damage in the presence of relatively low chemotherapy doses because the healthy cells also become sensitized, causing more severe side effects compared to radiation alone. Chemoradiotherapy has been proven best choice for HNC patients and to improve the treatment of laryngeal cancer but comes with more toxicity (Anon, 2019).

Targeted Therapy

Targeted therapy is used to treat the majority of HNSCC subtypes that have an "overexpression" of the Epidermal Growth Factor Receptor (EGFR). These subtypes might

produce too many of these receptors, which may be a key factor that induces their growth. In turn, targeted therapies were developed to block these receptors, with the main intention of slowing the growth of tumors. The two most commonly used EGFR inhibitors in HNCs are cetuximab and afatinib. Other two types of targeted therapies are: nivolumab and pembrolizumab, and are used to treat certain advanced HNCs. These medications stimulate the immune system to destroy cancerous cells, and so, are called immunotherapies (Anon, 2019).

2.8.3 Prognosis

Prognosis is when the outcome of the disease as well as its recurrence and the patient's status of survival can be predicted, in the presence or absence of the treatment either. Prognosis, is thus, simply the estimate of the likely course and treatment of the disease; the chance of either recurrence or survival. Prognoses are mostly predicted and grounded on several medical factors including cancer type, stage at diagnosis, treatment type, and patient's age at diagnosis etcetera.

2.8.4 Follow-Up/Recurrence

Cancer recurrence is when cancer comes back. When cancer cells were not fully removed or destroyed by the first treatment, the disease recurs; and this does not mean that the first treatment a patient received was wrong. It simply means that a small number of cancer cells, which were too small to show up during follow-up tests survived the treatment. Over time, these cells grew into cancer or tumors that can now be detected, which is termed recurrent cancer (cancer relapse). Cancer may recur in the same place as the original cancer (called a local recurrence) or by growing into lymph nodes (tissues) near the original cancer (called regional recurrence) or by metastasis (spread) to tissues (organs) far from original cancer (called distant recurrence) (Anon, 2016).

2.9 Works Related to ML Techniques in Recurrent HNSCC Prognosis

Some current research including (Alabi *et al.*, 2019; Cai *et al.*, 2019; Tang *et al.*, 2019; Singh *et al.*, 2019; Ribeiro *et al.*, 2017; Su *et al.*, 2017; Yang *et al.*, 2017; Exarchos *et al.*, 2011a; 2012b) employed ML techniques on markers of clinicopathologic and/or that of genomic for HNSCC (subtypes) prognosis. There are not many published articles that employed several supervised ML classifiers to train these classifiers on the combination of

clinicopathologic and genomic information for prognosis. Nevertheless, the findings from studies by Duda *et al.* (2001); Mitchell (1997) proved that a way to generate more accurate prognosis in the domain of cancer study is by the use of ML and that Kourou *et al.* (2015); Cruz and Wishart (2006) also proved that several ML techniques be trained for an optimum prognostic model.

Early-stage oral tongue squamous cell carcinoma (OTSCC) locoregional recurrence was predicted by Alabi *et al.* (2019). Their trained classifier (ANN), which was a feedforward neural network type, was used to analyze the usage of ANNs and Logistic Regression (LR) on 311 patients with OTSCC treated between 1979 and 2009. Numerous prognostic factors, such as WHO grade, tumor budding, depth of invasion, worst pattern of invasion, lymphocytic host response, and perineural host response, were examined in order to determine their effects on the prognostic result. Tumor budding and depth of invasion were revealed to be the two most important variables that can best predict locoregional recurrence of early-stage OTSCC, according to their findings, which demonstrated that an ANN classifier surpassed LG with an accuracy of 92.7%.

Another study that applied ML techniques in recurrent HNSCC regions was done by Exarchos *et al.* (2011a; 2012b). They formulated a Decision Support System (DSS) and Dynamic Bayesian Network (DBN) in their first and second study respectively, where clinical, imaging and genomic data were used to identify the prognosis that dictates the progression of Oral Squamous Cell Carcinoma (OSCC), and subsequently predict its potential recurrence (local versus metastatic). Even though both studies employed ANN, BNN, SVM, RF, and DT classifications, their first study trained these algorithms on clinical, imaging, and genomic data on 41 patients whereas these algorithms were trained on clinical, imaging, tissue genomic, and blood genomic data on 86 patients in their second study. Initially, in both studies, each classification considered was trained separately on various datasets used, where the respective predictions were integrated to produce a consensus classifier model, discriminating between patients with and without disease recurrence. Following feature selection and 10-fold cross-validation, clinicopathological markers (lymphoplasmacytic reaction, lymphovascular invasion, family history of malignancy, smoking, duration of smoking, ex-smoker, alcoholism, drinking habits, infection, physical agents, eating habits, precancerous lesions, tumor thickness, grade differentiation, and surgical margins) and genomic features (*THC2339617*, *CTAG1A*, *LPO*, *CLDN8*, *SCGB3A1*,

MSLN). In their second study, clinicopathological features (tumor thickness, grade differentiation, N staging, depth of invasion, smoking, p16ink4a, and eating habit) and genomic features (for example *p53* stain), were the significant predictors of the prognostic outcome, with DBN accuracy of 100%.

For the purpose of predicting the unique pattern of recurrence for locally advanced Nasopharyngeal Carcinoma (NPC), Cai *et al.* (2019) introduced molecular decision tree algorithms. A decision tree classifier was used as their prediction method, which was created to forecast the patient recurrence pattern (with versus without recurrence) for locally progressed NPC. It was trained on the integration of numerous genetic and clinicopathological variables. 13 molecules (*AKT1*, *Aurora-A*, *Bax*, *Bcl-2*, *N-Cadherin*, *CENP-H*, *HIF-1*, *LMP-1*, *C-Met*, *MMP-2*, *MMP-9*, *Pontin* and *Stathmin*, and *N stage*) with different expression levels in tumor specimens were chosen using these data on 136 patients to build the decision tree classifier. By employing a 10-fold cross-validation technique, their prognostic model was developed in the training subset and tested (validated) in the validation subset. They were able to predict individual recurrence patterns with an overall prognostic model accuracy of 84.5–95.2%. The DT classifier was shown to be an independent prognostic tool in predicting individual recurrence by multivariate analysis, which also supported the prediction of the DT.

Another study that applied ML (KNN, SVM, and Bagged Trees (BT)) techniques in recurrent HNSCC was that of Singh *et al.* (2019) which examined whether or not disease treatment with radiation therapy may be followed by recurrence. Their study was actually to determine whether the markers present in the heterogeneous regions of tumor in the pre-treatment PET scans of patients with HNSCC can serve as prognoses for disease recurrence. Information on the patient's gene mutation as an additional feature was included to determine its efficacy for radiation therapy treatment. The Cancer Genome Atlas (TCGA) identified *PIK3CA*, *CDKN2A*, and *TP53 genes* to be mutated genes in HNSCC. It was then found that, when gene expression features (dataset from TCGA) were combined with texture features (from The Cancer Imaging Archive (TCIA)) on 11 patients, there was an increase in the classification accuracy from 80% for texture features used only to 100% when gene expression features were combined for patients with recurrence, and from 60% to 100% for patients with nonrecurrence. Models were trained and tested using a 50-fold cross-validation

method. It was concluded that gene-expression features which, when combined with tumor texture, prediction of therapy response for recurrent HNSCC patients can be improved.

Further, Yang *et al.* (2017) employed an SVM classifier and Cox regression analysis on an 80-gene set potentially to identify the prognosis that predicts recurrence in Laryngeal Carcinoma (LC). After relevant genes were identified by Cox regression analysis as being associated with tumor relapse, the Protein-Protein Interaction (PPI) network was built using these genes. Then, using genes in certain PPI networks, an optimal SVM classifier was created that could categorize samples of recurrent LC. Their classifier determined the top ten (10) genes in certain PPI networks by ranking them according to their BC (betweenness centrality) values: *APP*, *NTRK1*, *TP53*, *PTEN*, *FNI*, *ELAVL1*, *HSP90AA1*, *XPO1*, *LDHA*, and *CDK2*. The SVM classifier demonstrated 100% accuracy for classifying recurrent cases from LC samples. Later, the effectiveness of the SVM classifier was evaluated using separate datasets to predict the recurrence or relapse of specific patients, and the results indicated a 97.47% accuracy rate.

2.10 Works Related to Stacked Ensemble Techniques in Cancer Study

Some current research including (Akinbohun, 2021; Akinbohun *et al.*, 2020) employed a stacked ensemble on head and neck cancer data, whilst (Adeyemi *et al.*, 2019; Kabir and Ludwig 2019; Kwon *et al.*, 2019) employed a stacked ensemble learning technique on breast cancer data. There is currently no published article that employed a stacked ensemble learning technique on recurrent head and neck cancer data. Nevertheless, the findings from studies by Kabir and Ludwig (2019); Kwon *et al.* (2019) proved that a way to generate the most accurate prognosis and promising prognostic model in the domain of cancer study is by the use of stacked ensemble technique.

Akinbohun (2021); Akinbohun *et al.* (2020) both proposed a stacked ensemble technique having three single base classifiers; KNN, NB, and DT (C4.5), where GLM was used as a meta-learner to stack these base classifiers in stacking ensemble in the diagnosis of HNC susceptibility able to; facilitate prompt referral and predict the cancer types around the regions of HNC (Sinonasal, nasopharyngeal, laryngeal, and thyroid) respectively. For Akinbohun *et al.* (2020), the accuracies of DT (93.21%), KNN (94.80%), and NB (94.12%) are less than the accuracy of the stacked ensemble model (95.11%). Both investigations concluded that, in regards to Akinbohun, a stacked ensemble gave outcomes that were more

accurate than those of the basic learners. As a result, healthcare systems can employ the layered model to diagnose HNC.

Adeyemi *et al.* (2019) proposed a stacked ensemble model (with 10-fold CV) having two single base classifiers; NB and SVM (with DT as a meta classifier), NB and DT (with SVM as a meta classifier) and SVM and DT (with NB as a meta classifier) for the classification of recurrence of breast cancer prognosis. The results showed that the performance of the stacked ensemble models that used SVM and NB as the meta-classifiers did not differ from one another; however, the stack ensemble model that used NB and SVM as base classifiers and the DT as a meta-classifier had the best performance overall. As a result, using the DT as a meta-classifier shown a greater ability to classify recurrent breast cancer than SVM and NB classifiers.

Kabir and Ludwig (2019) proposed a stacking ensemble-based algorithm, a technique that found the best-weighted average of varied base learners for the classification of various healthcare datasets including the Wisconsin Breast Cancer dataset, using GBM and DRF as base classifiers in one case and GBM, DRF, and DNN as base classifiers in another case. GLM was used as a meta-classifier in each case to obtain the best combination of base classifiers. It was concluded that the stacking ensemble having three base classifiers (with an accuracy of 99.29%) outperformed that of the stacking ensemble having two base classifiers (with an accuracy of 98.57%), but both had the same AUC (0.998) on breast cancer. The accuracy for the State-of-the-Art (SA) was 97.57%. Based on the same data set, accuracies of 99.29%, 98.57%, 99.29%, and 97.90% were obtained for GBM, DRF, DNN, and baseline ensemble techniques respectively; and with their respective AUC of 0.997, 0.997, 0.998, and 0.996. Their experimental results showed that the stacked ensemble model consisting of three base classifiers had the best performance accuracy compared to that with two base classifiers as well as single base and baseline ensemble models.

Kwon *et al.* (2019) also proposed a stacking ensemble-based algorithm, a method for classifying breast cancer that used GBM, DRF, DNN, and GLM as base learners in a stacking ensemble, and each of them as a meta-learner to stack the base learners. Based on the experimental findings, they came to the conclusion that utilising specific models as a meta-learner produced higher performance than using single classifiers, and that using GBM or GLM as a meta-learner is suitable as a supporting tool for categorising breast cancer data.

Table 2.1 Summary of Stacked Ensemble Techniques in the Cancer (Subtypes) Studies

Cancer Type	Base-learner	Meta-learner	Accuracy	Validation method	Reference
HNC	KNN, NB, DT	GLM (LMT)	95.11%	5-fold CV	Akinbohun (2021)
HNC	KNN, NB, DT	GLM (LMT)	95.11%	5-fold CV	Akinbohun <i>et al.</i> (2020)
Breast cancer	(NB & SVM), (NB & DT), (DT & SVM)	DT, SVM & NB respectively	72.38%, 70.98% & 83.0% respectively	10-fold CV	Adeyemi <i>et al.</i> (2019)
Breast cancer	(GBM & DRF), (GBM, DRF & DNN)	GLM	98.57%, 99.29% respectively	10-fold CV	Kabir and Ludwig (2019)
Breast cancer	GBM, DRF, DNN & GLM	GBM, DRF, DNN & GLM	97.96%, 97.37%, 97.96% & 97.96% respectively	5-fold CV	Kwon <i>et al.</i> (2019)

Table 2.2 Summary of ML Techniques in the Prognosis for Recurrent HNSCC (Subtypes)

Cancer Type HNSCC (subtypes)	ML Technique	Benchmark	Accuracy	Validation method	Training Data	Important Features	Reference
Locoregional (oral tongue)	ANN	LR	92.7%	Cross- entropy	Clinical & histopatho- logical	Tumor budding, depth of invasion	Alabi <i>et al.</i> (2019)
Nasopharyngeal	DT algorithm	Statistics	84.5–95.2% & AUC = 91.3%	10-fold cross- validation	Clinicopat- hological & molecular	AKT1, Aurora-A, Bax, Bcl-2, N- Cadherin, CENP- H, HIF-1 α , LMP-1, C-Met, MMP-2, MMP-9, Pontin and Stathmin, and N stage	Cai <i>et al.</i> (2019)
Laryngeal cancer	SVM	N/A	94.05%		Genomic		Tang <i>et al.</i> (2019)
Nasopharyngeal carcinoma	DT, KNN, LDA, LR, NB, RF & RBF-SVM	N/A	AUC= 88.3- 89.2%		Radiomic		Du <i>et al.</i> (2019)

HNSCC	SVM, KNN & Bagged Trees	N/A	100%	50-fold cross-validation	Clinical, imaging & genomic	PIK3CA, CDKN2A and TP53 genes	Singh <i>et al.</i> (2019)
Nasopharyngeal Carcinoma	ANN, KNN, SVM	Statistics	ANN:81.2%, KNN:77.5%, SVM:73.2% AUC=72.7-83.5%	10-fold cross validation	Radiomic		Li <i>et al.</i> (2018)
Laryngeal cancer	SVM	N/A			Genomic	PDIA3, MYH11, PDK1, SDC3, RPE65, LAMC3, BTK, and UPK1B	Su <i>et al.</i> (2017)
Laryngeal cancer	SVM	Cox Regression	100%		Genomic	APP, NTRK1, TP53, PTEN, FN1, ELAVL1, HSP90AA1, XPO1, LDHA and CDK2	Yang <i>et al.</i> (2017)
HNSCC	SVM	N/A	87.0%	Cross validation	Clinic-pathologic		Ribeiro <i>et al.</i> (2017)

					& genomic		
Oral Cancer	BN	ANN, SVM, DT, RF	100%	10-fold cross validation	Clinical, imaging tissue genomic, blood genomic	Smoking, <i>p53</i> stain, depth of invasion, extra-tumor spreading, grade differentiation, N staging	Exarchos <i>et al.</i> (2012b) and Exarchos <i>et al.</i> (2011a)
Oral Cancer	SVM	N/A	98%	Cross Validation	Clinicopathologic, molecular		Rosado <i>et al.</i> (2013)
Throat	ANN	Statistics	86%		Genomic		Kan <i>et al.</i> (2004)

2.10.1 Identified Gaps in the Review of Related Literature

Table 2.1 provides a summary of the stacked ensemble techniques based on ML algorithms that have been employed in developing stacked ensemble models on various cancer datasets. Table 2.2 provides a summary of ML techniques that have been employed in the prognosis for recurrent HNSCC subtypes. Though previous studies provide some useful results, there is still some area of cancer study that currently needs to be explored. The aspects that have not been explored can be put into two phases. In the first phase, there is currently no much exploration on a stacked ensemble technique for recurrent HNSCC prognosis (Chi-Chang *et al.*, 2021; Yarney *et al.*, 2017). Information gathered from the previous studies shows that the stacked ensemble ML techniques can produce more stable, unbiased, and reliable prognosis and prediction results at a higher level of accuracy compared to the statistical and standalone ML techniques (Akinbohun, 2021; Chi-Chang *et al.*, 2021; Akinbohun *et al.*, 2020). In the second phase, no standalone ML techniques in the literature is considered a generalised prognostic model for recurrent HNSCC prognosis based on the integration of clinicopathologic and genomic markers, being the domain of prognostic model development (Chi-Chang *et al.*, 2021; Yaliang *et al.*, 2015). Nonetheless, no prior study employed stacked ensemble ML techniques in the prognosis for recurrent HNSCC based on the integration of clinicopathologic and genomic markers. As a result, none of the previous studies has carried out the study of possibility of developing a classification model with a good generalisation ability to be considered a robust prognostic model for recurrent HNSCC prognosis (Chi-Chang *et al.*, 2021; Yarney *et al.*, 2017).

Based on these aftermaths, the present study proposes a Hybrid Ensemble Super Classification Algorithm (HESCA) model with the ability of learning the stacked generalisation on the integration of clinicopathologic and genomic markers.

CHAPTER 3

METHODS USED

3.1 Overview

This chapter is in two sections: the first section discusses the methodological framework of the study which specifies the study area, research design, study population, sample size, data collection instrument and procedures, and data handling/control. The second section discusses the theoretical background for the development of the models. This specifies the theories and techniques used to develop the existing models.

3.2 Methodological Framework

3.2.1 Study Area

The National Centre for Radiotherapy and Nuclear Medicine division of the KBTH is the study area. KBTH, which opened its doors on October 9th, 1923, is situated inside the Accra Metropolitan Assembly. The hospital, which is today the third-largest hospital in Africa, is well-known as the top national referral centre in Ghana. It contains more than 2000 beds, 21 clinical and diagnostic Departments, and three Centres: National Cardiothoracic Centre, the National Centre for Radiotherapy and Nuclear Medicine, and the Reconstructive Plastic Surgery and Burns Centre. The radiotherapy and oncology departments of the National Centre for Radiotherapy and Nuclear Medicine are responsible for the treatment of cancer. The center uses ionizing radiation and chemotherapy to treat solid (malignant) cancers as well as benign tumors. The statistics section of the radiotherapy department, from which the study's data was acquired, maintains a database of every cancer case. The facility employs at least 60 people, including physicians, nurses, oncologists, treatment radiologists, and biological scientists.

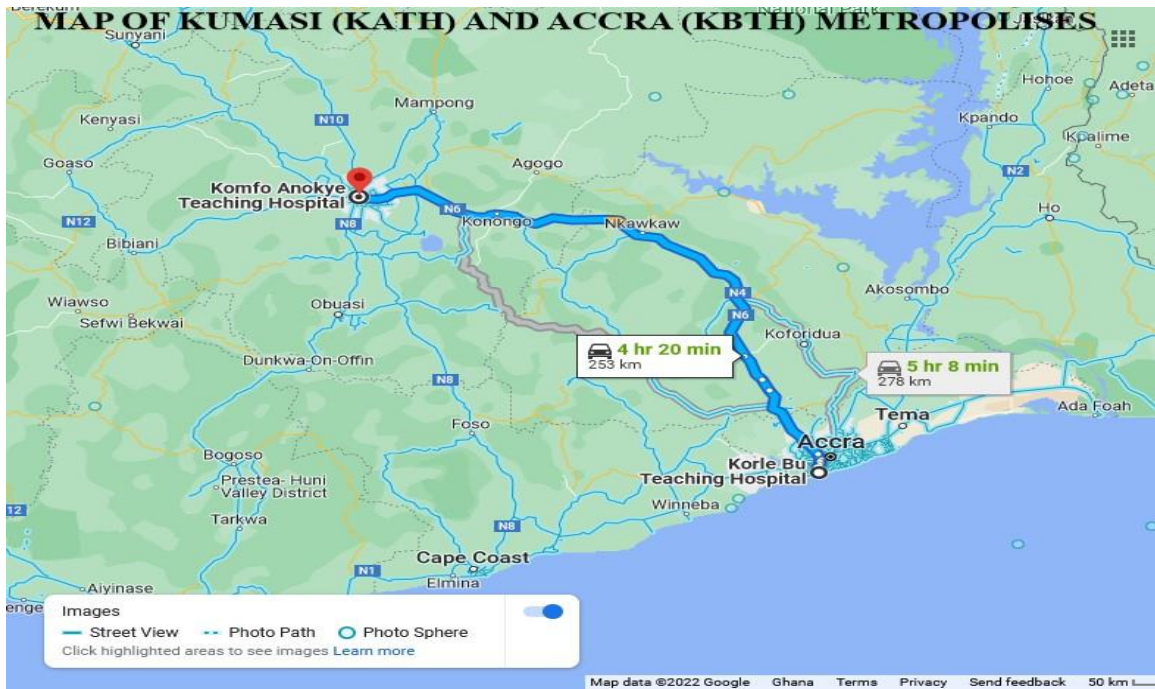


Figure 3.1 Study Location (Source: Map Data, 2022)

3.2.2 Study Design

The study makes of a retrospective cohort design of HNSCC patients based on their cancer records at the medical facilities, who had curative intent treatment at KBTH in Accra, but had or had no relapse within the 2016 – 2020 periods inclusive. These patients were either originally diagnosed of HNSCC at KBTH or KATH but those that were diagnosed at KATH had referrals to KBTH for treatment plan. Therefore, data on patients with referrals from KATH to KBTH is collected at KBTH for the analysis.

3.2.3 Study Population

The study population consists of 185 patients with ages ≥ 15 years who reported at KBTH and KATH and were previously diagnosed of HNSCC and all had curative intent treatment at KBTH between 2016 and 2020 calendar period. This calendar period is chosen because the required information on patients that were diagnosed of HNSCC before the year 2016 are insufficient. Patients below 15 years are mostly lost to follow-up and so there is not much medical information on them.

Inclusion Criteria

All patients of at least 15 years, that reported at the department of Radiotherapy and Oncology (KBTH) and KATH, and were initially diagnosed with primary HNSCC at the

site; larynx, nasopharynx, oropharynx, or hypopharynx between the studied calendar periods, and had curative intent treatment only, and had been followed-up until the end of 2020, and experienced cancer recurrence or no recurrence are included in the study.

Exclusion Criteria

All patients of any age that formerly had a diagnosis of HNSCC subtypes at these departments between the period under study but had palliative intent treatment are excluded from the present study.

3.2.4 Sample Size Determination

The population size of 185 for the calendar period 2016 to 2020 has 60 instances with no much information as these patients were lost to follow-up. As a result, convenience sampling technique is used to withdraw such instances from the dataset at hand. Thus, a sample size of the original study is computed using the Cochran (2007) formula:

$$n = N \times \frac{\frac{Z^2 \times p(1-p)}{\varepsilon^2}}{\left[N - 1 + \frac{Z^2 \times p(1-p)}{\varepsilon^2} \right]} \quad (3.1)$$

where n = sample size,

N = population size

Z = the z-value attributed to a 95% confidence interval (1.96),

p = proportion of patients with versus without recurrent HNSCC,

the $q = 1 - p$ = proportion of HNSCC patients with a second primary tumor, and

ε = precision (margin of error) set at 95% CI = 0.05.

Given population size, $N = 185$, $Z = 1.96$, $\varepsilon = 5\%$ or 0.05, and since p is unknown, let us assume $p = 0.5$.

$$\begin{aligned} n &= 185 \times \frac{\frac{(1.96)^2 \times 0.5(1-0.5)}{(0.05)^2}}{\left[185 - 1 + \frac{(1.96)^2 \times 0.5(1-0.5)}{(0.05)^2} \right]} \\ &= 125.087 \approx 125 \end{aligned}$$

The simulation in excel file is shown in appendix J.

3.2.5 Data Collection Instruments

Due to the nature of research objectives, the secondary data entered in the folders of patients with or without the recurrence at KATH and KBTH during their diagnoses and follow-up tests is collected and used.

3.2.6 Data Collection Procedures

The source of (Hospital electronic) database for recorded data of years 2016 – 2020 on patients with HNSCC is first accessed. Data points on patients' possible prognostic factors associated with versus without recurrent HNSCC as shown in Table 3.1 below are then extracted from folders based on a distinctive identification number assigned to each of them, using data extraction form as well as Microsoft Excel. Data points are finally reviewed for analysis.

Table 3.1 Demographic, Clinical, Pathological and Genomic Features

	Name of Features	Description
i	Gen	Gender
ii	Age	Patient's age
iii	Alc	Alcoholism
iv	Smoke	Smoking
v	Chew	Quid/Tobacco usage
vi	Site	Site of primary tumor
vii	Stage	Stage of tumor at diagnosis
viii	Grade	Pathological grade
ix	Size	Size of tumor
x	Inv	Depth of invasion
xi	Node	Lymph nodes
xii	PaT	Pathological stage of tumor
xiii	PIN	Lymph nodes pathologically
xiv	FHx	Family of cancer hereditary
xv	HPV	Human papillomavirus level
xvi	<i>p16</i>	p16 level
xvii	<i>p63</i>	p63 level
xviii	Treat	Treatment modality

3.2.7 Data Handling and Quality Control

To guarantee the uniformity and dependability of the data to be collected, the researcher hired and educated research assistants with prior experience in a comparable project. To make sure that all the necessary data had been properly gathered and entered, there was a daily check of data points and duplicate data entry. Before cleansing and editing of the data to remove the effects of inconsistencies or incompleteness, the data is loaded into the R software to verify for completeness. This ensures the data's quality. Since there are few missing values in the dataset, the data are cleaned using the imputing approach. The modes or high frequency-category values of the missing value in the categorical characteristics are used to fill the gap. This technique is useful because it is easy and fast, it changes the statistical nature of the data. A few missing values and outliers in the output feature are also replaced by the modal value. Information gathered on patients is saved on a computer and is classified as confidential information. Instead of the patient's name, a code number is assigned, which is kept a secret. Also, the data collected is anonymised.

The study considers numerous input features under various medical information that can be used to model medical cases. These include demographic features, clinicopathologic features, and genomic features. Each of these types of medical information may have several input features ranging from ten to as far as hundreds. However, in medical research, the number of instances of the dataset is usually small as much time is required to collect sufficient samples. Large training feature sizes will initially result in a high classifier performance for small-sized training examples with insufficiently large training data, but they will eventually degrade this high performance because there may be irrelevant and/or redundant features that could confuse the learner and cause model over-fitting, especially when there are few training examples and computational resources. In the face of avoiding the problem of model over-fitting, there is the need to perform feature selection techniques to select the input features that are most significant to the classifiers or clinical outcomes in the classification process. This process is required since the volume of data consists of more computational resources and is much more time-consuming. In this study, the vital goal of implementing the ensemble feature technique is to find the most accurate number of prognostic features for the small number of training examples of the HNSCC prognosis dataset.

3.2.8 Data Pre-processing

Data pre-processing is a technique that transforms raw data into a more effective space for easy learning. Several data pre-processing techniques mentioned in Section 3.2.7 are used. The prediction accuracy of a classifier model largely rests on the data quality. This makes it very important to pre-process the data at hand before the feature selection technique implementation.

The primary phase in data pre-processing is cleansing the data. It is discovered in the data collected that there is some missing and incomplete information in some instances. The missing values are mostly on chew, invasion, nodes, history, and HPV, and the incomplete examples are usually information on recurrence due to lost follow-up on a patient for which incomplete information was recorded. Several techniques can be used to handle missing training examples, such as mean, median or mode imputation, and case deletion. The latter deletes the instance that has missing examples under any feature; therefore, this technique is considered not feasible for this study as the size of training instances is very small. Categorical (nominal) data use mode imputation whereas continuous data uses either mean or median imputation (Acuna and Rodriguez, 2004). Based on Clark *et al.* (2003), and looking at the data at hand which is a categorical data, this study uses mode imputation. Thus, the accounted-for bias is less likely to occur, as the number of incomplete instances is small.

Once the data under study has been cleansed, the required format of categorisation and coding for predictive training is implemented. Here, there is a transformation of continuous features into nominal features. One-hot encoding technique is applied to features having more than two classes for normalisation of the dataset to learn, test, and prognosticate HNSCC recurrence. Based on this technique, the original 18 features as shown in Table 3.1 under study now give 35 features in the dataset as shown in Table 3.2. The label or target feature is also discretised into a factor consisting of two classes.

Table 3.2 Description of Features for 125 Instances

Feature	Description	Level (No.)	Feature	Description	Level (No.)
Gen (x_1)	Male Female	1 (92) 0 (33)	Inv (x_{10})	Cohesive Non-cohesive NA	1 (47) 0 (74) (4)
Age (x_2)	15-45 > 45	1 (51) 0 (74)	Node (x_{11})	Positive Negative NA	1 (61) 0 (44) (20)
Alc (x_3)	Yes No	1 (48) 0 (77)	PaT (x_{12})	T1 T2 T3 T4	0 (5) 1 (14) 2 (28) 3 (78)
Smoke (x_4)	Yes No	1 (38) 0 (87)	PIN (x_{13})	N0 N1 N2 N3	0 (34) 1 (14) 2 (43) 3 (34)
Chew (x_5)	Yes No	1 (24) 0 (93)	FHx (x_{14})	Yes No	1 (16) 0 (84)
Site (x_6)	Larynx Nasopharynx Oropharynx Hypopharynx	0 (40) 1 (70) 2 (12) 3 (3)	HPV (x_{15})	High-risk Low-risk NA	0 (38) 1 (15) (72)
Stage (x_7)	I II III IV	0 (7) 1 (23) 2 (33) 3 (62)	<i>p16</i> (x_{16})	Positive Negative NA	1 (74) 0 (36) (15)
Grade (x_8)	G1 G2 G3	0 (18) 1 (31) 2 (76)	<i>p63</i> (x_{17})	Positive Negative NA	1 (60) 0 (56) (9)
Size (x_9)	0-4cm > 4cm NA	0 (75) 1 (46) (4)	Treat (x_{18})	Chemotherapy (Chemo) Radiotherapy (RT) ChemoRT (CRT) Concurrent ChemoRT (CCRT) Surgery+RT Surgery+CRT Surgery+CCRT	0 (9) 1 (33) 2 (47) 3 (26) 4 (4) 5 (2) 6 (4)

NB: tumor invasion $\leq 10\text{mm}$:cohesive, and tumor invasion $> 10\text{mm}$:non-cohesive. G1, G2, and G3:Well, Moderately, and Poorly differentiated respectively, N2:(N2a,b,c), N3:(N3a,b), High-risk if HPV16 or HPV18, Low-risk if HPV6 or HPV11

Now, if compared to the training set of 125 instances, the 35 input features (of which there are 35 in total), are deemed to be excessive. To lessen the quantity of training features, a feature selection technique is therefore required. Thus, the ensemble feature selection technique only chooses the aspects that have been determined to be important to the HNSCC

prediction. Chapter 4, Section 4.2.1 provides an explanation of the specifics of how the ensemble feature selection technique was used in this study.

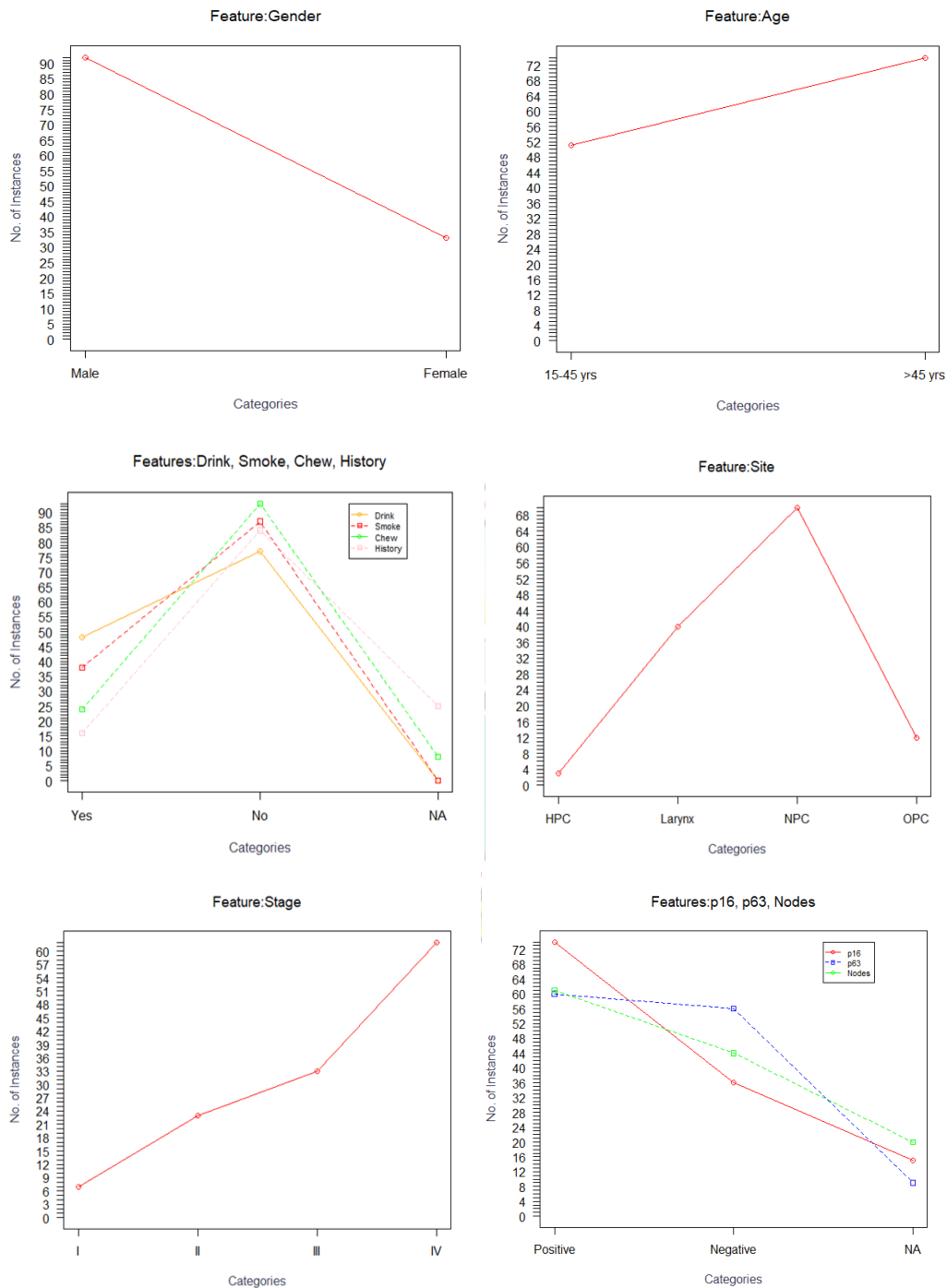


Figure 3.2 Graphs for Prognostic Features

Figure 3.2 shows the graphs of number of categories of each of the features considered in the study.

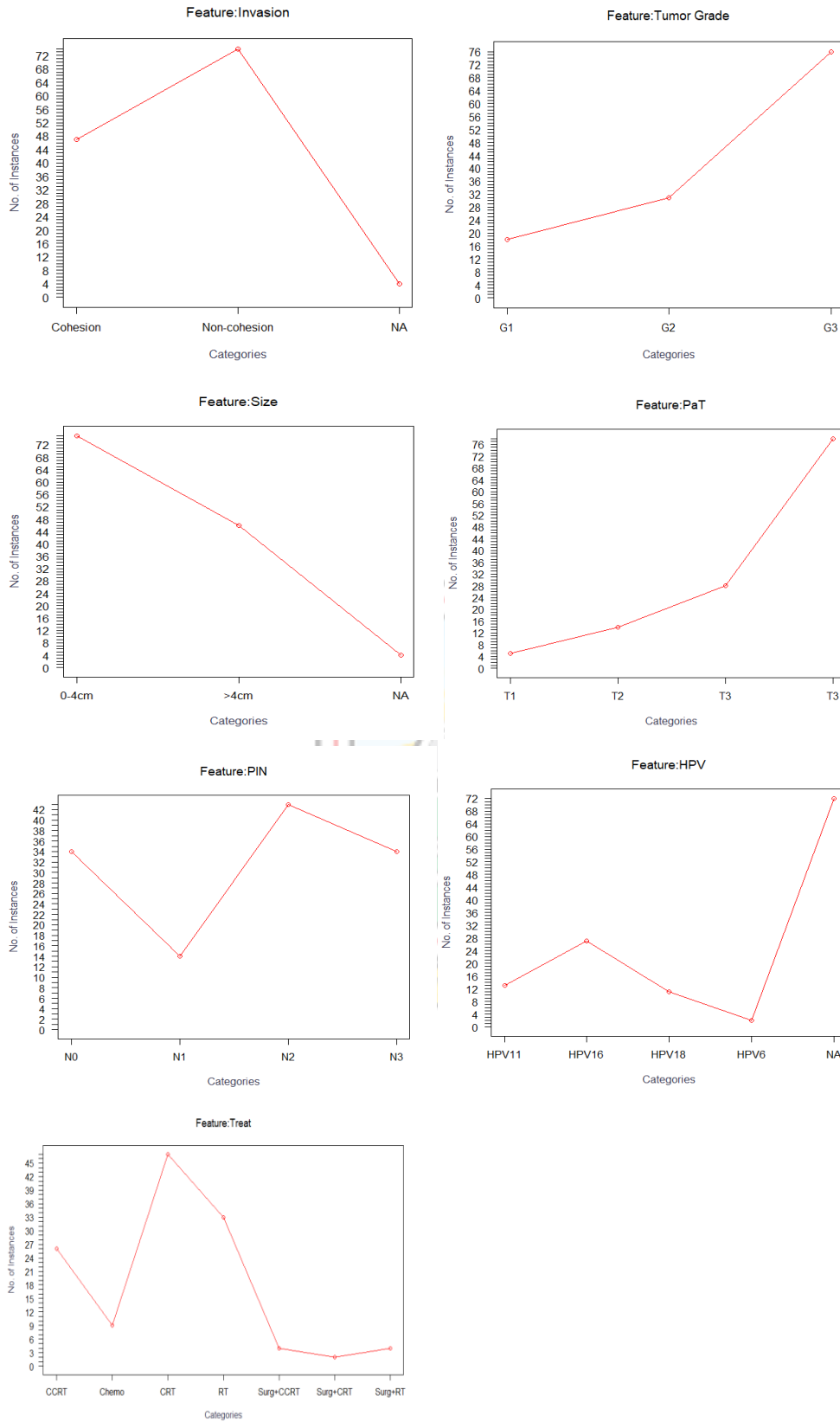


Figure 3.2 Graphs for Prognostic Features (continued)

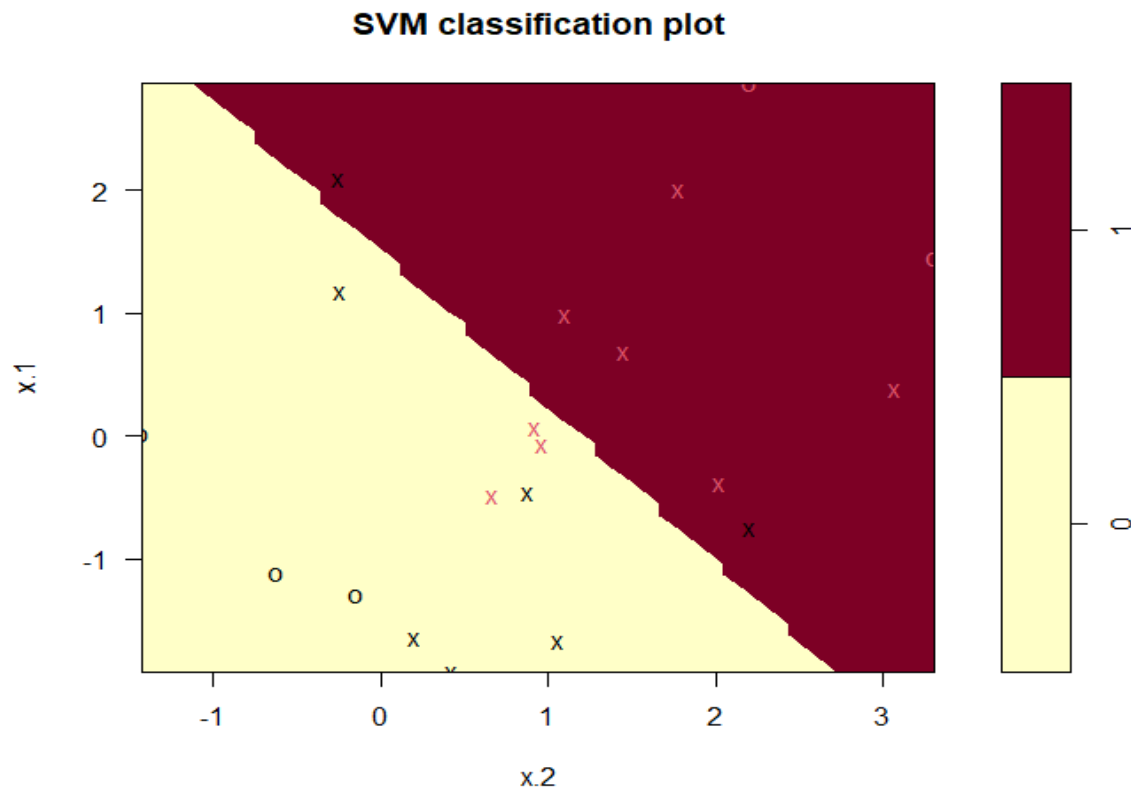


Figure 3.3 Plot of Linearity or Nonlinearity of the HNSCC Dataset

In order to check the linearity or nonlinearity of the HNSCC dataset, the linear SVM classifier that separates the training instances into two respective classes as positive and negative examples is used. It can be observed in Figure 3.3 that some training instances have been misclassified; some instances denoted by black colour belonging to the class 0 are misclassified among those instances belonging to class 1, and some instances denoted by red colour belonging to the class 1 are misclassified among those instances belonging to class 0. This shows that the HNSCC dataset under study is not linear or is nonlinear. This informs the choice of the classification algorithms that can be employed to learn the nonlinear dataset of the recurrent HNSCC prognosis.

3.3 Theoretical Background of Classifiers

Classification or supervised learning is one of the important tasks of the ML that tries to infer a function that maps feature values into class labels from the training data, and applies the function to the data with unknown class labels. In general, the classification learning aims at finding the model that attains good performance when predicting the unseen labels. To achieve good performance of a classification model on the available dataset, various

studies are using standalone (single) or homogeneous ensemble classifiers (classification models). Nevertheless, it is a complex task in the selection of a single data mining or ML classification model that achieves good generalisation ability for a given task. As result of this, the ensemble learning of multiple different learners were usually employed in the study for a given task in order to achieve good performance and generalisation ability of a classification model. This section discusses the theories behind the development of baseline stacked ensemble classification models, specifically baseline ensemble algorithms, supervised machine learning classification algorithms, and base feature selection algorithms.

3.3.1 Ensemble Learning

Ensemble learning offers credence to the ideas of the “wisdom of crowds,” which suggests that the decision-making of a larger group of learners is emphatically better compared to that of a single proficient learner. Meanwhile, ensemble learning defines a group of base learners, that works in an ensemble to achieve a strong and accurate final prediction. A single learner might not accurately perform well as they can be prone to overfitting or underfitting of the training data. However, combining these base learners can produce a strong one, as the bias or variance is reduced by their combination.

Ensemble techniques are often illustrated using decision trees but this technique may be prone to overfitting (high variance and low bias) when its pruning has not been performed, and it can also lend itself to underfitting (low variance and high bias) when it is very small like a decision stump, which is a decision tree with one level. It is important to note that a learner cannot generalise well to new or unexplored datasets when the training data is either overfitted or underfitted. Ensemble approaches are used to prevent this behavior and enable the learner to generalise to fresh training samples. Whereas decision trees can exhibit high variance or high bias, it is worth noting that it is not only the modelling technique that leverages ensemble learning to find the "sweet spot" within the tradeoff of bias variance. Thus, two main approaches are considered when selecting base learners:

The first approach is creating a single base learner or learning algorithm to create a set of homogeneous base learners that are learned with different techniques resulting in a homogeneous ensemble model. Homogeneous ensemble classifiers pool the predictions of multiple individual learners. The category of this ensemble learning can broadly be

highlighted in two techniques; bagging and boosting. The second approach is to employ different types of base learners or learning algorithms, forming a set of heterogeneous base learners to create a heterogeneous ensemble model, used in the stacking method. In general, three major meta-algorithms that provide effective techniques for combining base or weak learners can be considered (Singh *et al.*, 2020):

Bagging

Bagging, also known as Bootstrap Aggregation, is an ensemble ML technique. The idea of drawing at random, dataset with replacement and then using these different random subsets of the data to learn independent or different classifiers is what is called bootstrapping. If this technique is used to combine individual classifiers, this process is called bagging. Therefore, bagging is the process of building each classifier or tree using a unique random subset of the data that was drawn via replacement. The final prediction might be created by averaging (regression) or majority voting (classification) the predictions from each independent classifier. Random Forest is a frequently used algorithm or approach. Both feature selection and the training of the classification model in this work use the RF approach. A technique for ensemble learning called RF aims to simplify models that overfit the training set. Unlike random forest, which is an extension of bagging and also randomly selects subsets of features used in each data sample, bagging is an ensemble algorithm that fits multiple models on various subsets of a training dataset before combining the predictions from all models (Singh *et al.*, 2020; Breiman, 2001).

Boosting

Boosting is a homogeneous ensemble learning that ensembles weak decision tree learners into a robust one by minimising bias. Here, learning is sequential, where any weaknesses of predecessor learners are compensated by each successor learner in an ensemble. At every iteration, there is a combination of weak rules from every learner to produce a single robust prediction rule. Boosting techniques are focused on three popular methods including Adaptive Boosting (AdaBoost), Gradient Boosting (GradientBoost), and Extreme Gradient Boosting (XGBoost). For the study, GradientBoost or GBM is discussed and used for both feature selection and learning a classification model (Singh *et al.*, 2020).

The primary distinction between the two main forms of ensemble learning approaches, boosting and bagging, is the technique for which they are trained. Weak learners are learned

sequentially in boosting as opposed to parallel learning in bagging. This merely explains how a succession of learners is created, and how the weights of the misclassification data in each successive learner are enhanced. The learner can determine the factors it has to concentrate on to increase performance accuracy with the help of this redistribution of weights (Singh *et al.*, 2020).

Stacking

The stacking technique combines a heterogeneous of multiple base learners into a more robust one than base learners. This technique combines the predictions of different individual base learners to make a final robust prediction. Where weak or base learning algorithms are rightfully blended, a meta-learner with lower variance and bias can be developed (Singh *et al.*, 2020). The use of this ensemble learning has been observed to yield more robust outcomes in various studies in which they have been applied (Zang *et al.*, 2014; Cai *et al.*, 2019; Ragunthar and Selvakumar, 2019; Warsinske, *et al.*, 2019). By training a final meta-learner on the output predictions given by many base learners, base learners are simultaneously learnt and integrated. When the models being fit disagree, ensembles frequently perform better. Additionally, the idea of integrating several models seems to work well in practice, frequently outperforming single method implementations. Cross-validation is used in stacking to gauge the effectiveness of various base learning algorithms (Gremmell, 2018). The meta-learning method (Singh *et al.*, 2020) takes as input the output from the base learners, which is referred to as "level-one" data in the stacking technique (Wolpert, 1992). Table 3.3 (Zhou, 2012) provides an illustration of how stacking typically learns through the three basic processes listed below:

Step 1: Learn first-level classifiers based on the original training data set.

Step 2: Construct a new data set based on the output of base classifiers. It is assumed that for each example in D_s as $\{\mathbf{x}_i, y_i\}$, the corresponding example $\{\mathbf{x}'_i, y_i\}$ in the new dataset is constructed, where $\mathbf{x}'_i = \{h_1(\mathbf{x}_i), h_2(\mathbf{x}_i), \dots, h_T(\mathbf{x}_i)\}$.

Step 3: Learn a second-level classifier based on the newly constructed data set. For an unseen example \mathbf{x} , its predicted class label of stacking is $h'(\mathbf{x}) = h'(h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_T(\mathbf{x}))$, where $\{h_1, h_2, \dots, h_T\}$ are first-level classifiers and h' is the second-level classifier.

Table 3.3 Stacking Algorithm

Input	Dataset $D = \{\mathbf{x}_i, y_i\}_{i=1}^n$ ($\mathbf{x}_i \in \mathbb{R}^m, y_i \in Y$)
Output	A stacked ensemble classifier model H
1	Step 1: Learn first-level classifiers
2	for $t \leftarrow 1$ to T do
3	Learn a base classifier h_t based on D
4	end for
5	Step 2: Construct new data sets from D
6	for $i \leftarrow 1$ to n do
7	Construct a new data set that contains $\{\mathbf{x}'_i, y_i\}$, where $\mathbf{x}'_i = \{h_1(\mathbf{x}_i), h_2(\mathbf{x}_i), \dots, h_T(\mathbf{x}_i)\}$.
8	end for
9	Step 3: Learn a second-level classifier
10	Learn a new classifier h' based on the newly constructed data set
11	return $H(\mathbf{x}) = h'(h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_T(\mathbf{x}))$

3.3.2 Base Learners (Classifiers)

Base learners are those who make up an ensemble's component or individual learners and who are strategically blended. While firmly avoiding over-fitting, the base (weak) learner concentrates on accurately categorising the examples with the highest weights. For the purposes of this study, five base learners are taken into account, as follows:

Random Forest (RF) Classifier

Decision trees in the Bagging family are combined to form Random Forest. When building a base classifier, RF uses a variety of decision tree algorithms. Aside from the bootstrap sampling and majority voting used in bagging, RF integrates random feature space selection into the generation of learning sets to promote the diversity of base classifiers. The algorithm on Table 3.4 specifically describes the general rule of RF algorithm:

Table 3.4 Random Forest Algorithm

Input	Dataset $D = \{\mathbf{x}_i, y_i\}_{i=1}^n$ ($\mathbf{x}_i \in \mathbb{R}^n, y_i \in Y$)
Output	A stacked ensemble classifier model H
1	for $t \leftarrow 1$ to T do
2	Construct a Bootstrap dataset D_t by randomly sampling with replacement in D
3	Learn a decision tree h_t by applying <i>LearnDecisionTree(data=D_t iteration = 0, ParentNode = root):</i>
4	If stop criterion is satisfied, return
5	Randomly sample features in the whole feature space \mathbb{R}^n to get a new data set $\hat{D}_{current} = RandomSubset(D_{current})$
6	Find the best feature q^* according to impurity gain
7	Split data $(D_L, D_R) = split(D_{current}, q^*)$
8	Label the new parent node $v = parent.newchild(q^*)$
9	Conduct <i>LearnDecisionTree(D_L, iteration = iteration + 1, ParentNode = v)</i> and <i>LearnDecisionTree(D_R, iteration = iteration + 1, ParentNode = v)</i>
10	end for
11	return $H(\mathbf{x}) = \operatorname{argmax}_{y \in Y} \sum_{t=1}^T 1(h_t(\mathbf{x}) = y)$

Gradient Boosting Machine (GBM) Classifier

A machine learning method called gradient boosting is applied to tasks like classification and regression, among others. Gradient Boosting Classifier is used when the problem is a classification problem and Gradient Boosting Regressor when the label is continuous. According to Piryonesi and El-Diraby (2020) and Hastie *et al.* (2009), it provides a prediction model in the form of an ensemble of weak decision tree learners. The "Loss function" is the only distinction between the two. By employing gradient descent to add weak learners, the goal is to reduce this loss function. Since it is based on a loss function, alternative loss functions, such as Mean Squared Error (MSE), can be employed for regression problems. For classification tasks, different loss functions, such as log-likelihood, are used. Similar to other boosting methods, the gradient boosted trees methodology is constructed stage-by-stage, but it generalizes the other algorithms by enabling optimization for any arbitrary differentiable loss function. It frequently offers

predicted scores that are much higher than those of other algorithms and can manage missing data without the need for imputation. Despite its benefits, it has the following drawbacks: it might take a while to train and is sensitive to outliers and overfitting if there are too many trees. Table 3.5 below illustrates the algorithm of GBM.

Table 3.5 Gradient Boosting Algorithm

Input	Dataset $D = \{\mathbf{x}_i, y_i\}_{i=1}^n$ ($\mathbf{x}_i \in \mathbb{R}^n, y_i \in \{+1, -1\}$)
Output	A stacked ensemble classifier model H
1	Weight initialisation distribution W_1
2	for $t \leftarrow 1$ to T do
3	Learn base classifier h_t on D and W_t
4	Evaluation of base classifier $\varepsilon(h_t)$
5	Updating distribution of weight W_{t+1} based on $\varepsilon(h_t)$
6	end for
7	return $H = combination(\{h_1, \dots, h_T\})$

Naïve Bayes (NB) Classifier

The simplest Bayesian classifier, known as the naive Bayes classifier, has grown into a significant probabilistic model and has proved astonishingly successful in practice despite its high independence assumption. The naive Bayes model is a hypothetical example of a conditional probability model. Considering a problem instance represented by a vector $\mathbf{x} = (x_1, \dots, x_n)$ to be classified, where there are n features (independent features), it assigns to this instance probabilities $p(C_k | x_1, \dots, x_n)$ for each of K possible outcomes or classes C_k (Narasimha and Susheela 2011).

The problem with the above formulation is that building such a model using probability tables is impractical if there are many characteristics (n) or if each feature has a wide range of possible values. In order to make the model more manageable, it must be reformulated. The conditional probability can be broken down as follows using Bayes' theorem gives as Equation (3.1).

$$p(C_k | \mathbf{x}) = \frac{p(C_k) p(\mathbf{x} | C_k)}{p(\mathbf{x})} \quad (3.1)$$

Using Bayesian probability terminology, Equation (3.1) can be written as Equation (3.2).

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}} \quad (3.2)$$

Since the denominator of that fraction does not depend on C and the values of the characteristics x_i are known, the denominator is virtually constant in practice, and only the numerator is of relevance. The joint probability model serves as the numerator's equivalent. $p(C_k, x_1, \dots, x_n)$, and using the chain rule for repeated applications of conditional probability, this may be written as Equation (3.3).

$$\begin{aligned} p(C_k, x_1, \dots, x_n) &= p(x_1, \dots, x_n, C_k) \\ &= p(x_1 | x_2, \dots, x_n, C_k) p(x_2, \dots, x_n, C_k) \\ &= p(x_1 | x_2, \dots, x_n, C_k) p(x_2 | x_3, \dots, x_n, C_k) p(x_3, \dots, x_n, C_k) \\ &= \dots \\ &= p(x_1 | x_2, \dots, x_n, C_k) p(x_2 | x_3, \dots, x_n, C_k) \\ &\quad \dots p(x_{n-1} | x_n, C_k) p(x_n | C_k) p(C_k) \end{aligned} \quad (3.3)$$

Now the "naïve" conditional independence assumptions come into play. Assume that all features in \mathbf{x} are mutually independent, conditional on the category C_k . Under this assumption,

$$p(x_i | x_{i+1}, \dots, x_n, C_k) = p(x_i | C_k)$$

and can be written as Equation (3.4).

$$\Rightarrow p(x_1, \dots, x_n | C_k) = \prod_{i=1}^n p(x_i | C_k) \quad (3.4)$$

Thus, the joint model can be expressed as Equation (3.5).

$$\begin{aligned}
p(C_k|x_1, \dots, x_n) &\propto p(C_k, x_1, \dots, x_n) \\
&\propto p(C_k)p(x_1|C_k)p(x_2|C_k)p(x_3|C_k) \dots \\
&\propto p(C_k) \prod_{i=1}^n p(x_i|C_k)
\end{aligned} \tag{3.5}$$

where, \propto denotes proportionality.

This means that under the above independence assumptions, the conditional distribution over the class variable C is given as Equation (3.6).

$$p(C_k|x_1, \dots, x_n) = \frac{1}{Z} p(C_k) \prod_{i=1}^n p(x_i|C_k) \tag{3.6}$$

where the evidence $Z = p(\mathbf{x})$ is given as Equation (3.7).

$$Z = p(\mathbf{x}) = \sum_k p(C_k) p(\mathbf{x}|C_k) \tag{3.7}$$

Where Equation (3.7) is a scaling factor dependent only on x_1, \dots, x_n , that is, a constant if the values of the feature variables are known (Narasimha and Susheela 2011).

Now, based on the maximum a posteriori (*MAP*) decision rule, the corresponding Bayes classifier, a function that assigns a class label $\hat{y} = C_k$ for some k is given as Equation (3.8).

$$\hat{y} = \arg \max_{k \in \{1, \dots, K\}} p(C_k) \prod_{i=1}^n p(x_i|C_k) \tag{3.8}$$

where \hat{y} is the \mathbf{x} class estimated based on its features x_1, \dots, x_n .

Recurrence HNSCC classification: Here is an illustration of an HNSCC classification problem using naive Bayesian classification. This is about the challenge of categorising the prognosis of HNSCC into recurrence and non-recurrence. Remember that the HNSCC prognoses are derived from a variety of classes of HNSCC that can be treated as sets of prognoses, and that the independent probability that the i -th prognosis of a specific HNSCC happens in an HNSCC patient D from classes C can be written as

$$p(x_i|C)$$

Thus, the probability that a given HNSCC contains all the prognosis x_i , given a class C is given as Equation (3.9).

$$p(D|C) = \prod_i P(x_i|C) \quad (3.9)$$

The question here is: “what is the probability that a given HNSCC patient D belongs to class C ?” In other words, what is $p(C|D)$? By the definition, this is given as Equation (3.10).

$$p(D|C) = \frac{p(D \cap C)}{p(C)} \quad (3.10)$$

and $p(C|D)$ is also given as Equation (3.11).

$$p(C|D) = \frac{p(D \cap C)}{p(D)} \quad (3.11)$$

By Bayes’ theorem, the results of the equations are resolved into a statement of probability in terms of likelihood given as Equation (3.12).

$$p(C|D) = \frac{p(C)p(D|C)}{p(D)} \quad (3.12)$$

Now, assume that there are only two mutually exclusive classes, r (recurrence) and $\neg r$ (non-recurrence), such that every feature or prognosis is in either one or the other of Equation (3.13).

$$p(D|r) = \prod_i P(x_i|r) \quad (3.13)$$

and equation (3.14).

$$p(D|\neg r) = \prod_i P(x_i|\neg r) \quad (3.14)$$

Using the Bayesian result above, these can be written as Equation (3.15)

$$p(r|D) = \frac{P(r)}{p(D)} \prod_i P(x_i|r) \quad (3.15)$$

and Equation (3.16).

$$p(-r|D) = \frac{P(-r)}{p(D)} \prod_i P(x_i|-r) \quad (3.16)$$

Dividing Equation (3.15) by Equation (3.16) gives:

$$\frac{p(r|D)}{p(-r|D)} = \frac{p(r) \prod_i P(x_i|r)}{p(-r) \prod_i P(x_i|-r)}$$

By re-factorisation, one obtains the equation below:

$$\frac{p(r|D)}{p(-r|D)} = \frac{p(r)}{p(-r)} \prod_i \frac{P(x_i|r)}{P(x_i|-r)}$$

Thus, the probability ratio $p(r|D)/p(-r|D)$ can be expressed in terms of a series of likelihood ratios. The actual probability $p(r|D)$ can be easily computed from $\log p(r|D)/p(-r|D)$ based on the observation that $p(r|D) + p(-r|D) = 1$. Taking the logarithm of all these ratios, this can be obtained as Equation (3.17).

$$\ln \frac{p(r|D)}{p(-r|D)} = \ln \frac{p(r)}{p(-r)} + \sum_i \ln \frac{P(x_i|r)}{P(x_i|-r)} \quad (3.17)$$

This technique of log-likelihood ratios is a common technique in statistics. A situation where there are two mutually exclusive alternatives as this example, the conversion of a log-likelihood ratio to a probability takes the form of a sigmoid curve. Finally, the HNSCC patient can be classified as follows:

It is a recurrence if $p(r|D) > p(-r|D)$ that is,

$$\ln \frac{p(r|D)}{p(-r|D)} > 0,$$

Otherwise, it is non-recurrence.

Equivalently,

$$\begin{cases} \ln \frac{p(r|D)}{p(\neg r|D)} > 0, & \text{recurrence} \\ < 0, & \text{othrewise} \end{cases}$$

Despite what appear to be oversimplified assumptions, naive Bayes classifiers have performed admirably in a variety of real-world applications, most notably the categorization of cancer and the filtering of spam and documents. To estimate the essential parameters, they only need a limited number of training instances. When compared to more complex methods, naive Bayes classifiers can be very quick. Each distribution can be individually estimated as a one-dimensional distribution due to the decoupling of the class conditional feature distribution. This in turn aids in resolving issues brought on by the dimensionality curse.

Deep Neural Network (DNN) Classifier

Deep neural networks (DNNs) are enhanced versions of multi-layered, ordinary ANNs. Due to their outstanding ability to learn both the underlying structure of the input data vectors and the nonlinear input-output mapping, DNN models have recently gained a lot of attention. Neural Networks (NNs) have vastly been used in cancer research over the last decades. It has been shown that neural network analysis is particularly suitable in situations for which there is ill defined task to be solved, and algorithmic solution development is difficult. This situation is exactly that which applies to cancer data analysis, which requires a highly nonlinear approach to computation (Zheng *et al.*, 2022).

Thus, a computational model of an artificial neuron is one that is influenced by biological neurons. Dendrites, cell bodies, axons, and synapses (output dendrites coupled to the dendrites of neighboring neurons) make up a neuron system. Signals are sent to the biological neurons through synapses located on the dendrites as shown in Figure 3.4 (Chattopadhyay and Guha, 2004).

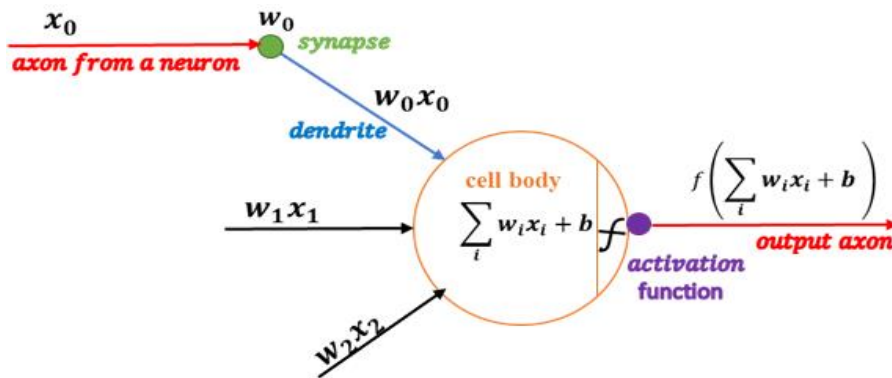


Figure 3.4 Biologically inspired Neural Network (Karparthy, 2016)

The Architecture of ANN can be explained as the neurons (nodes) in the network that are connected in layers (input and output layers) by edges. According to a specific pattern, the neurons in various levels are connected to one another (Munakata, 2008). Neural network can be categorised into subgroups based on the connections of layers: single-layer network (single-layer perceptron) and multi-layer network (multi-layer perceptron). Mostly, the neural network may consist of at least one middle layer termed as hidden layer(s) (Abdul-Kareem *et al.*, 2001).

Single-Layer Perceptron Feedforward Neural Networks: This ANN has only two layers: input layer and output layer without any hidden layer. Neurons are put in layers, where the outputs of neurons from one layer are connected to the inputs of neurons from the next layer as shown in Figure 3.5.

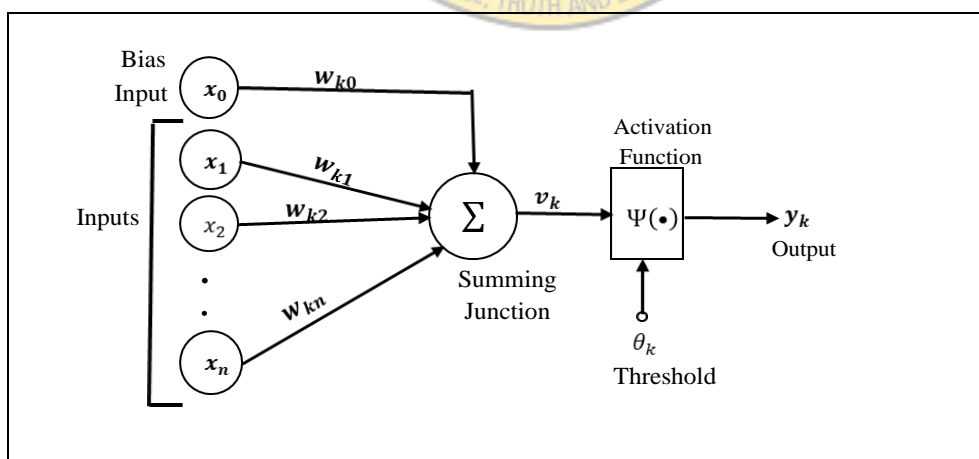


Figure 3.5 An ANN Model Architecture

Let (x_1, \dots, x_n) be the inputs and (w_{k1}, \dots, w_{kn}) be the corresponding weight, and still w_{k0} denote the bias in the learning process. Weight is assigned to every input in the layer

and their product $\mathbf{w}\mathbf{x}$ are obtained as weighted input. Add $\mathbf{w}\mathbf{x}$ and w_{k0} to produce the n input. Sum the weighted inputs to produce v_k parameter represented as Equation (3.18).

$$v_k = \sum_{l=0}^n w_{kl}x_l \quad (3.18)$$

Add v_k to the bias x_0 , and pass the outcome via the $\Psi(\cdot)$, to obtain Equation (3.19)

$$\Psi\left(x_0 + \sum_{l=0}^n w_{kl}x_l\right) = \hat{y}_k \quad (3.19)$$

which is produced as the output (Beale and Hagan, 2012; Munataka, 2008; Graupe, 2007). The three activation functions that are most frequently employed to regulate the signal input are the *linear transfer function*, *tangent-sigmoid function*, and *log-sigmoid function*. These activation functions are chosen by trial and error based on the specific problem at hand.

Multilayer Feedforward Perceptron Neural Networks: The Multilayer Perceptron (MLP) feedforward neural network, which is taken into consideration in this section and is depicted in Figure 3.6, is the most generally and often utilised ANN paradigm in many real-life applications. Let y_i denote the label/target variable (quantitative or qualitative) of the i^{th} patient and let $q_i = \{q_{ij}\}$ denote a vector of inputs (features or predictor variables) or covariate of any measured for each i^{th} patient. Assume that the hidden layer has P number of neurons. The input into neuron k ($k = 1, 2, \dots, P$) prior to activation, is the linear function $w'_k q_{ij}$ where $w'_k = \{w_{kj}\}$ a vector of unknown connection weights peculiar to the k^{th} neuron, including a bias. In the hidden layer, each neuron performs weighted inputs summation (n_i) prior to activation, then it is passed to a nonlinear activation function given below

$$f_k\left(b_k^{(1)} + \sum_{j=1}^q w_{kj}q_j\right)$$

selecting hyperbolic tangent transformation as the hidden layer's activation function $f(x_i) = (e^{x_i} - e^{-x_i}) / (e^{x_i} + e^{-x_i})$ being the emission of neuron for input features (Okut *et al.*, 2014).

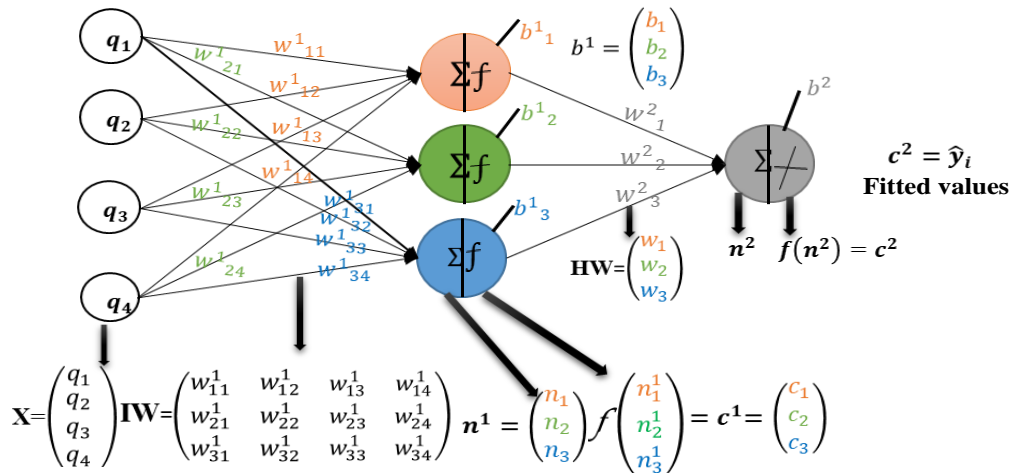


Figure 3.6 A Multi-layer Perceptron Feedforward Model

The Figure 3.6 is a designed ANN with 4 inputs (q_i), where each input has up to 3 connections with the neurons in a hidden layer via coefficients $w^{(l)}_{kj}$ (the l^{th} layer, j^{th} neuron, and k^{th} input feature) with each hidden and output neuron having a bias parameter $b_j^{(l)}$. Now, \mathbf{X} = inputs, \mathbf{IW} = weights from input layer to hidden layer (being 12 weights), with hidden layer biases b^1 (3 biases), \mathbf{HW} = weights from hidden to output layer (3 weights), with output layer biases b^2 (1 bias). So that, $n^1 = \mathbf{IW}\mathbf{X} + b^1$ is the weighted input summation of the first layer, with $c^1 = f(n^1)$ as the output of the second/hidden layer, $n^2 = \mathbf{HW}c^1 + b^2$ is the weighted input summation of the hidden layer, with $\hat{y} = c^2 = f(n^2)$ as the predicted output of the network. This ANN has $12+3+3+1= 19$ total number of parameters.

Outputs from neurons of the hidden layer become inputs to the neurons of the next layer. Here, after the activation function in the neuron of the hidden layer, outputs from these neurons are sent as inputs to the neuron of the output layer with weighted summation is given as below:

$$\sum_{k=1}^p w_k^i f_k \left(b_k^{(1)} + \sum_{j=1}^q w_{kj} q_j \right) + b^{(2)},$$

where, $w_k = \{w_{kj}\}$ is the vector of unknown strengths of connections for k^{th} neuron, including a bias; w_k are specific weights to j^{th} neuron, and $b^{(1)}$ and $b^{(2)}$ are respectively the parameters of biases in the hidden and output layers. Finally, given the same or another

activation function $h(\cdot)$, the label/target variable in the training set can be predicted as Equation (3.20)

$$h \left[\sum_{k=1}^p w_k^i f_k(\cdot) + b^{(2)} \right] = c^2 = \hat{y} \quad (3.20)$$

The predicted output \hat{y}_i value from the output layer in Equation (3.20) as shown in Figure 3.6 can then be estimated as Equation (3.21).

$$\hat{y} = h \left[\left(\left(\frac{e^{(b_1^1 + w_{11}^1 q_1 + w_{12}^1 q_2 + w_{13}^1 q_3 + w_{14}^1 q_4)} - e^{(-b_1^1 - w_{11}^1 q_1 - w_{12}^1 q_2 - w_{13}^1 q_3 - w_{14}^1 q_4)}}{e^{(b_1^1 + w_{11}^1 q_1 + w_{12}^1 q_2 + w_{13}^1 q_3 + w_{14}^1 q_4)} + e^{(-b_1^1 - w_{11}^1 q_1 - w_{12}^1 q_2 - w_{13}^1 q_3 - w_{14}^1 q_4)}} \right) w_1^2 + \left(\frac{e^{(b_2^1 + w_{21}^1 q_1 + w_{22}^1 q_2 + w_{23}^1 q_3 + w_{24}^1 q_4)} - e^{(-b_2^1 - w_{21}^1 q_1 - w_{22}^1 q_2 - w_{23}^1 q_3 - w_{24}^1 q_4)}}{e^{(b_2^1 + w_{21}^1 q_1 + w_{22}^1 q_2 + w_{23}^1 q_3 + w_{24}^1 q_4)} + e^{(-b_2^1 - w_{21}^1 q_1 - w_{22}^1 q_2 - w_{23}^1 q_3 - w_{24}^1 q_4)}} \right) w_2^2 + \left(\frac{e^{(b_3^1 + w_{31}^1 q_1 + w_{32}^1 q_2 + w_{33}^1 q_3 + w_{34}^1 q_4)} - e^{(-b_3^1 - w_{31}^1 q_1 - w_{32}^1 q_2 - w_{33}^1 q_3 - w_{34}^1 q_4)}}{e^{(b_3^1 + w_{31}^1 q_1 + w_{32}^1 q_2 + w_{33}^1 q_3 + w_{34}^1 q_4)} + e^{(-b_3^1 - w_{31}^1 q_1 - w_{32}^1 q_2 - w_{33}^1 q_3 - w_{34}^1 q_4)}} \right) w_3^2 \right) + b^{(2)} \right] \quad (3.21)$$

Generalised Linear Model (GLM) Classifier

The Generalised Linear Model (GLM) specifically the logistic regression (LR) with its weighting system (i.e., β -coefficients) is another popular binary classification model which also performs feature selection. To establish comparability between the various ranges of feature values, a Z-transformation is carried out as a preprocessing step. The essential features are represented by the β -coefficients of the derived regression model, which acts as an importance measure (Neumann *et al.*, 2017) is given by Equation (3.22)

$$\ln \left[\frac{p(y)}{1 - p(y)} \right] = \ln(\text{odds}) = a + b_1 x_1 + b_2 x_2 + \dots + b_n x_n \quad (3.22)$$

In terms of p as indicated in equation (3.23)

$$p(y) = \frac{e^{(a + b_1 x_1 + b_2 x_2 + \dots + b_n x_n)}}{1 + e^{(a + b_1 x_1 + b_2 x_2 + \dots + b_n x_n)}} \quad (3.23)$$

where, $\ln[p(y)/1 - p(y)] =$ natural log of odds (logit),

$p(y) =$ event probability,

$x_1, x_2, \dots, x_n =$ training features,

$a = y$ intercept,

$b_1, b_2, \dots, b_n =$ gradient

3.4 Feature Selection Stanzas Ensemble Feature Selection

In some situations, FS techniques may provide instability and unreliability results based on several reasons; such as the nature of the complexity of multiple relevant features for the dataset with high dimensionality (Pyka *et al.*, 2013; Dybowski *et al.*, 2010; He and Yu, 2010). It has been shown by some previous researchers that there is no single optimal FS technique (Yang *et al.*, 2004). Gini-coefficient which is largely used in medical predictions (Llorca and Delgado-Rodríguez, 2002) has been proven to yield unstable results when given unbalanced training data (Boulesteix *et al.*, 2012; Sandri and Zuccolotto, 2008). To correct the unreliable and unstable instability of FS techniques in training ML algorithms, a hybrid Ensemble Feature Selection (EFS) technique is proposed in regard to the ensemble learning intuition. The EFS combines multiple FS methods and their normalised outputs to quantitative ensemble importance; and thus, reimburses biases of individual FS techniques.

3.4.1 The Feature Selection Technique

Ensemble approaches may be used to increase the robustness of feature selection strategies, just like in the case of supervised learning. In fact, it is frequently noted that numerous different feature subsets may or may not produce equally optimal results in big feature/small sample size domains (Saeys *et al.*, 2007), and EFS may lessen the chance of selecting an unstable subset. The optimal subset or ranking of features may be approximated more accurately by EFS than by other feature selection methods, which itself may produce feature subsets that can be regarded as local optima in the space of feature subsets. Last but not least, a feature selector's representational strength could limit the search space, making it impossible to find optimal subsets. By combining the results of various feature selectors, ensemble feature selection may help to solve this issue.

Gradient Boosting Machine Feature Selection (GBM-FS) employs a Weighted Feature Importance (WFI) metric that is robust enough to receive the scores of each attribute according to importance after the boosted tree is constructed. The importance of each attribute is provided by the model that scores their importance, by its decision-making while decision trees are being constructed. In general, feature importance assigns a score to each

attribute defining its significance role. This feature importance is computed explicitly, where features are compared with one another and ranked in the dataset. This rank of features by each decision tree forms a feature subset of each tree model. The quantity of each attribute split point is used to determine each decision tree's relevance, which is then weighted by the quantity of observations coming from that node. The algorithm's effectiveness and performance are enhanced by using this split. (Upadhyay *et al.*, 2021).

Purity (Gini Index) is specifically used to choose the split points or to determine a more precise error function. An ensemble model of these decision trees aggregates the feature importance of each tree across all individual decision trees. The most promising characteristics from a dataset are utilized to create subsets using the model-based feature selection class. This method emphasizes using WFI to incorporate the preprocessing with the model, reducing the amount of training features by deleting redundant and unnecessary features from the dataset, and lengthening training times (Upadhyay *et al.*, 2021).

3.5 V-Fold Cross-Validation Technique

The most significant and popular automatic model complexity optimisation technique is cross-validation. It fixes the overfitting issue. Blocks of v identical sizes make up the entire data collection \mathbf{X} . After that, the algorithm is tested on the last block after being learned v times on $v - 1$ blocks. The computed errors from V are averaged, and the value of λ_{opt} with the lowest mean error is selected to train the final model on the entire set of data \mathbf{X} . This is illustrated as algorithm for V-fold cross-validation in Figure 3.7.

<p><i>V – Fold Cross Validation</i>(\mathbf{X}, v) Divide learning set into v equally sized blocks $\mathbf{X} = \mathbf{X}_1, \dots, \mathbf{X}_v$ For all $\lambda \in \{\lambda_{min}, \dots, \lambda_{max}\}$ For all $i \in \{1, \dots, v\}$ For each v, train a model of complexity λ on \mathbf{X}/\mathbf{X}_i Compute the error $E(\lambda, \mathbf{X}_i)$ on the test set \mathbf{X}_i Compute the mean error $E(\lambda) = \frac{1}{v} \sum_{i=1}^v E(\lambda, \mathbf{X}_i)$ Choose the value $\lambda_{opt} = \operatorname{argmin}_{\lambda} E(\lambda)$ with smallest mean error Train the final model with complexity λ_{opt} on the entire learning set \mathbf{X}</p>

Figure 3.7 Algorithm for V-Fold Cross-Validation (Yaliang *et al.*, 2015)

3.6 Detecting Multicollinearity using Variance Inflation Factor

As the name suggests, a Variance Inflation Factor (VIF) quantifies how much the variance of an independent feature is inflated by its interaction or correlation with the other predictors in the dataset. If the VIF is 1, then there is no connection between the j th predictor and the other predictor variables, hence the variance of the j th coefficient is not inflated in any way. According to the common rule of thumb, VIF greater than 5 calls for additional examination, whereas VIF greater than 10 indicate substantial multicollinearity that needs to be corrected. The variance inflation factor for the j th predictor is specifically given as Equation (3.24).

$$\text{VIF}_j = \frac{1}{1 - R_j^2} \quad (3.24)$$

where R_j^2 is the R-square value obtained by regressing the j -th predictor on the remaining predictors.

3.7 Good Fit Learning Curves

Goodness-of-fit learning curves is one of the core parts of any machine learning algorithm existing between model overfitting and underfitting. A good fit is achieved using the training loss on training dataset and validation/testing loss on validation dataset, both of which decrease to a point of stability leaving a negligible gap called “*generalisation gap*” between the values of the two final losses. The training loss value is almost always lower as compared to the validation loss value. Continued training of a good fit is likely to result into overfitting. Thus, good fit learning curves is achieved if

- (i) There is a decrease of training loss curve to a point of stability.
- (ii) There is a decrease of validation/testing loss to a point of stability leaving a small gap with the training loss.

3.8 Model Evaluation Measurements, Validation and Comparison

In evaluating the performance of the model, the predictions of that model are compared to the actual labels on set of examples. When the set of examples has been used to train the model, then the performance measurement is effectively on the training set. Meanwhile, if this set of (unseen or held-out) examples has not been employed to learn the model, then the performance measurement is on the test set. Classification predictive learning identifies the

class to which a given instance belongs. The classifier for binary classification task has two classes $\{-1, 1\}$ or $\{0, 1\}$ from which the classifier model chooses (Dom *et al.*, 2008). The following metrics; accuracy, recall, specificity, precision, F1- score, Area Under Receiver Operating Characteristic (AU-ROC) curve, and logarithmic loss (log-loss) are considered under this study for model evaluation purpose.

Predicting a patient with recurrence as recurrence signifies a true positive (TP), while predicting a patient with recurrence as nonrecurrence signifies a false negative (FN). Similarly, predicting a patient with nonrecurrence as recurrence is a false positive (FP), whereas predicting a patient with nonrecurrence as nonrecurrence is a true negative (TN). Table 3.6 below displays the confusion matrix table for the prognosis of HNSCC recurrence.

Table 3.6 Confusion Matrix for Prognosis of HNSCC Recurrence

		Actual conditions	
		Recur (+)	Nonrecur (-)
Predicted outcomes	Recur (+)	True positive (TP)	False positive (FP)
	Nonrecur (-)	False negative (FN)	True negative (TN)

Recur: Recurrence, Nonrecur: Nonrecurrence

Classification accuracy is simply the expression of correct predictions as a percentage of the total predictions and is represented as Equation (3.25).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \times 100\% \quad (3.25)$$

Specificity is the true negative rate that measures the ratio of true negative conditions to all patients with non-recurrence. The probability of classifying a patient as non-recurrent when actually he is non-recurrent is represented as Equation (3.26).

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \times 100\% \quad (3.26)$$

The probability of classifying a patient as a recurrent when actually he is a non-recurrent is 1-specificity.

Precision value is the expression of correct positive predictions as a percentage of all correct or incorrect positive predictions is represented as Equation (3.27).

$$\text{Precision (P)} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\% \quad (3.27)$$

Recall or sensitivity is the expression of correct positive predictions as a percentage of all predictions that are actually positive. In this context, it is the expression of true positive conditions as a percentage of all the patients with recurrence. The probability of classifying a patient as a recurrent when actually he is a recurrent is represented as Equation (3.28).

$$\text{Recall (R)} = \frac{\text{TP}}{\text{TP}+\text{FN}} \times 100\% \quad (3.28)$$

It may be preferable to prioritise recall or precision more highly depending on the application. The need of recall and precision is crucial for many applications in the meanwhile. One popular metric that combines precision and recall into a single metric is called F1-Score, which is computed as the harmonic mean of precision and recall as Equation (3.31).

$$F_1 = \left(\frac{\text{Recall} + \text{Precision}}{2} \right)^{-1} \quad (3.29)$$

$$= \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}} \quad (3.30)$$

$$= 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (3.31)$$

The Receiver Operating Characteristic (ROC) curve plots the sensitivity as against the specificity aimed at threshold values changing. When all the examples are classified as negative, it means there is the highest value of threshold. So, the true negative rate is 1.0 (100%) and the true positive rate is 0.0 (0%). Contrary, when all the examples are classified as positive, then the value of the threshold is at the lowest. So, the true positive rate is 1.0 (100%) and the true negative rate is 0.0 (0%). The ROC curve can be viewed as single value called the Area Under the Curve (AUC). AUC calculates the area under the ROC curve. A single value that falls within [0, 1] interval. Better is the prediction as larger is the area under the curve (Adbul-Kareem, 2001).

3.8.1 Binary Cross-Entropy/Logarithmic Loss

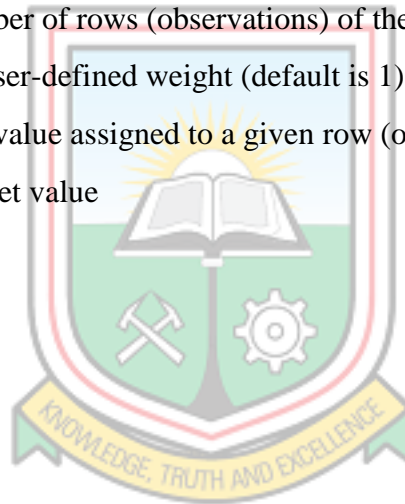
The Logarithmic loss (log-loss) metric can be used to evaluate the performance of the binomial or multinomial classifier. The negative average of the log of the corrected projected probability for each case is referred to as log loss. In contrast to AUC, which measures a model's ability to correctly categorise a binary target, log-loss measures how

closely a model's predicted values match the actual/true value (in the case of binary classification, either 0 or 1). In other words, it gauges the degree of uncertainty in anticipated labels depending on how much they deviate from the actual labels. The higher the log loss value, the more the predicted probability deviates from the actual value. Thus, better predictions are indicated by a smaller log-loss value. When the model output provides the likelihood of a binary result, log-loss is a suitable performance metric. (Megha, 2020). The log-loss equation for binary classification is given as Equation (3.32).

$$\text{Log - loss} = -\frac{1}{N} \sum_{i=1}^N w_i (y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i)) \quad (3.32)$$

where:

- N is the total number of rows (observations) of the corresponding data frame
- w is the per row user-defined weight (default is 1)
- p is the predicted value assigned to a given row (observation)
- y is the actual target value



CHAPTER 4

DEVELOPMENT OF HYBRID RECURRENT HEAD AND NECK SQUAMOUS CELL CARCINOMA PROGNOSTIC MODEL

4.1 Overview

This chapter confers the developing of a hybrid ensemble classification system for recurrent HNSCC prognosis. It specifies the techniques used to develop the hybrid stacked ensemble classification model. The base classifiers (GBM, DRF, DNN, GLM, and NB) are the frequent effective ML techniques that are mostly applied extensively to cancer study. In medical studies, different ML techniques including classification, clustering, and regression analysis are needed. The most common technique in ML that is used to classify and predict the predefined classes or categories of labels is the classification (supervised learning). The present study makes use of five supervised ML techniques in a stacking ensemble to identify, classify and predict the categories of the label HNSCC with levels of individual recurrence patterns (recurrence verses nonrecurrence).

4.2 Development of Recurrent HNSCC Prognostic (HESCA) Model

The proposed Hybrid Ensemble Super Classification Algorithm (HESCA) model is by a conjunction of five base classifiers (GBM, DRF, DNN, GLM, and NB), with 10-fold cross-validation (10-CV), with GBM ensemble feature selection (GBM-FS). The HESCA model learning technique consists of a combination of two main components: one of these components is the *ensemble of supervised feature selection algorithm*; and the other component is the *ensemble of supervised machine learning classification algorithms*, with *10-Cross-Validation*. The proposed HESCA model is a stacked ensemble model having all the individual base models and the GBM meta-model in a stacking ensemble (with 10-CV) based on the optimal feature subset (gradient boosted features) provided by the ensemble feature selection technique GBM-FS. The architecture of the feature selection techniques and the architecture of the proposed stacked ensemble model are shown in Figure 4.1 and Figure 4.2 respectively.

4.2.1 Proposed Ensemble Feature Selection Technique

Similar to feature selection, the Ensemble Feature Selection (EFS) by the HESCA model combines individual base feature selectors to select the most significant input features in the

training phase that produce improved robust outcomes for the HESCA model. The idea behind the EFS is to solve the instability and unreliability problem that can be accounted for using a single feature selector (Xu *et al.*, 2019; Wang *et al.*, 2019). Feature selection aims at overcoming the problem of dimensionality in the training data that improves the model's accuracy compared to that consisting of full-input features. Undoubtedly, a cost-effective model can be produced. This is considered crucial in clinical study in which fewer feature input implies lower diagnostic or prognostic cost as well as test.

Meanwhile, the main purpose of GBM-FS (Xu *et al.*, 2019) in the present study is to identify optimum feature subset (number of features) given the number of training instances under the study that the proposed HESCA model can learn to produce a robust prognostic model for recurrent HNSCC prognosis. The purpose for employing ensemble technique in selecting features is to enhance the overall selection of the optimal features to reduce instability that can be caused by a single feature selector. Five feature selection techniques are experimented: two of which are ensemble ML techniques (GBM and DRF) for homogeneous ensemble FS; and three of which are standalone ML techniques (DNN, NB, and GLM) for heterogeneous single FS, have been selected and implemented in this study. The weighted voting ensemble technique for ensemble feature selection assigns various weights to the features based on specific criteria and selects the features based on the weight. The information on HNSCC patients under study considered as training features range from gender denoted as Gen (x_1) to treatment type denoted as treat (x_{18}) as shown in Figure 4.1.

Each technique; GBM, DRF, DNN, GLM, and NB is used to select features where each one provides a feature subset by sorting these features in order of their importance. Consider a labeled dataset $D = \{\mathbf{x}_i, y_i\}_{i=1}^n$ with n instances and feature space $\mathbf{x} = (x_1, x_2, \dots, x_n)$. Consider also $FS = \{FS_1, FS_2, \dots, FS_t\}$ consisting of t number feature selectors (FS). Each FS_i offers a feature subset $FS_i = \{x_1^i, x_2^i, \dots, x_{n-1}^i\}$, and $n-1$ means the maximum features that the FS_i selects. For each feature in the i^{th} subset, a weight of importance is computed. Based on a given threshold, the $\alpha\%$ features exceeding a threshold are selected from each ranked list of feature importance to form feature subsets. The optimal feature subset of the dataset as new dataset $D_{new} = \{\mathbf{x}'_i, y'_i\}_{i=1}^n$ based on various feature selection techniques is obtained using HESCA model to learn and evaluate on FS_i .

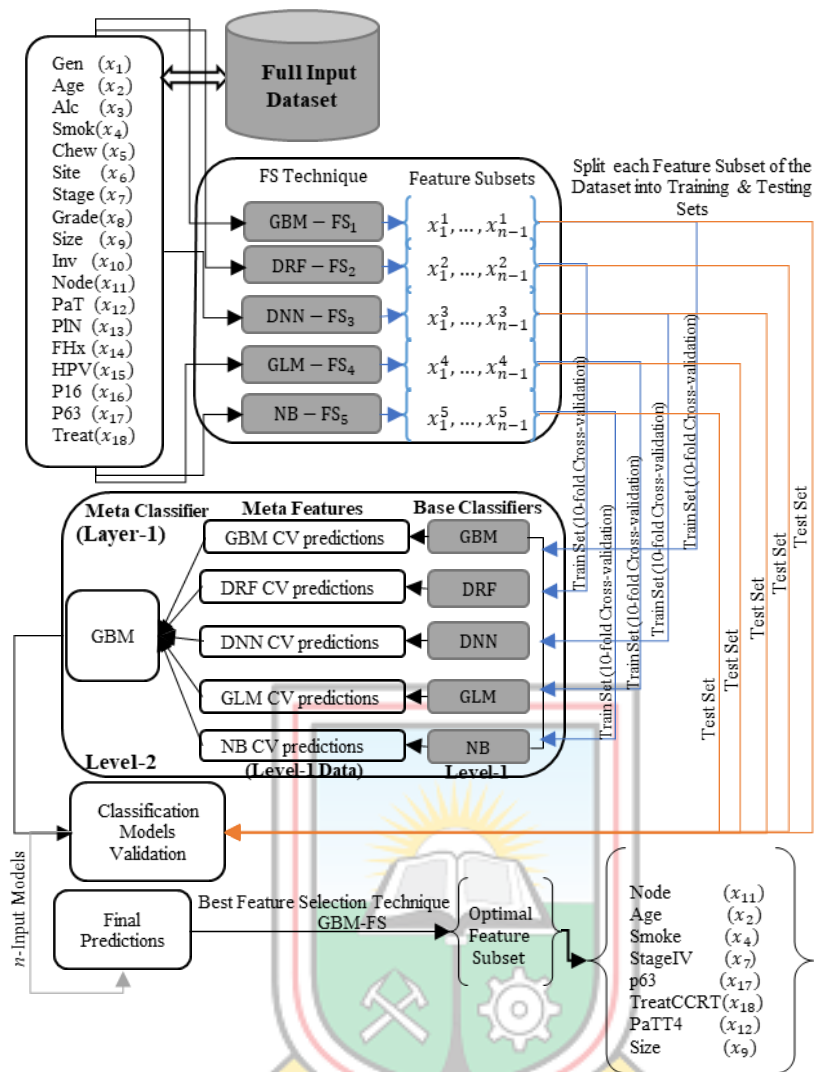


Figure 4.1 Architecture of Feature Selection Techniques

4.2.2 The Proposed Stacked Ensemble Classification Model for HESCA

This section presents a detailed description of the proposed stacked ensemble classification (HESCA) model for recurrent HNSCC. This classification model is based on stacking ensemble technique, and is called the HESCA model. By stacked ensemble technique, the HESCA model is developed on the optimal feature subset (gradient boosted features) using the model hyperparameter values.

In general, two steps—*Selection* and *Combination*—are taken into consideration when creating a HESCA model. The choice of the component classifiers is thought to be crucial for the effectiveness of the HESCA model, and the diversity and accuracy of these classifiers are the key factors in this regard (Rokach, 2010; Dietterich, 2001). The predictions of the individual classifiers are combined using a variety of techniques with various philosophies.

This model is the adaptation of the two already existing models: Kabir and Ludwig (2019); Kwon *et al.* (2019).

The stacked ensemble classification model; specifically stacked ensemble learning proposed by Kabir and Ludwig (2019) is a super learning ensemble model that found the optimal weighted average of diverse base learners for classification of various healthcare datasets. Kwon *et al.* (2019) likewise proposed a stacking ensemble model that found the best meta-learner for classifying breast cancer. Note that both super learning ensemble and stacking ensemble mean the same learning process. Kabir and Ludwig (2019) employed three (GBM, DRF, and DNN) and two (GBM and DRF) different machine learning algorithms as base classifiers, and used GLM as the meta-classifier in each case to determine the final stacked ensemble result. Their best stacked ensemble model consisted of three base models and a GLM meta-model. The algorithm of this technique is shown in Table 4.1. In contrast, Kwon *et al.* (2019) used the machine learning (ML) algorithms GBM, DRF, GLM, and DNN as base classifiers. Each of these methods was then used as a meta-classifier to stack base classifiers in order to create a robust meta-classifier model in a stacked ensemble learning with four basis classifiers. Their best stacked ensemble model consisted of four base models and a GBM meta-model. The algorithm of this technique is shown in Table 4.2. In stacked ensemble learning, the meta-learning algorithm is specified in building a classifier with the purpose to enhance the overall performance (generalisation ability) of a classification model, to regularise the linear model, to minimise the cross-validated risk of a loss function of interest such as squared error loss or rank loss that a base learner may cause (LeDell, 2016) so as to maximise the Area Under ROC Curve. The meta-learning algorithm takes predicted labels (predictions) made by the first-level classifiers as input feature space to learn meta-model. This study presents a stacked ensemble model having five base models (with 10-fold cross-validation) and GBM meta-model, and GBM ensemble feature selection (GBM-FS) model. This is where 10-fold cross-validation is performed on each base classifier using GBM-FS optimal feature subset, and the cross-validated predicted labels provided by the five base models along the original class labels serving the level-1 dataset, is used to learn meta-learning algorithm.

Table 4.1 Baseline Stacked Ensemble Algorithm with V-fold Cross Validation

Input:	Dataset $D = \{\mathbf{x}_i, y_i\}_{i=1}^n$; learning rate $\alpha > 0$
	$C = \{h_1, h_2, \dots, h_L\}$ – classifiers set which constitute the ensemble.
	h' = meta-learner
Output:	An ensemble classifier H
Step 1: Adopt CV approach in preparing a training set for meta-classifier	
Randomly split D_s into V equal-size subsets: $D = \{D_1, D_2, \dots, D_V\}$	
for $v \leftarrow 1$ to V do	
Step 1.1: Learn first-level classifiers $\{h_1, h_2, \dots, h_L\}$	
for $l \leftarrow 1$ to L do	
Learn a classifier h_{vl} from D/D_v	
end for	
Step 1.2: Construct a training set for second-level classifiers	
for $\mathbf{x}_i \in D_v$ do	
Get a record $\{\mathbf{x}'_i, y_i\}$, where $\mathbf{x}'_i = \{h_{v1}(\mathbf{x}_i), h_{v2}(\mathbf{x}_i), \dots, h_{vL}(\mathbf{x}_i)\}$	
end for	
end for	
Step 2: Learn second-level classifier	
Learn a <i>new classifier</i> h' from the collection of $Z = \{\mathbf{x}'_i, y_i\}_{i=1}^n$	
end for	
Return $H(\mathbf{x}) = h' (h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_L(\mathbf{x}))$	
Step 3: Predict unseen example (testing set)	
for each $\mathbf{x} \in D_t$ do	
Apply an ensemble classifier $H(\mathbf{x})$ on \mathbf{x} .	
end for	

Table 4.2 State-of-the-Art Stacked Ensemble Algorithm with V-fold Cross Validation

Input:	Dataset $D = \{\mathbf{x}_i, y_i\}_{i=1}^n$; learning rate $\alpha > 0$
	$C = \{h_1, h_2, \dots, h_L\}$ – classifiers set which constitute the ensemble.
Output:	An ensemble classifier H
Step 1: Adopt CV approach in preparing a training set for meta-classifier	
Randomly split D_s into V equal-size subsets: $D = \{D_1, D_2, \dots, D_V\}$	
for $v \leftarrow 1$ to V do	
Step 1.1: Learn first-level classifiers $\{h_1, h_2, \dots, h_L\}$	
for $l \leftarrow 1$ to L do	
Learn a classifier h_{vl} from D/D_v	
end for	
Step 1.2: Construct a training set for second-level classifiers	
for $\mathbf{x}_i \in D_v$ do	
Get a record $\{\mathbf{x}'_i, y_i\}$, where $\mathbf{x}'_i = \{h_{v1}(\mathbf{x}_i), h_{v2}(\mathbf{x}_i), \dots, h_{vL}(\mathbf{x}_i)\}$	
end for	
end for	
Step 2: Learn second-level classifier	
Re-learn first-level classifiers h'_l from the collection of $Z = \{\mathbf{x}'_i, y_i\}_{i=1}^n$	
end for	
Return $H(\mathbf{x}) = h' (h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_L(\mathbf{x}))$	
Step 3: Predict unseen example (testing set)	
for each $\mathbf{x} \in D_t$ do	
Apply an ensemble classifier $H(\mathbf{x})$ on \mathbf{x} .	
end for	

To generate the HESCA model for better performance, the novel approach for recurrent HNSCC prognosis in cancer medical settings is proposed. For the purpose of the study, the system adapts to enhance the existing stacked ensemble models of Kabir and Ludwig (2019), and Kwon *et al.* (2019) as shown in Table 4.1 and Table 4.2 respectively above, by improving in the areas of:

- (i) More base classifiers against few as used by both studies.
- (ii) More diverse meta-classifiers against few as used by both studies.

- (iii) More base classifiers and each used as a meta-classifier against few used by Kwon *et al.* (2019) and where no such was used by Kabir and Ludwig (2019).
- (iv) Incorporation of the regularisation technique as not done by both studies.

The philosophy that inspires the study is that, “*the learner is improved as it learns from more diverse of its co-learners when regularised.*”

Therefore, five selected ML classifiers: GBM, DRF, DNN, GLM, and NB are employed to learn and evaluate the performance of the proposed HESCA model based on the optimal feature subset generated by the GBM-FS technique. The steps to the proposed adapted learning technique are presented as follows:

i. Classification model data

Consider a labeled optimal feature subset data $D_{new} = \{\mathbf{x}'_i, y'_i\}_{i=1}^n$ with n instances and feature vectors $\mathbf{x}' = (x'_1, x'_2, \dots, x'_{n-1}, y'_n)$. Assume that the training set is comprised of n independent and identically examples, $\{O_1, O_2, \dots, O_n\}$, where $O_i = (x'_i, y'_i)$; here, x'_i is vector of covariate or feature value and y'_i is the outcome.

ii. Data sample for classification model

Partition the new dataset D_{new} into two subsets; D_s as training set and D_t as testing set, whereby the class information is known for the training set, and it is unknown for the testing set (unseen data). These data sets are referred to as level-0 data.

iii. Set up the ensemble

- Specify the library of L base learning algorithms, $\{\psi_1, \psi_2, \dots, \psi_L\}$, each of which is indexed by an algorithm class as well as a specific set of model hyperparameters.
- Specify meta-learning algorithms (classifiers).

iv. Learn base classifiers

Learn each of the L base classifiers on the training set D_s . That is, learn classifier ψ_k on the training set. These classifiers learned on the entire training set are referred to as “full-fit” classifiers.

v. Perform stacking with V -fold cross-validation: Create training set and validation set based on D_s .

- Randomly split the training set \mathbf{x}' into V mutually exclusive and exhaustive equal-size folds $D_s = \{D_{s1}, D_{s2}, \dots, D_{sV}\}$, where $v = \mathbf{x}'^{(1)}, \mathbf{x}'^{(2)}, \dots, \mathbf{x}'^{(V)}$.
 - Within each fold v , where $v = 1, 2, \dots, V$, $(V-1)/V$ or $(V-1)$ and $1/V$ of the dataset are used for training set (T_r) and validation set (V_d) respectively.
- (a) Learn first-level classifiers based on the training dataset.
- Perform V -fold cross validation on each of L classifiers and collect the cross validated predicted labels from of these L classifiers. That is, for each base classifier in the ensemble ψ_L , V -fold cross-validation is used to generate V cross-validated predictions associated with I^{th} classifier. These V -dimensional vectors of cross-validated predictions become the L columns of Z .
 - within each fold v and at each round, learn ψ_k on T_r and validate it on V_d to obtain meta-feature space or prediction functions $\{\hat{\psi}_1, \hat{\psi}_2, \dots, \hat{\psi}_L\}$, where $\hat{\psi}_k = \{z_{v^1k}, z_{v^2k}, \dots, z_{v^nk}\}$, which constitute the input feature space for the meta classifiers. It is worth noticing that, each classifier is learned (fitted) V times. These classifiers learned across the V cross-validation folds are referred to as “cross-validated fit” classifiers.
 - combine the cross-validated predictions (predicted labels or values) from all folds of all learners to generate the so-called L column ($n \times L$) matrix of cross-validated predictions. This matrix is also commonly referred to as Z of k -fold cross-validated predictions.
 - combine ($n \times L$) matrix with the original label (Y) of the training data to obtain **Level-one data**.

As shown in Fig. 4.2, here, level-one dataset is the combination of the cross-validated predictions provided by first-level classifiers termed as meta-features and the original class labels $(y'_1, y'_2, \dots, y'_n)$. It is assumed that for each example in D_s as $\{\mathbf{x}'_i, y'_i\}$, the corresponding example $\{\mathbf{x}''_i, y_i\}$ in the new dataset is constructed, where $\mathbf{x}''_i = \{\psi_{v1}(\mathbf{x}'_i), \psi_{v2}(\mathbf{x}'_i), \dots, \psi_{vL}(\mathbf{x}'_i)\}$.

- (b) Learn meta classifiers based on the newly constructed dataset Z .
- Stack first-level classifiers using each base classifier (re-learn k^{th} base-classifier on the level-one data).

- Each provides a meta-feature space or prediction function $\{\hat{\psi}'_1, \hat{\psi}'_2, \dots, \hat{\psi}'_L\}$.

The cross-validated predictions made by the L base classifiers become the inputs for the meta classifiers. Therefore, once the meta classifiers are generated, combine the first-level classifiers along with the original class labels. Learn the meta-classifiers based on the newly constructed dataset. The best meta-classifier ψ_{best} , provides the output space $\hat{\psi}_{\text{best}}$. Each meta classifier provides a (meta-feature space) prediction function $\{\hat{\psi}'_1, \hat{\psi}'_2, \dots, \hat{\psi}'_L\}$, which then become the output. It is worth mentioning for stacked ensemble that, all base classifiers must have been learned with the same number of cross-validation folds, and they must all use the same fold assignment to ensure that the same observations are used.

vi. Generate output/results

Finally, use the hybrid ensemble super classification algorithm model to generate predictions on the test set D_t (unseen example) to predict class label. Then, the HESCA model predicts the class label of an unseen example \mathbf{x}' as;

$$H(\mathbf{x}') = \psi_{\text{best}}\{\psi_1(\mathbf{x}'), \psi_2(\mathbf{x}'), \dots, \psi_L(\mathbf{x}')\} \quad (4.5)$$

where, $\{\psi_1, \psi_2, \dots, \psi_L\}$ are the first-level classifiers and ψ_{best} is the meta classifier.

Assumptions of the HESCA model

In the formulation of the HESCA model, the following assumptions are made. It is assumed that;

- all patients of at least 15 years were initially diagnosed with any of the HNSCC subtypes only under study between the 2016 and 2020 calendar period.
- all patients were treated with curative intent only and were followed up until the end of this calendar period.
- treatment is proportional to the stage of the tumor at diagnosis for all patients.
- all patients treated with curative intent can only experience either recurrence or non-recurrence.
- sample items are all independent and identically distributed.
- the model does not allow patients treated with palliative intent.

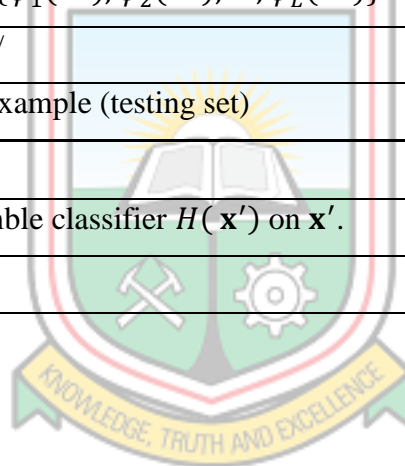
- g. the model does not allow patients treated with curative intent and later developed a second primary tumor.

HESCA model learns through the algorithm presented in Table 4.3 and the architecture of the HESCA model is shown in Figure 4.2.

Table 4.3 Proposed Hybrid Ensemble Super Classification Algorithm for HESCA

Input:	Dataset $D = \{\mathbf{x}_i, y_i\}_{i=1}^n; \alpha > 0$
	GBM boosted feature subset $D_{new} = \{\mathbf{x}'_i, y'_i\}_{i=1}^n$
	D_s - Training set
	D_t - Testing set
	V_d - Validation set from D_s
	T_r – training set from D_s
	EFS = $\{FS_1, FS_2, \dots, FS_t\}$ —feature selection algorithms which constitute EFS.
	<i>Early stopping</i> – which constitutes the regularisation hyperparameter
	$C = \{\psi_1, \psi_2, \dots, \psi_L\}$ —classifiers set which constitute the ensemble.
Output:	A HESCA classifier model H
/*Phase I: Feature Selection*/	
Step 1: Obtain feature subsets from various feature selectors	
for algorithm FS_i in $\{FS_1, FS_2, \dots, FS_t\}$	
Use dataset D to do feature selection by feature selector FS_i	
for $i \leftarrow 1$ to t do	
Get weight sequence of t feature selectors	
end for	
Step 2: Get best feature sequence according to α	
Sort FS base on W	
$\alpha\%$ features in subset are put first	
Return SUBSET _{best}	
/*Phase II: Training*/	
Step 3: Adopt CV approach in preparing a training set for meta-classifier	
Randomly split D_s into V equal-size subsets: $D_s = \{D_{s1}, D_{s2}, \dots, D_{sV}\}$	
for $v \leftarrow 1$ to V do	
Step 4.1: Learn first-level classifiers $\{\psi_1, \psi_2, \dots, \psi_L\}$	

for $l \leftarrow 1$ to L do
Learn a classifier ψ_{vl} from D_s/D_{sv}
end for
Step 4.2: Construct a training set for second-level classifier
for $\mathbf{x}'_i \in D_{sv}$ do
Get a record $\{\mathbf{x}''_i, y'_i\}$, where $\mathbf{x}''_i = \{\psi_{v1}(\mathbf{x}'_i), \psi_{v2}(\mathbf{x}'_i), \dots, \psi_{vL}(\mathbf{x}'_i)\}$
end for
end for
Step 4: Learn meta classifier
for $l \leftarrow 1$ to L do
Re-learn each first-level classifier ψ'_l from the collection of $Z = \{\mathbf{x}''_i, y'_i\}_{i=1}^n$
end for
Return $H(\mathbf{x}') = \psi_{\text{best}}\{\psi_1(\mathbf{x}'), \psi_2(\mathbf{x}'), \dots, \psi_L(\mathbf{x}')\}$
/*Phase III: Evaluation*/
Step 5: Predict unseen example (testing set)
for each $\mathbf{x}' \in D_t$ do
Apply an ensemble classifier $H(\mathbf{x}')$ on \mathbf{x}' .
end for



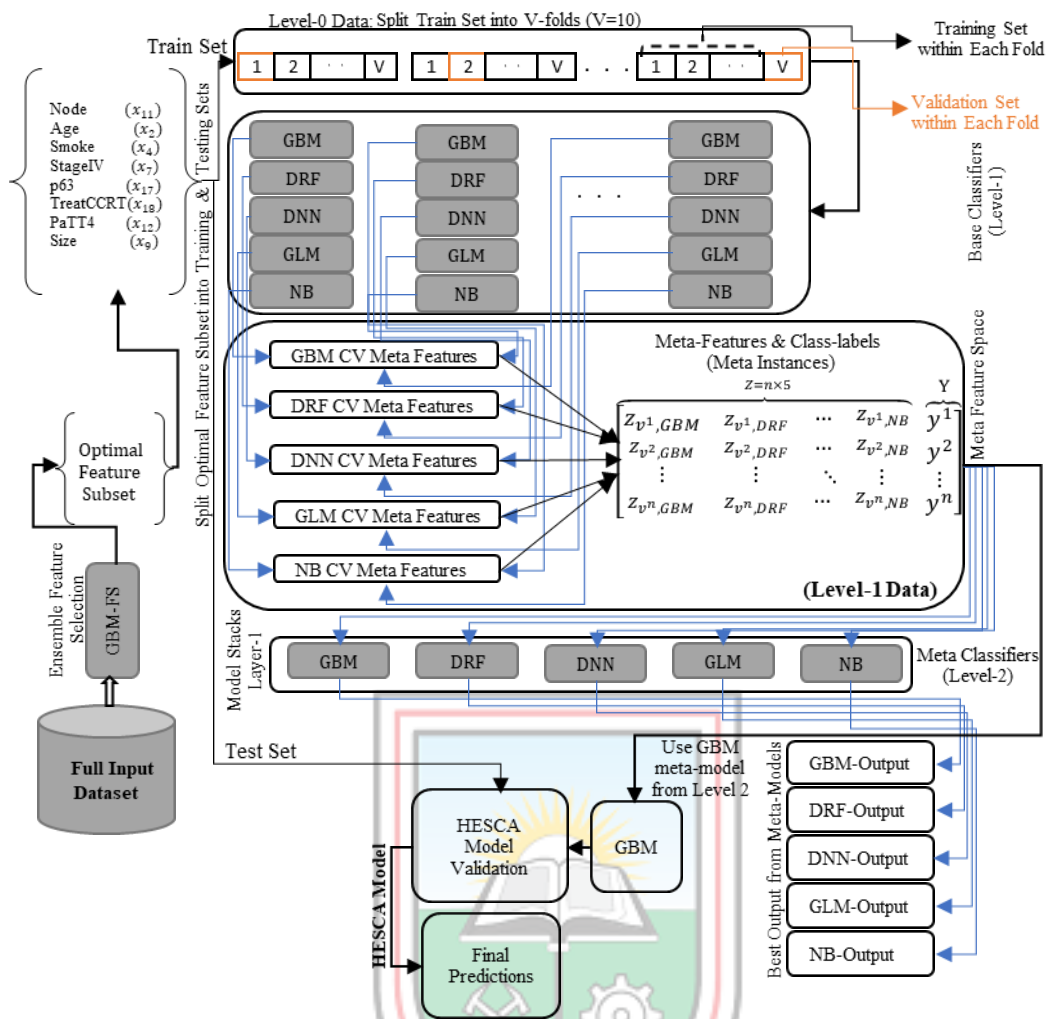


Figure 4.2 Architecture of HESCA Model for Recurrent HNSCC Prognosis

The architecture of the HESCA model for recurrent HNSCC prognosis in Figure 4.2 explains from data pre-processing to model evaluation. The original data is subjected ensemble feature selection using GBM, which provides the gradient boosted features termed as the optimal feature subset of the original data. This feature subset becomes the new dataset which is split into training and testing sets on which the proposed hybrid model is learned and evaluated respectively. The training set is further partitioned into V-fold. At level-1, the V-fold cross-validation is performed on each base classifier model based on the training set, each of which provides the cross-validated predictions termed as the meta-features. These meta-features are combined with the original class labels to form a level-1 dataset for meta-classifiers. Here, each base classifier serves as a meta-learning classifier in order to determine the best meta-classifier model for recurrent HNSCC. At level-2 or layer-1, each base classifier model is learned on the level-1 dataset. The outputs of each classifier model at the base-learning (level-1) and at the meta-learning (level-2) show that the GBM

is the best meta-classifier model. This indicates that the GBM can be used as a meta-classifier model straightforward in a stacking ensemble consisting of five base classifier models in the learning of recurrent HNSCC data without verifying other base classifier models at the meta-learning or stacking stage. Now, based on the output of the GBM meta-classifier model, the evaluation is performed to determine the final predictions of the HESCA model.

HESCA Model Hyper-parameters Identification

The HESCA model is developed based on the hyperparameters obtained by random grid search as shown in Table 4.4 below.



Table 4.4 The HESCA Model Hyperparameters for Recurrent HNSCC Prognosis

Classifiers	Hyperparameters in grid search	Hyperparameters fixed values
GBM	max_depth = c(7, 9), min_rows = c(1, 3, 5), learn_rate = c(0.01, 0.1), sample_rate=c(0.5, 0.75, 1), col_sample_rate=c(0.8, 0.9, 1)	ntrees = 5000 nfolds = 10 fold_assignment = "Modulo" keep_cross_validation_predictions = TRUE stopping_metric = "logloss" stopping_tolerance = 0.0001 stopping_round = 50
DRF	max_depth = c(9, 30), mtries = 3, min_rows = c(1, 5, 10), sample_rate = c(0.5, 0.75, 1), col_sample_rate = (0.8, 0.9, 1)	ntrees = 5000 nfolds = 10 fold_assignment = "Modulo" keep_cross_validation_predictions = TRUE stopping_metric = "logloss" stopping_tolerance = 0.00001 stopping_round = 50
DNN	activation=c("Rectifier", "Maxout", "Tanh"), hidden = list c(5, 5, 5, 5, by 10), epochs = c(50, 100, 200), l1 = c(0, 1e-3, 1e-5), l2 = c(0, 1e-3, 1e-5), rate =c(0, 0.1, 0.005, 0.001)	epochs = 20 nfolds = 10 fold_assignment = "Modulo" keep_cross_validation_predictions = TRUE stopping_metric = "logloss" stopping_tolerance = 0.0001 stopping_round = 10
NB	laplace=c(0, 5, by 0.5)	nfolds = 10 fold_assignment = "Modulo" keep_cross_validation_predictions = TRUE stopping_metric = "logloss" stopping_tolerance = 0.0001 stopping_round = 10
GLM	alpha=c(0.1)	nfolds = 10 remove_collinear_columns = TRUE fold_assignment = "Modulo" keep_cross_validation_predictions = TRUE stopping_metric = "logloss" stopping_tolerance = 0.0001 stopping_round = 10

Implementation of the HESCA Model

First, using the training and testing datasets from the original dataset, the HESCA model is appropriately learned and tested. Figure 4.3 illustrates the poor performance of this model on training and testing sets, which is a full-input HESCA model made out of full-input features (redundant and irrelevant features). To improve the generalisation ability of the HESCA model on both training and testing sets, feature selection is performed using five ML classifiers under study, where each provides a feature subset and where each ranks its features based on their importance. Once the promising feature subsets are obtained through individual feature selection techniques, they are effectively used for training and testing using the HESCA model to identify the optimal feature subset for that feature selection technique (model).

Meanwhile, based on this optimal feature subset provided by the best feature selection model, each base classifier is effectively trained and tested. To improve the generalisation ability of these base classification models, the stacked generalisation technique with 10-fold cross-validation is implemented; where stacked ensemble models are developed by using each base classifier as a meta classifier to stack base classifiers in a stacking ensemble consisting of five base classifiers. The best-stacked ensemble model is identified based on the outputs provided by the components of stacked ensemble models. Thus, the best-stacked ensemble model is the model with the best performance both at the base learning level and the meta-learning level with outperforming performance. Generally, the proposed Hybrid Ensemble Super Classification model for recurrent HNSCC is by far a conjunction of the Gradient Boosted Features (GBF)-GBM ensemble feature selection (GBM-FS) model, five base classification models (GBM, DRF, DNN, GLM, and NB), and a GBM meta-model with 10-fold cross-validation.

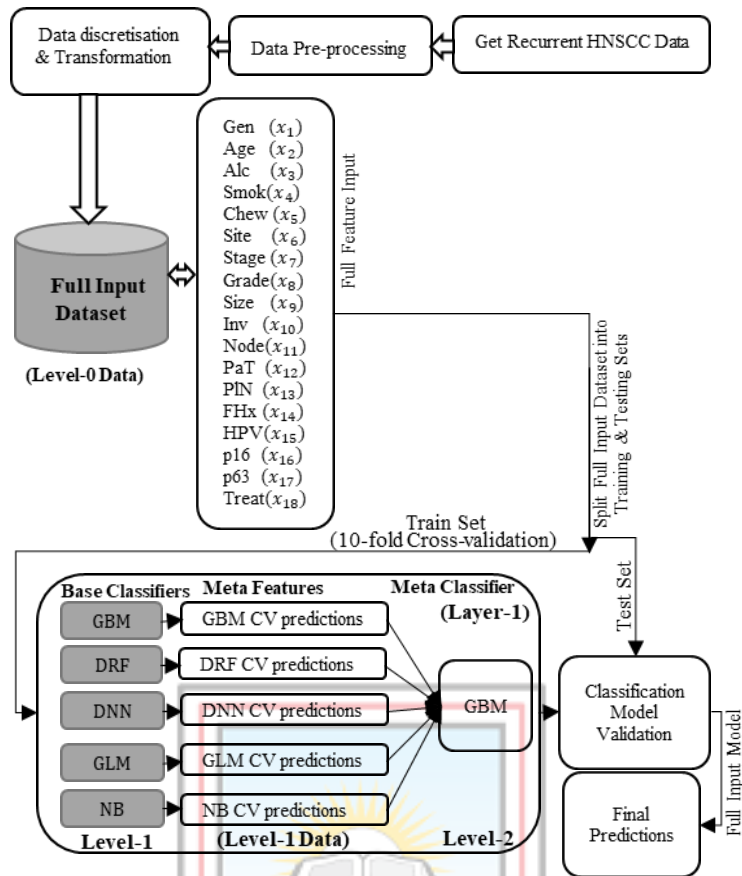


Figure 4.3 Architecture of HESCA Model with Full-Input Features

The Figure 4.3 explains the architecture of the HESCA model with full-input features. Having developed the HESCA model parameters as shown in Table 4.4, it can be applied to the original dataset that has no feature selection. Here, the original dataset is split into training and testing sets. The HESCA model is learned on the training set and evaluated on the testing set.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Overview

This chapter presents the analyses and discussions of the proposed HESCA prognostic model for recurrent HNSCC prognosis that identifies, classifies, and predicts the recurrence patterns as *recurrence* or *nonrecurrence* of HNSCC patients after 1 to 5 years. Within this period, the patients had the diagnosis which was followed by treatment with curative intent.

The dataset consisting of clinical, pathological, and genomic information under study are available from NCRNM at the Radiotherapy and Oncology Department (ROD). 15 clinicopathologic features were identified with the aid of the HNC experts at ROD. Regarding the features of genomic data, only three features were identified and considered in the study, which are; *p16*, *p63*, and *HPV* as a result of time, cost, and the limitation to medical tissues. A total of 185 HNSCC instances were provided with the help from the staff of ROD, and based on the sample size of the study, 125 instances were selected and considered for the study. The mode imputation technique discussed in Chapter 3 is implemented in the original dataset as a way to cleanse the data. To normalise the features into the binary range [0, 1], one-hot encoding is implemented on features with multi-levels. The HESCA model is implemented on the normalised dataset. Next, the feature selection techniques are implemented on the normalised dataset for optimal feature subsets. This optimal feature subset of the HNSCC dataset is partitioned into two sets: training set (75%) and testing set (25%) based on the experimental model performance on the these sets if compared to its performance on the 70% (or 80%) training set and 30% (or 20%) test set.

The proposed HESCA model is a classifier model with a 10-fold CV implemented on the gradient boosted feature training set generated from the GBM-FS ensemble feature selection technique to develop HESCA prognostic model and its prediction performance is evaluated on the test set. For the evaluation, the results of the HESCA model with full-input features are compared with that of the HESCA model with GBFs (8-input features) provided by the ensemble feature selection technique of GBM-FS. Also, the results of the HESCA model are compared with the results of five base models (GBM, DRF, DNN, GLM, and NB) as well as two baseline stacked ensemble models and one state-of-the-art stacked ensemble model. Finally, the HESCA model is used to classify and predict the 5-year recurrent

HNSCC prognosis pattern using a Partial Dependence Plot (PDP) and Individual Conditional Expectations (ICE). Thus, this chapter answers the four core objectives of the study.

5.2 Multicollinearity check using Variance Inflation Factor Technique

To check for multicollinearity of features in the recurrent HNSCC dataset, Variance Inflation Factor (VIF) is used and discussed as shown the Table 5.1.

Table 5.1 Variance Inflation Factor (VIF) of Features

SN	Feature Name	VIF Value
1	Gender	3.998071
2	Age	1.950284
3	Alcohol	3.373524
4	Smoke	3.823554
5	Chew	2.017331
6	Site	2.606477
7	Stage	2.010550
8	Grade	1.933437
9	Size	1.973634
10	Invasion	2.582278
11	Nodes	1.283542
12	PaT	4.122057
13	PIN	3.984016
14	FHx	2.410912
15	HPV	3.010458
16	<i>p16</i>	3.021032
17	<i>p63</i>	3.295038
18	Treat	1.758475

Table 5.1 shows the Variance Inflation Factor (VIF) for each of the 18 independent features. It can be observed that the VIF of each feature is less than 5. For severe multicollinearity problems, the VIF should be greater than 10. Since the value for each of the 18 features is smaller; much less than 10, it can be concluded that there is no multicollinearity among the

features as far as this dataset is a concern. This is shown in Figure 5.1 and can be observed that VIF of every feature lies below the value of 5 along the vertical axis. Thus, it can further be concluded that these features in the dataset under consideration are independent and identically distributed; hence, one can proceed to build the classification model for recurrent HNSCC prognosis.

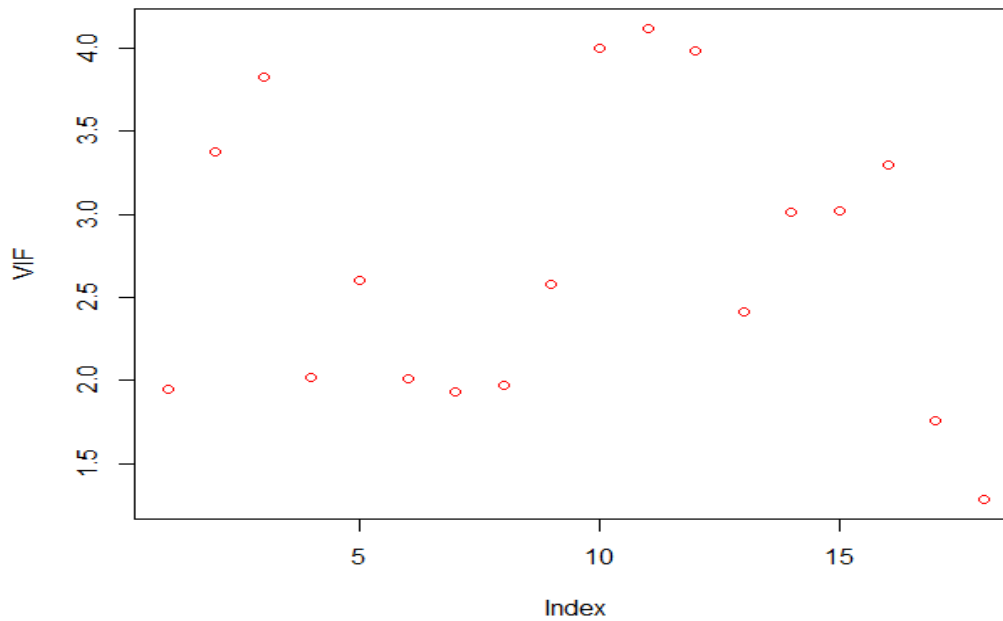


Figure 5.1 Variance Inflation Factor Plot for Multicollinearity

Table 5.2 Performance Metrics of HESCA Model with Full-Input Features

Metrics	HESCA Model on Original Training Set	HESCA Model on Original Test Set
Accuracy	0.3441	0.3438
Log-loss	0.8025	1.0435
Recall	0.3023	0.3846
Specificity	0.8571	0.1667
Precision	0.9630	0.6667
F1-Score	0.4602	0.4878
AUC	0.4879	0.4364

Table 5.2 shows the performance of the HESCA model with full-input features on both the training set and test set. It can be observed that the accuracies for training and test sets of

the HESCA model with full-input features are 34.41% (with a high log-loss value of 0.8025) and 34.38% (with a high log-loss value of 1.0435) respectively. It should be noted that these log-loss values are much higher than the accuracy of the model with full-input features, indicating that the model is not learning poorly. This calls for feature selection to reduce the number of features (remove irrelevant and/or redundant features) in the dataset given a fixed number of training instances. The log-loss value should be between 0 and 1 inclusive. The log-loss value (1.0435) for the HESCA model on test set exceeds 1, indicating that the predicted probability for a given class is less than $\exp(-1)$ or around 0.368. Therefore, looking at this log-loss value, it can be expected in the case that the model only give less than a 36% probability estimate for the actual class.

5.3 Feature Selection Techniques

To implement the feature selection technique, the HESCA model is implemented to the overall dataset and the performance metrics are recorded. This informs the choice of feature selection. Five classifiers under study GBM, DRF, DNN, NB, and GLM are commonly employed in this research and are used for feature selection. Each one provides a feature subset in which features are ranked according to their importance. A threshold of 60% is used to obtain feature subsets as presented in Table 5.3 and Table 5.4. To ascertain the optimal feature subset, each feature subset of the dataset is trained and validated using the HESCA model. This is achieved using the R programming language of the statistical software based on the H2O package to implement the proposed HESCA model.

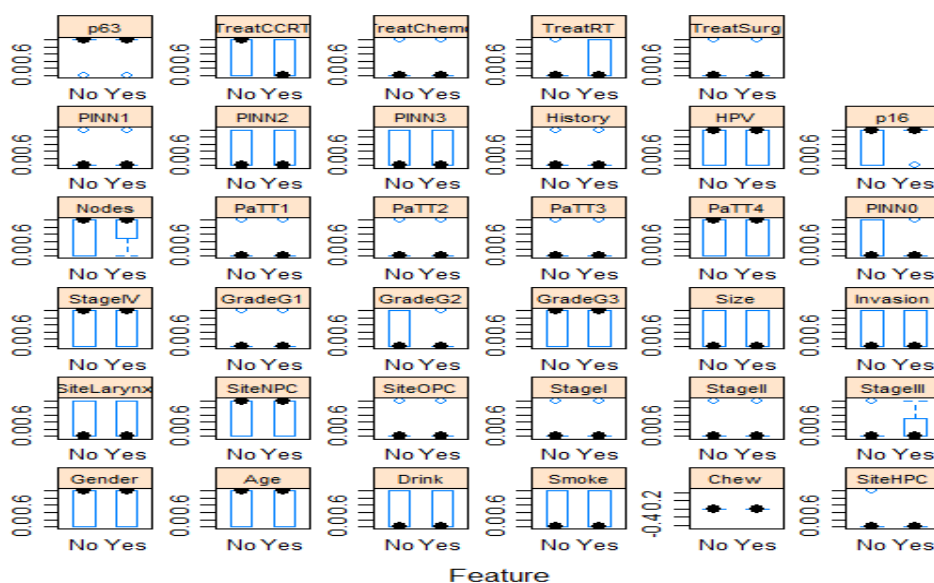


Figure 5.2 Boxplots for Features

Figure 5.2 shows a boxplot of training features, which has subplots and each of which has two blue plots representing the Yes and No categories of the label, indicating the region for which the training examples lie. The top and bottom of the box represent the 25th and 75th percentiles respectively with a black dot representing the mean. Considering the training feature TreatCCRT (concurrent chemotherapy treatment). The placement positions of the means in the two plots are different, indicating that the said feature TreatCCRT could be a significant prognosis that predicts the label as represented in Figure 5.2.

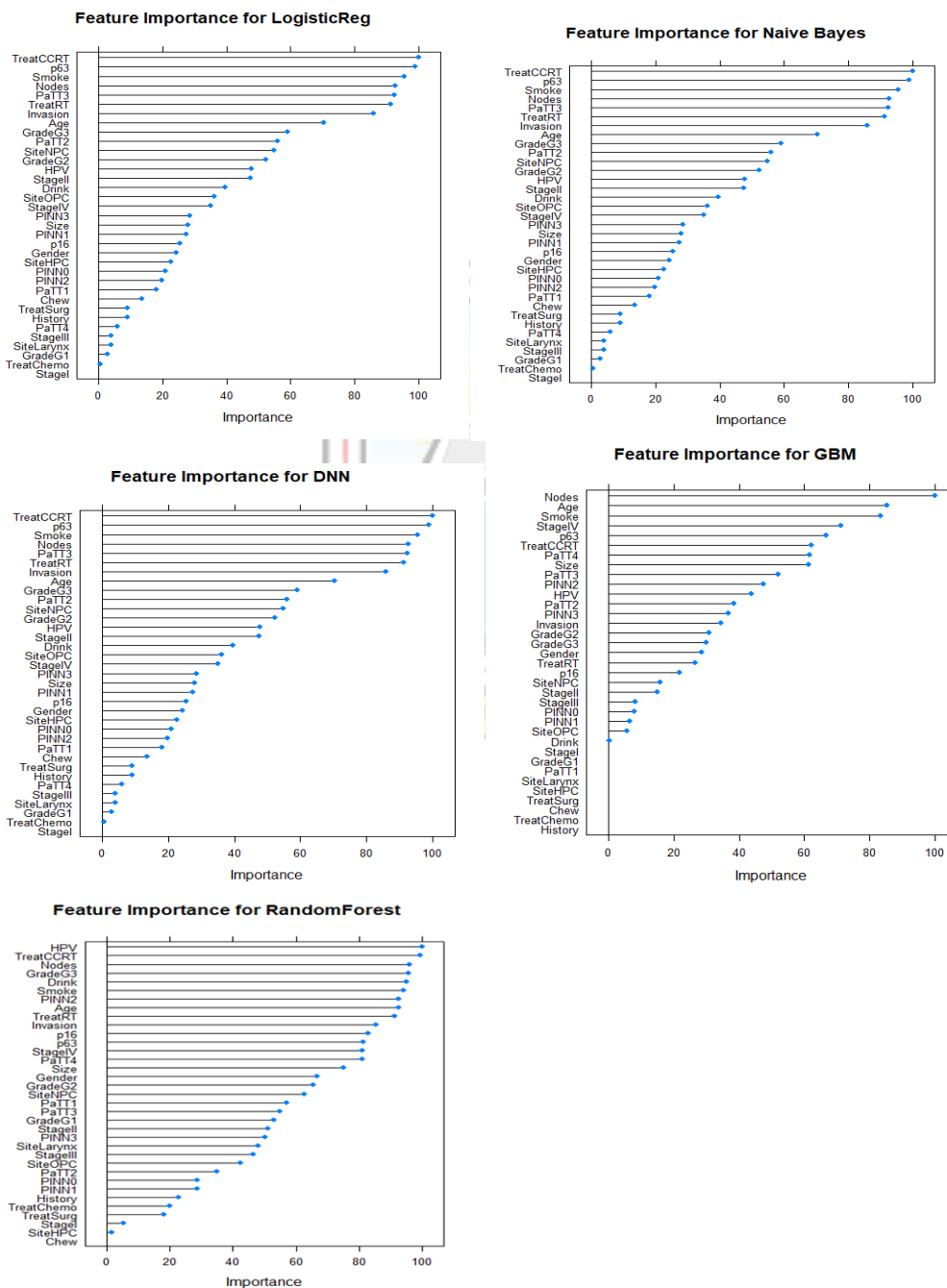


Figure 5.3 Ranks of Features

Figure 5.3 shows the rank of top most 20 features provided by each FS technique. Each FS technique ranks the features according to their importance to the class label. Table 5.3 shows the selected features ranked in Figure 5.3.

Table 5.3 Top 20 Most Important Features Selected

Feature Selection Techniques					
DNN-FS, GLM-FS, and NB-FS		GBM-FS		DRF-FS	
Features	Rank	Features	Rank	Features	Rank
TreatCCRT	1.0000	Nodes	1.0000	HPV	1.0000
p63	0.9900	Age	0.8548	TreatCCRT	0.9944
Smoke	0.9568	Smoke	0.8350	Nodes	0.9586
Nodes	0.9269	StageIV	0.7126	GradeG3	0.9565
paTT3	0.9236	p63	0.6666	Drink	0.9510
TreatRT	0.9136	TreatCCRT	0.6222	Smoke	0.9425
Invasion	0.8605	PaTT4	0.6167	PINN2	0.9267
Age	0.7043	Size	0.6133	Age	0.9263
GradeG3	0.5914	PaTT3	0.5212	TreatRT	0.9142
PaTT2	0.5581	PINN2	0.4753	Invasion	0.8545
SiteNPC	0.5482	HPV	0.4382	p16	0.8276
GradeG2	0.5216	PaTT2	0.3844	p63	0.8136
HPV	0.4784	PINN3	0.3662	StageIV	0.8105
StageII	0.4751	Invasion	0.3440	PaTT4	0.8091
Drink	0.3953	GradeG2	0.3079	Size	0.7503
SiteOPC	0.3621	GradeG3	0.2994	Gender	0.6681
StageIV	0.3488	Gender	0.2851	GradeG2	0.6534
PINN3	0.2857	TreatRT	0.2644	SiteNPC	0.6277
Size	0.2791	p16	0.2170	PaTT1	0.5710
PINN1	0.2724	SiteNPC	0.1583	PaTT3	0.5473

Table 5.3 shows the top 20 significant features using feature selection techniques. Based on default, the top 20 features are ranked based on their importance whilst ignoring the rest 15 under this study which are far below the significance. Feature subsets provisionally considered optimal for each feature selector are obtained using the threshold that ranges

between 60% and 100% so that features lying within the threshold are potentially considered important. Thus, the feature is considered important in the feature subset if it is assigned 60% to 100% weight of importance.

Table 5.4 Optimal Feature Subsets by various Feature Selection Techniques

FS Technique	Feature Subsets
GBM-FS	<i>Nodes, Age, Smoke, StageIV, p63, TreatCCRT, PaTT4, Size</i>
DRF-FS	<i>HPV, TreatCCRT, Nodes, GradeG3, Drink, Smoke, PINN2, Age, TreatRT, Invasion, p16, p63, StageIV, PaTT4, Size, Gender, GradeG2, SiteNPC</i>
DNN-FS, GLM-FS, and NB-FS	<i>TreatCCRT, p63, Smoke, Nodes, paTT3, TreatRT, Invasion, Age</i>

Table 5.4 shows the provisional optimal feature subsets produced by various feature selection techniques under consideration. To identify the optimal feature subset of the prognosis for the recurrent HNSCC dataset, the HESCA model is used to learn on each feature subset that each feature selection technique provides, and the results are shown in Table 5.5.

5.4 HESCA Model

The HESCA model is implemented on the dataset consisting of an 8-input feature subset generated by the GBM-FS technique by the ensemble. The stacking technique with 10-fold cross-validation is applied to the training set of the optimal feature subset. Tables 5.6, and 5.8 respectively show the performances (with different metrics) of the base classifiers, and meta-classifiers on the training set, while Table 5.7 shows the performance metrics of the 10-fold cross-validation set on the first-level (base) classifiers. To evaluate the HESCA model, the testing dataset is used. Tables 5.10, and Table 5.12 respectively show the performance with different evaluation results of the base classifiers and meta-classifiers on the test set.

Table 5.5: Performance Metrics of HESCA Model on various Feature Subsets

Dataset	Metrics	Feature Selectors				
		GBM-FS	DRF-FS	DNN-FS	GLM-FS	NB-FS
Training set	Accuracy	0.9677	0.9140	0.9032	0.9032	0.9032
	Log-loss	0.1172	0.2854	0.2864	0.2864	0.2864
	AUC	0.9952	0.9677	0.9164	0.9164	0.9164
Testing set	Accuracy	0.9063	0.7813	0.7500	0.7500	0.7500
	Log-loss	0.2959	0.4046	0.5246	0.5246	0.5246
	AUC	0.9251	0.7536	0.7319	0.7319	0.7319

Table 5.5 shows the best feature selection technique for recurrent HNSSC prognosis dataset using the accuracy, log-loss, and AUC on training and test data. The HESCA model with 8-input features provided by the GBM ensemble feature selection technique has the best training accuracy of 96.77% with the least log-loss value (0.1172), and test accuracy of 90.63% with the least log-loss value (0.2959) as compared to the accuracies and log-loss values of the HESCA model with different input features provided by other feature selection techniques used in the study. Similarly, the AUC values of (0.99523) and AUC (0.92512) obtained on the training and testing set respectively are best for the HESCA model with gradient-boosted features (8-input features provided by the GBM-FS technique) as compared to the AUC of the HESCA model with different input features provided by other feature selection techniques on both the training set and test set. This proves that the feature subset provided by the GBM ensemble feature selection (GBM-FS) technique becomes the optimal metrics for recurrent HNSSC prognosis datasets. Now, this optimal feature subset of the data becomes the experimental data on which all other analyses regarding this study were carried out. The training and evaluation results of the HESCA model on GBM-FS features are respectively shown in Table 5.8 and Table 5.12. Figures 5.4 and 5.5 below explain the feature selection technique for recurrent HNSSC prognosis based on the accuracy, log-loss, and AUC of the HESCA model for training and testing data respectively.

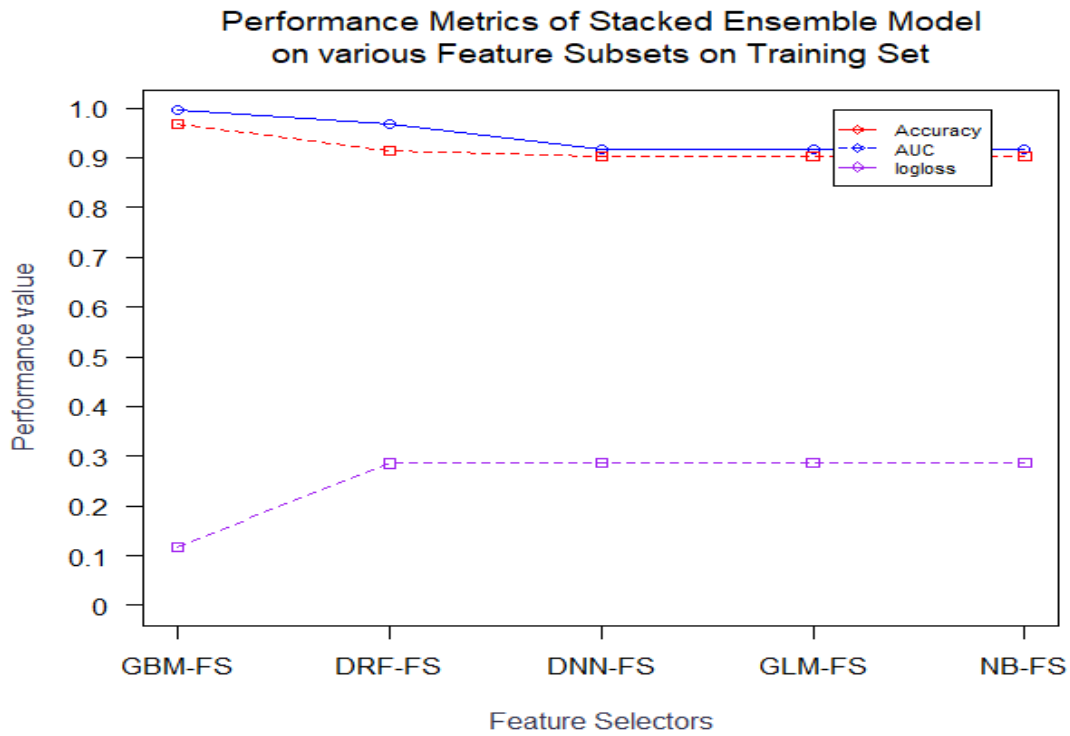


Figure 5.4 Plot of Performance of Feature Selection Techniques on Train Sets

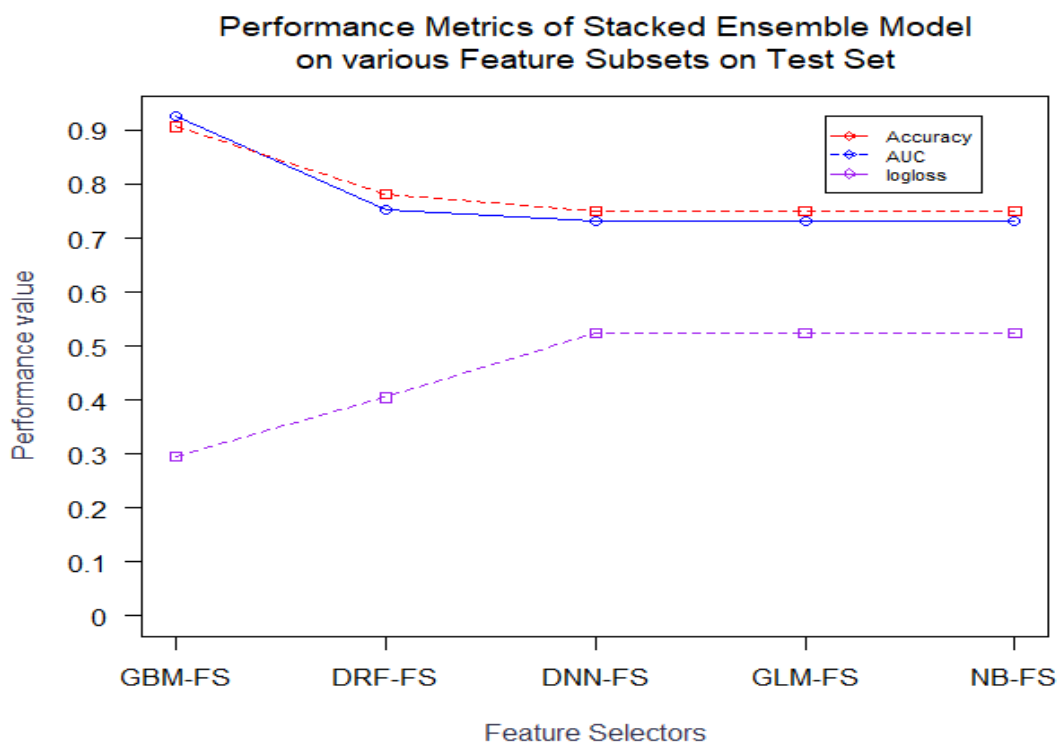


Figure 5.5 Plot of Performance of Feature Selection Techniques on Test Sets

5.4.1 HESCA Model Training on Training Data Set

This section presents the training results of classification models and HESCA model analyses on the training set based on the optimal feature subset from the GBM-FS technique used in this study.

Base Classifiers on Training Set without 10-fold Cross-validation

The base classifiers are learned on the training set (75% of the dataset). Here, no 10-fold cross-validation is implemented on the training data. The accuracy, Area Under the ROC curve (AUC) and other performance metrics were determined. The results generated from each base classifier on training without 10-fold cross-validation are shown in Table 5.6, and the ROC plots of base classifiers without 10-fold cross-validation are shown in Figure 5.6.

Table 5.6 Performance of Base Classifiers on Training Set based on GBM-FS Optimal Feature Subset

Metrics	Base Classifiers				
	GBM	DRF	DNN	GLM	NB
Accuracy	0.9140	0.8280	0.8387	0.7957	0.7957
Log-loss	0.2838	0.5021	0.7200	0.4851	0.5926
Recall	0.9000	0.7222	0.6552	0.6000	0.6000
Specificity	0.9178	0.8533	0.9219	0.8677	0.8788
Precision	0.7500	0.5417	0.7917	0.6250	0.6667
F1-Score	0.8100	0.6191	0.7170	0.6122	0.6316
AUC	0.9330	0.7416	0.8795	0.7769	0.7298

Table 5.6 shows the performance of five base classifiers with GBM-FS optimal feature subset based on the training set used in the study. The results in Table 5.6, shows that the stacked ensemble techniques gave best result. The accuracy, log-loss, and AUC analysis on the hand shows that the GBM base classifier has the highest accuracy value (91.40%) with the least log-loss value (0.2838) and AUC analysis of (0.9330) compared to other base classifiers. Considering the recall, the GBM classifier has the best recall metric of 90.00%. By considering the specificity and precision, the DNN base classifier has the best specificity value (92.19%) and precision value (79.17%). The ROC curve analysis of each base classifier on the training set without 10-fold cross-validation is shown in Figure 5.6 below.

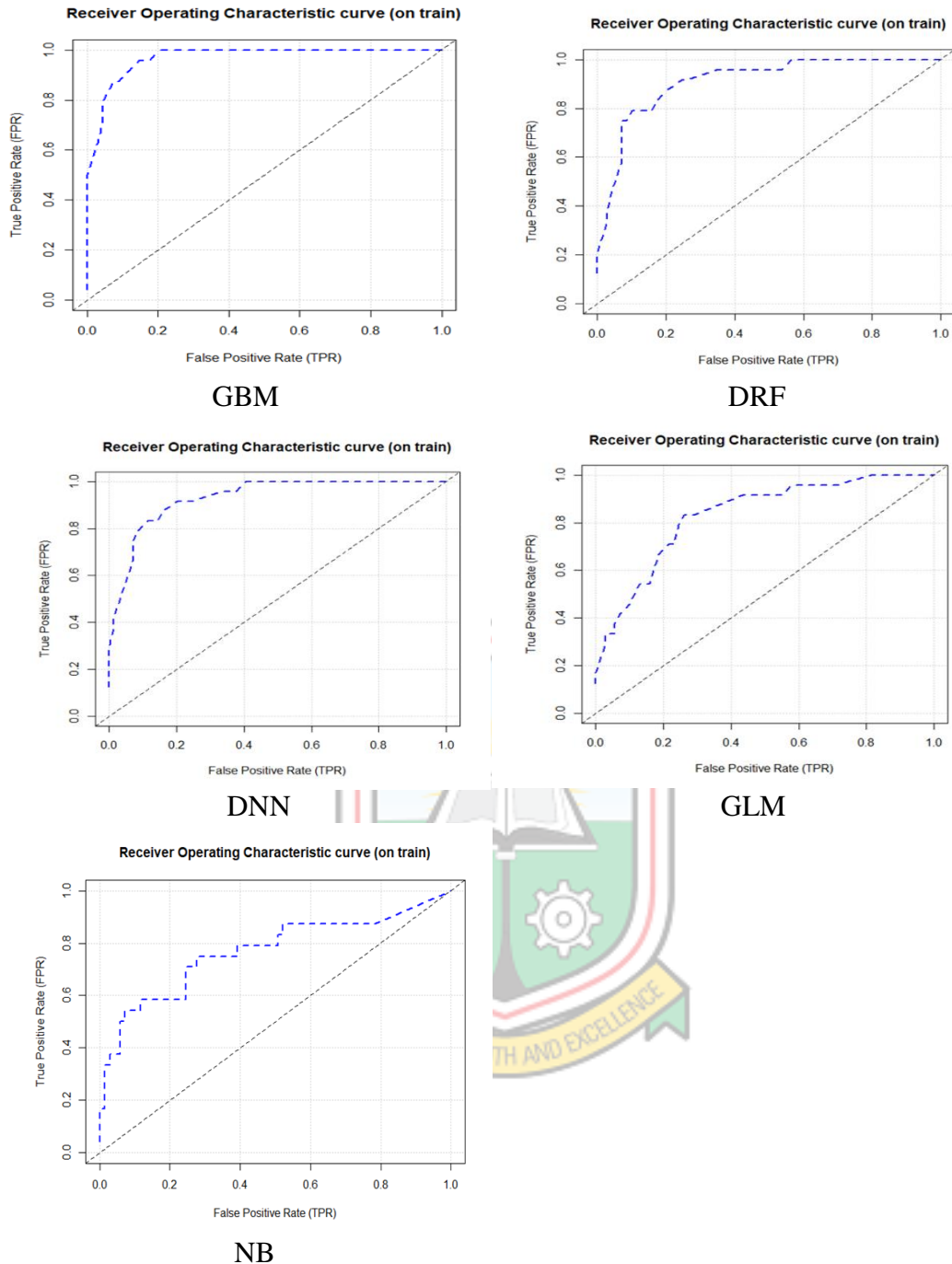


Figure 5.6 ROC Curve Analysis of Base Classifiers on Training Set

First-Level Classifiers on Training Set with 10-fold Cross-validation

The first-level classifiers are the five base classifiers on which a 10-fold cross-validation is implemented based on the training data set (75% of the dataset) used in this study. The cross-validated predictions made by these classifiers serve as input features to the meta-classifiers in a stacking ensemble. These predictions made by these base classifiers along with the original class labels give the level-one data, on which meta-classifiers are learned.

The accuracy, log loss, area under the ROC curve (AUC), and other performance metrics were determined, and the results generated from each first-level classifier models are shown in Table 5.7.

Table 5.7 Performance Metrics of Base Classifiers on 10-Fold Cross-Validation Set

Metrics	10-fold Cross-Validation on Base Classifiers				
	GBM	DRF	DNN	GLM	NB
Accuracy	0.8280	0.8172	0.5484	0.7527	0.6667
Log-loss	0.4564	1.5120	1.7216	0.4683	0.6167
Recall	0.8630	0.8714	0.8462	1.0000	0.9130
Specificity	0.7000	0.6522	0.3333	0.9855	0.4255
Precision	0.9130	0.8841	0.4783	0.6000	0.6087
F1-Score	0.8873	0.8764	0.6112	0.7500	0.7304
AUC	0.7947	0.7597	0.6051	0.7023	0.7289

Table 5.7 shows the performance of the cross-validated predictions of 10-fold cross-validation set on first-level classifiers used in this study. The results in Table 5.6, shows that the stacked ensemble techniques gave best result. The accuracy, log-loss, and AUC analysis on the hand shows that the GBM classifier had the highest accuracy value (82.80%) with the least log-loss value (0.4564) and AUC analysis of (0.7947). Considering the recall, the GLM classifier has the best recall metric (100%). Best precision (0.9130) and specificity (0.9855) were obtained for GBM and GLM classifiers respectively. The log-loss values for DRF and DNN exceed 1, indicating that the predicted probability for a given class is less than $\exp(-1)$ or around 0.368. Therefore, looking at the log-loss of DRF and DNN greater than 1, can be expected in the case that these models only give less than a 36% probability estimate for the actual class.

As a supporting tool for classifying the prognosis of HNSCC recurrence, the cross-validated predictions of these five base classifiers were stacked using each of them as a meta-classifier in a stacking ensemble. The results of each meta classifier on the training set are shown in Table 5.8.

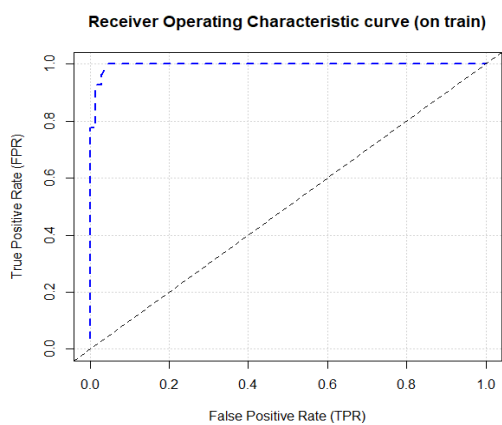
Meta Classifiers on Training Set

Here, each of the base classifiers serves as a meta classifier, each of which is learned on the *level-one* dataset provided by the base classifiers as the first-level classifiers at the first-level of training stacked ensemble. Here, the first-level classifiers were stacked at layer-one of stacking using each base classifier as a meta classifier. Again, here, no 10-fold cross-validation was implemented on meta-classifiers in learning the level-one dataset. The predictions made by these meta-classifiers became the output of the stacked ensemble classification model. The accuracy, area under the ROC curve (AUC), and other performance metrics were determined. The results generated from each meta classifier training are shown in Table 5.8, and their respective ROC plots are shown in Figure 5.7.

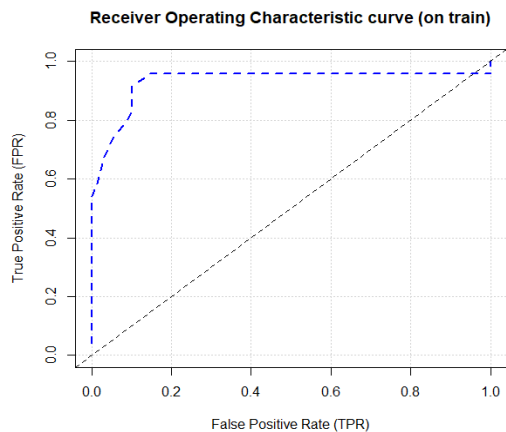
Table 5.8 Performance Metrics of Meta Classifiers on Level-one Training Set

Metrics	Meta Classifiers				
	GBM	DRF	DNN	GLM	NB
Accuracy	0.9677	0.9140	0.9247	0.9355	0.9355
Log-loss	0.1172	0.3139	0.5123	0.2986	0.2038
Recall	0.9000	0.8333	0.8400	0.9091	0.9091
Specificity	1.0000	0.9420	0.9559	0.9437	0.9437
Precision	1.0000	0.8333	0.8750	0.8333	0.8333
F1-Score	0.9474	0.8333	0.8571	0.8696	0.8696
AUC	0.9952	0.9134	0.9200	0.9834	0.9671

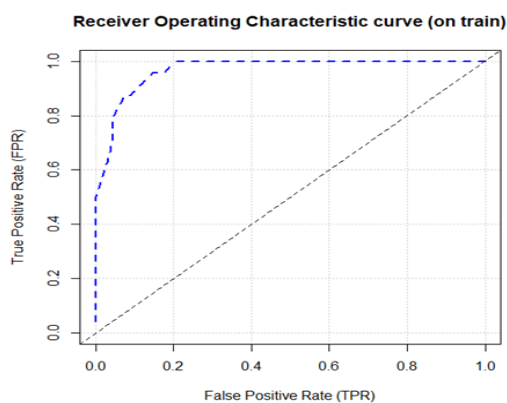
Table 5.8 shows the performance metrics of meta classifiers using each base classifier as a meta classifier and by learning each on the level-one training set used in this study. Considering the performance metrics of these meta-classifiers on the predictions made by the base classifiers in the stacking ensemble consisting of five base classifiers along with the original class labels of the training set in Table 5.8, the best results were obtained using stacked ensemble techniques. The best accuracy (96.77%), log loss (0.1172), specificity (100%), precision (100%), F1-Score (0.9474), and AUC (0.9952) were obtained for the GBM meta-classifier. The best recall value (0.9091) was obtained for GLM meta-classifier and also for the NB meta-classifier. The ROC curves of each meta-classifier on the training set are shown in Figure 5.7 below.



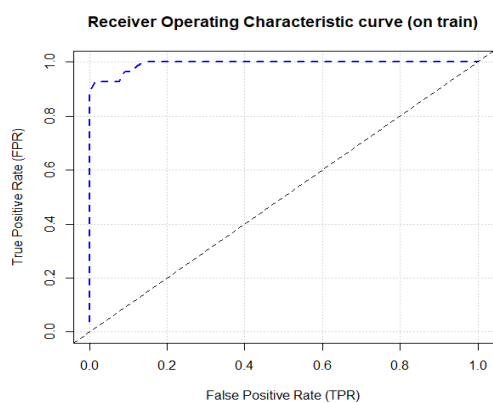
GBM



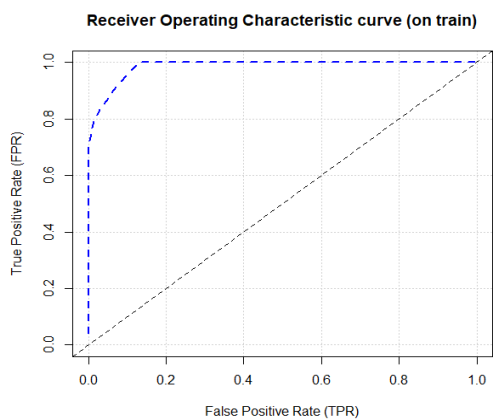
DRF



DNN



GLM



NB



Figure 5.7 ROC Curve Analysis of Meta Classifiers on Training Set

5.4.2 HESCA Model Evaluation on Test Set

This section presents the evaluation results of classification models and the HESCA model analyses on the test set based on the optimal feature subset from the GBM-FS technique.

Base Classifiers on Testing Set

The base classifiers are learned on the training set (75% of the data) and are evaluated on the test data set (25% of the dataset). The accuracy, log-loss, area under the ROC curve (AUC), and other evaluation metrics were determined. The results generated from the evaluation of each base classifier are shown in Table 5.9 and Table 5.10, and their respective ROC plots are shown in Figure 5.8.

Table 5.9 Performance Matrix of Base Classifiers on Test Set

Classifiers	Predicted Outcomes	Actual conditions	
		Recurrence	Non-Recurrence
GBM	Recurrence	8	1
	Non-Recurrence	4	19
DRF	Recurrence	7	2
	Non-Recurrence	6	17
DNN	Recurrence	8	1
	Non-Recurrence	8	15
GLM	Recurrence	4	5
	Non-Recurrence	2	21
NB	Recurrence	8	1
	Non-Recurrence	7	16

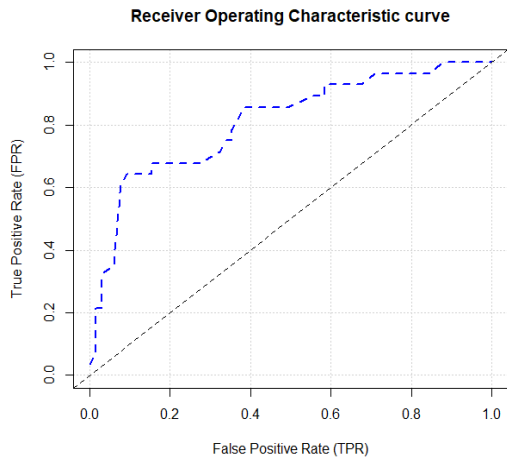
Table 5.9 shows the classification or prediction of HNSCC patients as recurrence or nonrecurrence made by base classifiers. The classification prediction by the GBM model shows that, 8 patients were diagnosis of HNSCC and had curative intent treatment but had recurrence after treatment based on the actual data, and the model also predicts or classifies them as belonging to the recurrence category so they should be considered recurrent patient. Thus, this is a correct prediction or classification. Similarly, 19 means that, there were 19 patients that were diagnosed and treated with curative intent and had nonrecurrence based on the reference data, and the model also classifies them as nonrecurrence. This is also correct prediction or classification. It can also be observed based on the Naive Bayes classification that, the model classifies 8 patients as recurrence and they actually had recurrence; giving correct classification. Similarly, the model classifies or predicted 16 patients to be nonrecurrence and actually; they had nonrecurrence, giving correct

classification. Meanwhile, the NB model predicts 1 patient to have recurrence when actually he/she had nonrecurrence; giving a misclassification. Similarly, the model predicts or classifies 7 patients as nonrecurrence when actually they had recurrence; giving a misclassification. And so on.

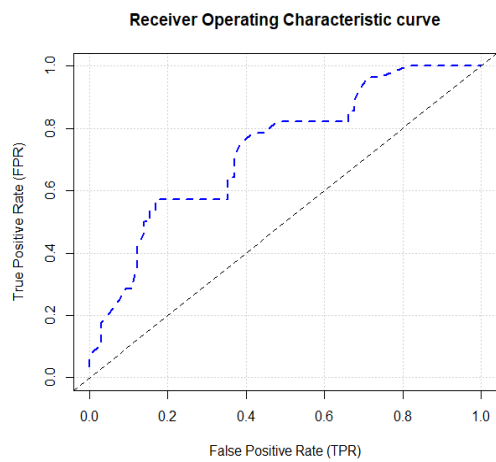
Table 5.10 Performance Metrics of Base Classifiers on Test Set

Metrics	Base Classifiers				
	GBM	DRF	DNN	GLM	NB
Accuracy	0.8438	0.7500	0.7188	0.7813	0.7500
Log-loss	0.4686	0.5156	0.7310	0.5038	0.4948
Recall	0.6667	0.5385	0.5000	0.6667	0.5333
Specificity	0.9500	0.8947	0.9375	0.8077	0.9412
Precision	0.8889	0.7778	0.8889	0.4444	0.8889
F1-Score	0.7619	0.6364	0.6400	0.5333	0.6667
AUC	0.8285	0.7536	0.7778	0.8140	0.8019

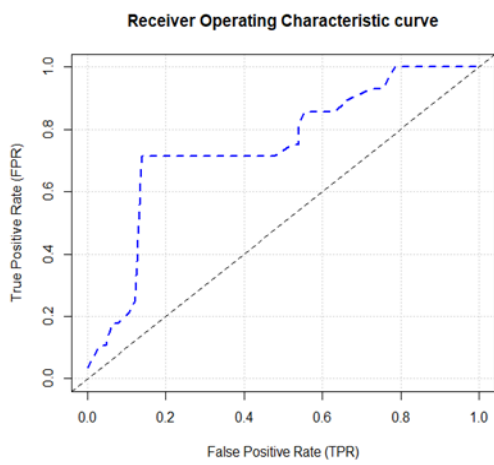
Table 5.10 shows the performance of each of the five base classifiers on test set used in this study. The results in Table 5.10, shows that the stacked ensemble techniques gave best result. The accuracy, log-loss, specificity, and AUC analysis on the other hand shows that the GBM classifier had the highest accuracy of (84.38%) with the least log-loss value (0.4686), specificity value (95.00%), and AUC analysis of (0.8285) compared to other base classifiers. Considering the recall, GBM and GLM classifiers have the same best recall metric (0.6667), and the same best precision metric (88.89%) was recorded for GBM, DNN and GLM classifiers. ROC curve of each base classifier on test set are shown in Figure 5.8 below.



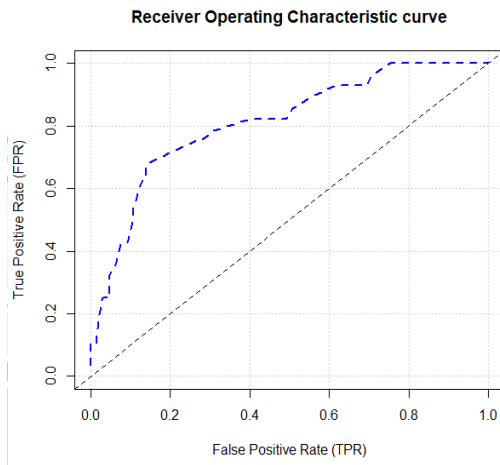
GBM



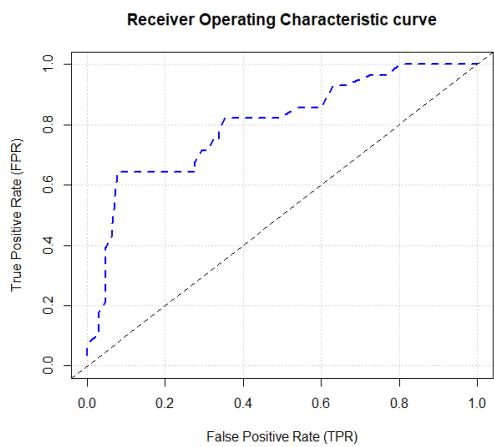
DRF



DNN



GLM



NB

Figure 5.8 ROC Curve Analysis of Base Classifiers on Test set

Meta Classifiers on Test Set

As a tool supporting the classification of the prognosis of HNSCC recurrence; to improve the generalisation ability of the classification model, the cross-validated predictions

provided by each of the five base classifiers are stacked along with the original class labels using each base classifier as a meta-classifier in a stacking ensemble. The meta classifiers are evaluated on the test set (25% of the dataset) to assess the performance of each meta-model on an unseen class labels based on their evaluation metrics. The accuracy, area under the ROC curve (AUC), and other evaluation metrics were determined. The results generated from the evaluation of each meta classifier are shown in Table 5.11 and Table 5.12, and their respective ROC curve analysis are shown in Figure 5.9.

Table 5.11 Classification Matrics of Meta Classifiers on Test Set

Classifiers	Predicted Outcomes	Actual conditions	
		Recurrence	Non-Recurrence
GBM	Recurrence	9	0
	Non-Recurrence	3	20
DRF	Recurrence	9	0
	Non-Recurrence	7	16
DNN	Recurrence	6	3
	Non-Recurrence	1	22
GLM	Recurrence	8	1
	Non-Recurrence	6	17
NB	Recurrence	8	1
	Non-Recurrence	3	20

Table 5.11 shows the prediction or classification made by second-level classifiers on HNSCC patients as recurrence and nonrecurrence. The classification prediction by the GBM model shows that, 9 patients were diagnosis of HNSCC and had curative intent treatment but had recurrence after treatment based on the actual data. The model also predicts or classifies them as belonging to the recurrence category so that they are considered recurrent patient. This confirms that the prediction is indeed correct. On the other hand, a classification prediction of 20 patients signifies that, there were 20 patients that were diagnosed and treated with curative intent and had nonrecurrence based on the reference data, and the model also classifies them as nonrecurrence, indicating that the patients are correctly classified. It can also be observed based on the DNN classification that, the model classifies 6 patients as recurrence and they actually had recurrence; giving correct classification. Similarly, the model classifies or predicted 22 patients to be nonrecurrence

and actually; they had nonrecurrence, giving correct classification. However, the DNN model predicts 3 patients to have recurrence when actually they had nonrecurrence; giving a misclassification. Similarly, the model classifies 1 patient as nonrecurrence when the patient actually had recurrence; giving a misclassification.

Table 5.12 Performance Metrics of Meta Classifiers on Test Set

Metrics	Meta Classifiers				
	GBM	DRF	DNN	GLM	NB
Accuracy	0.9063	0.7813	0.8750	0.7813	0.8750
Log-loss	0.2959	0.5095	0.5854	0.4406	0.4208
Recall	0.7500	0.5625	0.8571	0.5714	0.7273
Specificity	1.0000	1.0000	0.8800	0.9444	0.9524
Precision	1.0000	1.0000	0.6667	0.8889	0.8889
F1-Score	0.8571	0.7200	0.7500	0.6957	0.8000
AUC	0.9251	0.7150	0.8937	0.9179	0.8961

On the test set employed in the work, the performance of the stacked ensemble of meta-classifier models is displayed in Table 5.12. Here, all the five base classifiers are stacked using each of them as a meta-classifier. The results in Table 5.12, shows that the stacked ensemble techniques gave best result. The accuracy, log-loss, F1-Score, and AUC analysis on the other hand shows that the GBM classifier had the highest accuracy of (90.63%) with the least log-loss value (0.2959), F1-Score (85.71%), and AUC analysis of (0.9251) compared to other meta-classifiers. Considering the specificity and precision metrics, the GBM and the DNN meta-classifiers both have the same best metric (100%), and the best recall metric (85.71%) is recorded for the DNN meta-classifier. ROC curve of each meta classifier on test set is shown in Figure 5.9 below.

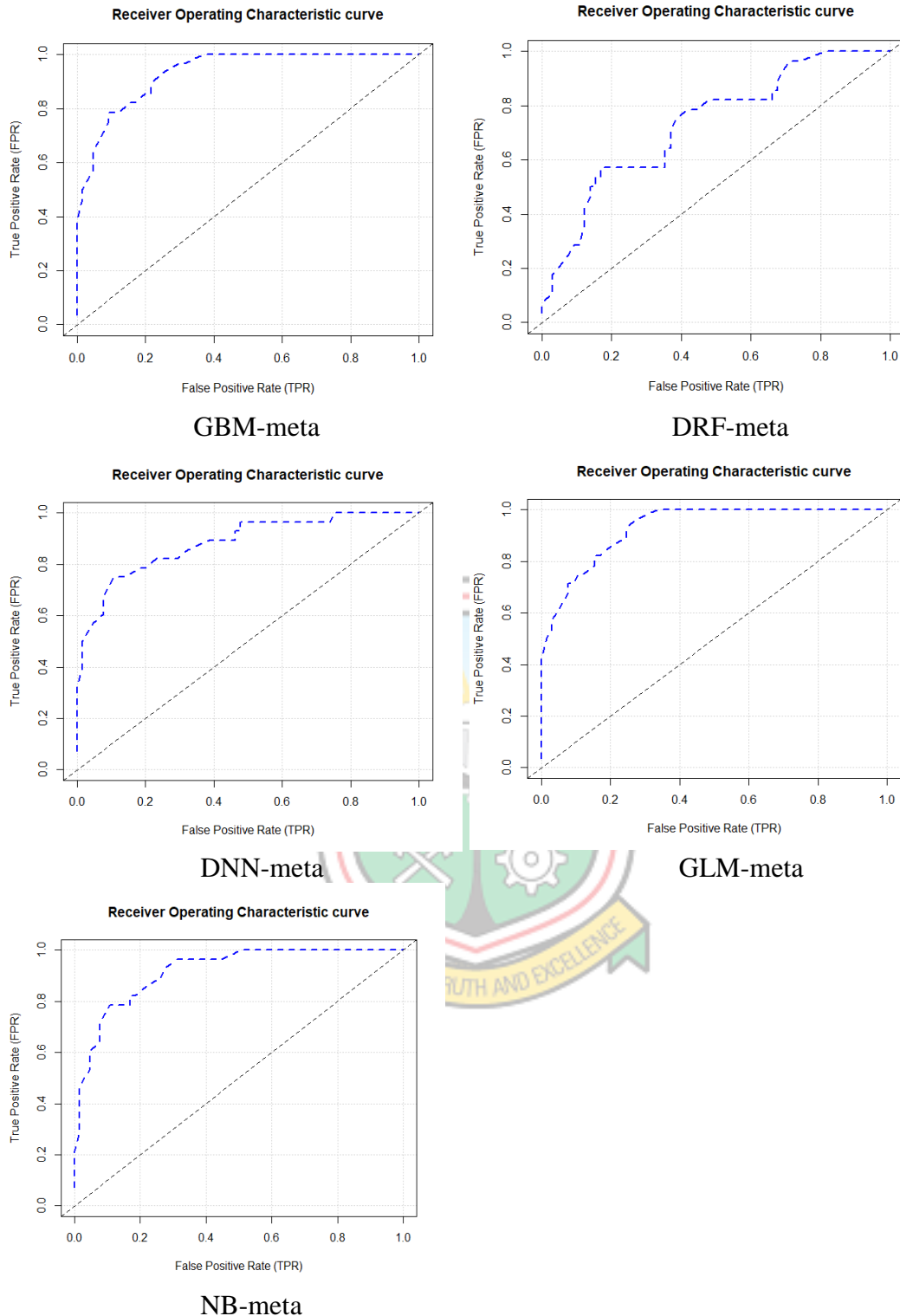


Figure 5.9 ROC Curve Analysis of Meta Classifiers on Test set

5.5 Baseline Stacked Ensemble Classification Techniques

The recurrent HNSCC dataset is tested using the baseline stacked ensemble classification techniques in existence. Two baseline stacked ensemble techniques; which are the stacked ensemble consisting of two base classifiers, and the stacked ensemble consisting of three

base classifiers, including one state-of-the-art stacked ensemble technique which is a multi-level stacked ensemble technique consisting of four base classifiers are adapted, trained and tested on the training and the testing set used in this study, and their results are verified, compared and discussed in Section 5.6.

5.5.1 Stacked Ensemble Model (GBM & DRF) with GLM Meta-Classifier

A baseline stacked ensemble technique having two base classifiers (GBM and DRF) is learned with 10-fold cross-validation on the training set (75% of the dataset) and evaluated on the testing set (25% of the dataset). Here, the GBM and DRF serve as base classifiers and are stacked using the GLM as a meta-classifier. The accuracy, area under the ROC curve (AUC), and other evaluation metrics were determined. The metrics or results generated from this baseline stacked ensemble model (here, termed stacked ensemble-GLM1) based on the training set and the test set are shown in Table 5.14, and its performance matrix is shown in Table 5.13.

Table 5.13 Classification Matrix for Stacked Ensemble_GLM1 on Training and Test Sets: Stack GBM and DRF using GLM

Classifiers	Predicted Outcomes	Actual conditions	
		Recurrence	Non-Recurrence
Training set	Recurrence	20	6
	Non-Recurrence	4	65
Test set	Recurrence	9	0
	Non-Recurrence	15	8

Table 5.13 shows the prediction or classification made by baseline stacked ensemble-GLM1 on HNSCC patients as recurrence and nonrecurrence. Based on the training output, it can also be observed that the classification model classifies 20 patients as recurrence and actually they had recurrence; giving correct classification or prediction. On the other hand, the model classifies or predicts 65 patients to have nonrecurrence and actually they had nonrecurrence; giving correct classification. However, the model predicts 6 patients to have recurrence when actually they had nonrecurrence; giving a misclassification. Similarly, the model classifies 4 patients as nonrecurrence when actually they had recurrence; giving a misclassification. Similar interpretation goes for classification matrix based on the test output.

Table 5.14 Performance Metrics of Stacked Ensemble_GLM1 on Training and Test Set: Stack GBM and DRF using GLM

Metrics	Stacked Ensemble_GLM1	
	Training Set	Test Set
Accuracy	0.9140	0.5313
Log-loss	0.3450	0.7880
Recall	0.8333	0.3750
Specificity	0.9420	1.0000
Precision	0.8333	1.0000
F1-Score	0.8333	0.5455
AUC	0.9073	0.4493

Table 5.14 shows the performance of the training and evaluation metrics of the Stacked Ensemble-GLM1 model having two base classifiers (GBM and DRF) with the GLM as a meta-classifier on training and testing sets. The model recorded the accuracy (91.40%), log-loss (0.3450), recall (83.33%), specificity (94.20%), precision (83.33%), F1-Score (83.33%), and AUC analysis of (0.90731) approximately on training set. The evaluation metrics of the model on test set were determined, with the accuracy (53.13%), log-loss (0.7880), recall (37.50%), specificity (100%), precision (100%), F1-Score (54.55%), and the AUC analysis of (0.4493) approximately.

5.5.2 Stacked Ensemble Model (GBM, DRF & DNN) with GLM Meta-Classifier

A baseline stacked ensemble technique having three base classifiers (GBM, DRF, and DNN) is learned on the training set (75% of the dataset) with 10-fold cross-validation and evaluated on the test set (25% of the dataset). Here, the GBM, DRF and DNN serve as base classifiers and are stacked using the GLM as a meta-classifier. The accuracy, area under the ROC curve (AUC), and other evaluation metrics were determined. The results generated from this baseline stacked ensemble model (here, termed stacked ensemble-GLM2) based on the training set and the test set are shown in Table 5.16, and its classification matrix is shown in Table 5.15.

Table 5.15 Classification Matrix for Stacked Ensemble_GLM2 on Training and Test Sets: Stack GBM, DRF and DNN using GLM

Classifiers	Predicted Outcomes	Actual conditions	
		Recurrence	Non-Recurrence
Training set	Recurrence	18	6
	Non-Recurrence	1	68
Test set	Recurrence	7	2
	Non-Recurrence	1	22

Table 5.15 shows the prediction or classification made by baseline stacking ensemble-2 on HNSCC patients as recurrence and nonrecurrence. Based on the training results, it can also be observed that the classification model classifies 18 patients as recurrence and actually they had recurrence; giving correct classification or prediction. On the other hand, the model classifies or predicts 68 patients to have nonrecurrence and actually they had nonrecurrence; giving correct classification. Meanwhile, the model predicts 6 patients to have recurrence when actually they had nonrecurrence; giving a misclassification. Similarly, the model predicts or classifies 1 patient as nonrecurrence when actually he/she had recurrence; giving a misclassification or incorrect prediction.

Table 5.16 Performance Metrics of Stacked Ensemble_GLM2 on Training and Test Sets: Stack GBM, DRF and DNN using GLM

Metrics	Stacked Ensemble_GLM2	
	Training Set	Test Set
Accuracy	0.9247	0.9063
Log-loss	0.3075	0.4267
Recall	0.9474	0.8750
Specificity	0.9189	0.9167
Precision	0.7500	0.7778
F1-Score	0.8372	0.8235
AUC	0.9043	0.8623

Table 5.16 shows the performance of training and evaluation metrics of the Stacked Ensemble-GLM2 model having two base classifiers (GBM, DRF and DNN) with GLM as a meta classifier on the training and testing sets. The model records the accuracy (92.47%),

log-loss (0.3075), recall (94.74%), specificity (91.89), precision (75.00%), F1-Score (83.72), and the AUC analysis of (0.9043) approximately on the training set. The evaluation metrics of the model on test set were determined, with accuracy (90.63%), log-loss (0.4267), recall (87.50%), specificity (91.67%), precision (77.78%), F1-Score (82.35), and the AUC analysis of (0.8623) approximately.

5.5.3 State-of-the-Art (SA) Stacked Ensemble Model

The State-of-the-Art (SA) stacked ensemble technique having four base classifiers (GBM, DRF, DNN and GLM) is learned on the training set (75% of the dataset) with 10-fold cross-validation and evaluated on the test set (25% of the dataset). Here, the GBM, DRF, DNN and GLM serve as base classifiers, and each of which serves as a meta-classifier in a stacking ensemble. The accuracy, area under the ROC curve (AUC), and other evaluation metrics were determined. The results generated from this SA stacked ensemble model based on the training set and the testing set are shown in Table 5.18 and Table 5.19 respectively, and its performance matrix is shown in Table 5.17.

Table 5.17 Classification Matrix for State-of-the-Art (SA) Stacked Ensemble Model on Test Set

Classifiers	Predicted Outcomes	Actual conditions	
		Recurrence	Non-Recurrence
GBM	Recurrence	8	1
	Non-Recurrence	4	19
DRF	Recurrence	9	0
	Non-Recurrence	11	12
DNN	Recurrence	5	4
	Non-Recurrence	3	20
GLM	Recurrence	4	5
	Non-Recurrence	1	22

Table 5.17 shows the prediction or classification made by State-of-the-Art (SA) for classifying HNSCC patients as recurrence and nonrecurrence. It can also be observed that the stacked ensemble model with GBM meta-model classifies 8 patients as recurrence and actually had recurrence; giving correct classification or prediction. Similarly, the model classifies 19 patients to have nonrecurrence and actually they had nonrecurrence; giving

correct classification. Meanwhile, the model predicts 1 patient to have recurrence when actually he/she had nonrecurrence; giving a misclassification or incorrect prediction. Similarly, the model predicts or classifies 4 patients as nonrecurrence when actually they had recurrence; giving a misclassification or incorrect prediction. Also, the classification model with DRF meta-model classifies 9 patients as recurrence and actually had recurrence; giving correct classification or prediction. Similarly, the model classifies or predicts 12 patients to have nonrecurrence and actually they had nonrecurrence; giving correct classification. Meanwhile, the model classifies 0 (or no) patient as recurrence when actually 0 or no patient are classified as nonrecurrence; giving a misclassification or incorrect prediction. Similarly, the model predicts or classifies 11 patients as nonrecurrence when actually they had recurrence; giving a misclassification or incorrect prediction. And similar interpretation goes for other meta models.

Table 5.18 Performance Metrics of State-of-the-Art (SA) Stacked Ensemble Model on Training Set

Metrics	Meta Classifiers			
	GBM	DRF	DNN	GLM
Accuracy	0.9355	0.9032	0.8602	0.8925
Log-loss	0.2667	0.3449	2.5134	0.2993
Recall	0.9091	0.8261	0.6897	0.7333
Specificity	0.9437	0.9286	0.9375	0.9683
Precision	0.8333	0.7917	0.8333	0.9167
F1-Score	0.8696	0.8085	0.7547	0.8148
AUC	0.9816	0.9058	0.8693	0.9164

Table 5.19 Performance Metrics of State-of-the-Art (SA) Stacked Ensemble Model on Test Set

Metrics	Meta Classifiers			
	GBM	DRF	DNN	GLM
Accuracy	0.8438	0.6563	0.7813	0.8125
Log-loss	0.4406	0.5917	0.8348	0.4888
Recall	0.6667	0.4500	0.6250	0.8000
Specificity	0.9500	1.0000	0.8333	0.8148
Precision	0.8889	1.0000	0.5556	0.4444
F1-Score	0.7619	0.6207	0.5883	0.5714
AUC	0.9179	0.6208	0.6715	0.8333

Table 5.18 and Table 5.19 respectively show the performance of the State-of-the-Art (SA); a stacking ensemble having four base classifiers, including GBM, RF, DNN, and GLM on the training and the testing data used in this study. Here, each base classifier is used as a meta-classifier in a stacking ensemble of four base classifiers. Looking at the results in Table 5.18 and Table 5.19, for the training and the testing sets used in this research, the best results were obtained using stacking ensemble techniques. Based on the training set in Table 5.18, the best accuracy (93.55%) with the least log-loss (0.2667), recall (90.91%), F1-Score (86.96%), and the AUC analysis of (0.9816) were recorded for the GBM meta-classifier compared to other meta-classifiers. The best specificity (96.83%) and precision (91.67%) were obtained for the GLM meta-classifier. Considering the testing set in Table 5.20, the GBM meta-classifier has the highest accuracy value (90.63%) with the least log-loss metric (0.4406), F1-Score (76.19%), and AUC analysis of (0.9179) compared to that of other meta-classifiers. The DRF meta-classifier has the best metric (100%) for both specificity and precision; and the best recall metric (80.00%) was recorded for the GLM meta-classifier. The log-loss values for DNN meta-classifier is 2.5134, which does not lie between 0 and 1 inclusive, indicating that the predicted probability for a given class is less than $\exp(-1)$ or around 0.368. Therefore, it can be expected in the case that this model only gives less than a 36% probability estimate for the actual class.

5.6 Comparative Analysis of the Results

Here, the results obtained from base models, the HESCA models, and the stacked ensemble models on both the training and testing sets as discussed in Section 5.4 and Section 5.5 respectively are compared and discussed. Table 5.20 compares performance metrics of the HESCA model with the full-input features and the HESCA model with the 8-input features. Table 5.21, and Table 5.22 compare performance metrics of base classifiers and the HESCA model all with the GBM-FS features (8-input features) on the training data and the testing data respectively. Table 5.24 compares performance metrics of baseline stacked ensemble models and the HESCA Model with the GBM-FS features on the training and the testing sets. Table 5.24 compares performance metrics of the state-of-the-art (SA) stacked ensemble model and the HESCA model with the GBM-FS features performance on the training and testing sets. And Table 5.25 summarises the comparison of baseline stacked ensemble models and the state-of-the-art model with the HESCA model based on the GBM-FS features.

Table 5.20 Comparison of HESCA model with full input features and HESCA model with GBM-FS features

Metrics	Training Set		Testing Set	
	HESCA Model on Original Training Set	HESCA Model on GBM-FS	HESCA Model on Original Test Set	HESCA Model on GBM-FS
Accuracy	0.3441	0.9677	0.3438	0.9063
Log-loss	0.8025	0.1172	1.0435	0.2959
Recall	0.3023	0.9000	0.3846	0.7500
Specificity	0.8571	1.0000	0.1667	1.0000
Precision	0.9630	1.0000	0.6667	1.0000
F1-Score	0.4602	0.9474	0.4878	0.8571
AUC	0.4879	0.9952	0.4364	0.9251

Table 5.20 shows the performance of the HESCA model with full-input features and the HESCA model with 8-input features. The performance metrics including the accuracy, log-loss, recall, specificity, precision, F1-Score, and AUC were obtained on both the training set (to assess model training performance) and the testing set (to evaluate model performance on an unseen labels). In terms of accuracy, the accuracy of (96.77%) with the log-loss (0.1172) on the training set and (90.63%) with the log-loss (0.2959) on test set were

obtained for the HESCA model with 8-input features (optimal feature subset) as compared to the accuracy of (34.41%) with the log-loss (0.8025) on the training set and (34.38%) with the log-loss (1.0435) on the testing set obtained for HESCA model with full-input features. Similar to AUC analysis, the value of (0.9952) on the training set and the value of (0.9251) on the testing set were obtained for the HESCA model with 8-input features, compared to the AUC value of (0.4879) on the training set and the value (0.4364) on the testing set obtained for the HESCA model with full-input features. The recall value (90.00%), specificity value (100%), precision value (100%) and F1-Score (94.74%) were obtained for the HESCA model with 8-input features based on the training set, as compared to the recall value (30.23%), specificity value (85.71%), precision value (96.30%) and F1-Score (46.02%) obtained for the HESCA model with full-input features based on the training set. Similarly, the recall value of (75.00%), specificity value (100%), precision value (100%) and F1-Score (85.71%) were obtained for the HESCA model with 8-input features based on the testing set as compared to the recall value (38.46%), specificity value (16.67%), precision value (66.67%) and F1-Score (48.78%) obtained for the HESCA model with full-input features based on the testing set. It can be deduced that the HESCA model with 8-input on gradient-boosted features outperforms the HESCA model with full-input features.

Table 5.21 Comparison of Base Models and HESCA Model Performance on Training Data based on GBM-FS

Metrics	Base Models					Stacked Model
	GBM	DRF	DNN	GLM	NB	HESCA Model
Accuracy	0.9140	0.8280	0.8387	0.7957	0.7957	0.9677
Log-loss	0.2838	0.5021	0.7200	0.4851	0.5926	0.1172
Recall	0.9000	0.7222	0.6552	0.6000	0.6000	0.9000
Specificity	0.9178	0.8533	0.9219	0.8677	0.8788	1.0000
Precision	0.7500	0.5417	0.7917	0.6250	0.6667	1.0000
F1-Score	0.8100	0.6191	0.7170	0.6122	0.6316	0.9474
AUC	0.9330	0.7416	0.8795	0.7769	0.7298	0.9952

Table 5.21 shows the comparative performance metrics of base (standalone) models and the HESCA model on the training data based on the 8-input dataset. It can be observed that the HESCA model had the best accuracy (96.77%) with the least log-loss value of (0.1172), specificity (100%), precision (100%), F1-Score (94.74%), and AUC (0.9952) compared to

the base models. It is interesting to observe that the best recall value of (90.00%) was recorded for both the GBM base model and the HESCA model. The information in Table 5.21 is presented graphically in Figure 5.10 below. In general, it can be deduced that the HESCA model outperforms base models, based on the 8-input dataset of the training data used in the study.

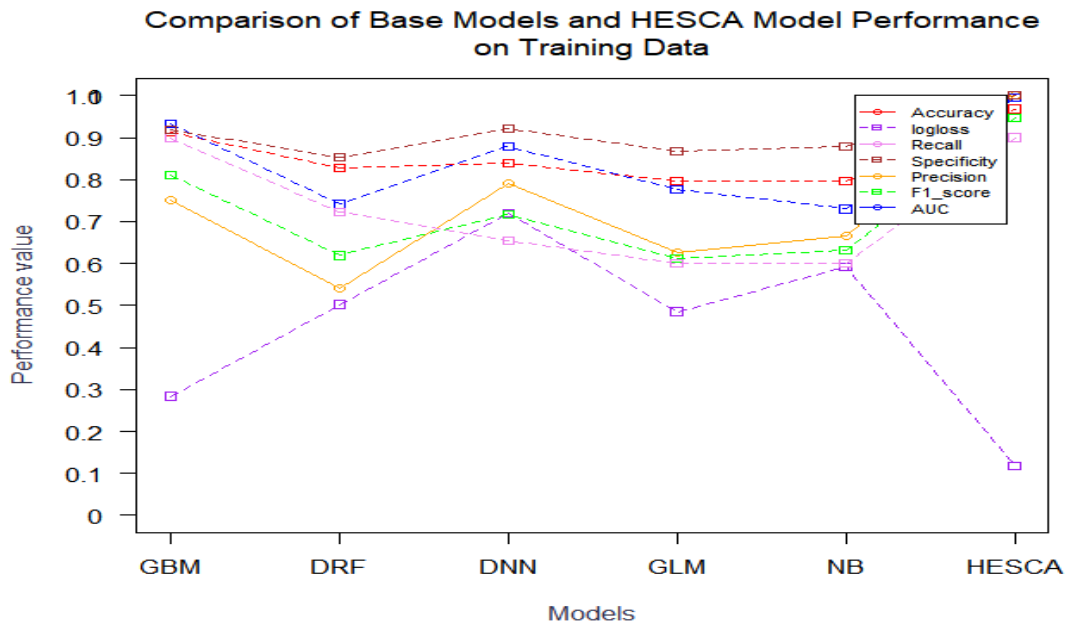


Figure 5.10 Graph of Base Models versus HESCA Model on Training Set

Table 5.22 Comparison of Base Models and HESCA Model Performance on Test Data based on GBM-FS

Metrics	Base Models					Stacked Model
	GBM	DRF	DNN	GLM	NB	HESCA Model
Accuracy	0.8438	0.7500	0.7188	0.7813	0.7500	0.9063
Log-loss	0.4686	0.5156	0.7310	0.5038	0.4948	0.2959
Recall	0.6667	0.5385	0.5000	0.6667	0.5333	0.7500
Specificity	0.9500	0.8947	0.9375	0.8077	0.9412	1.0000
Precision	0.8889	0.7778	0.8889	0.4444	0.8889	1.0000
F1-Score	0.7619	0.6364	0.6400	0.5333	0.6667	0.8571
AUC	0.8285	0.7536	0.7778	0.8140	0.8019	0.9251

Table 5.22 shows the comparative performance metrics of base models and the HESCA model on the 8-input testing data. It can be observed that the HESCA model has the best

accuracy value (90.63%) with the least log-loss value (0.2959), recall (75.00%), specificity (100%), precision (100%), F1-Score (85.71%), and AUC analysis of (0.9251) as compared to the base models. The information in the Table 5.22 is presented graphically in the Figure 5.12 below. In effect, it can be deduced that the HESCA model outperforms the base models based on the 8-input testing set of the dataset used in this study, indicating better predictions on patients with recurrent HNSCC prognosis.

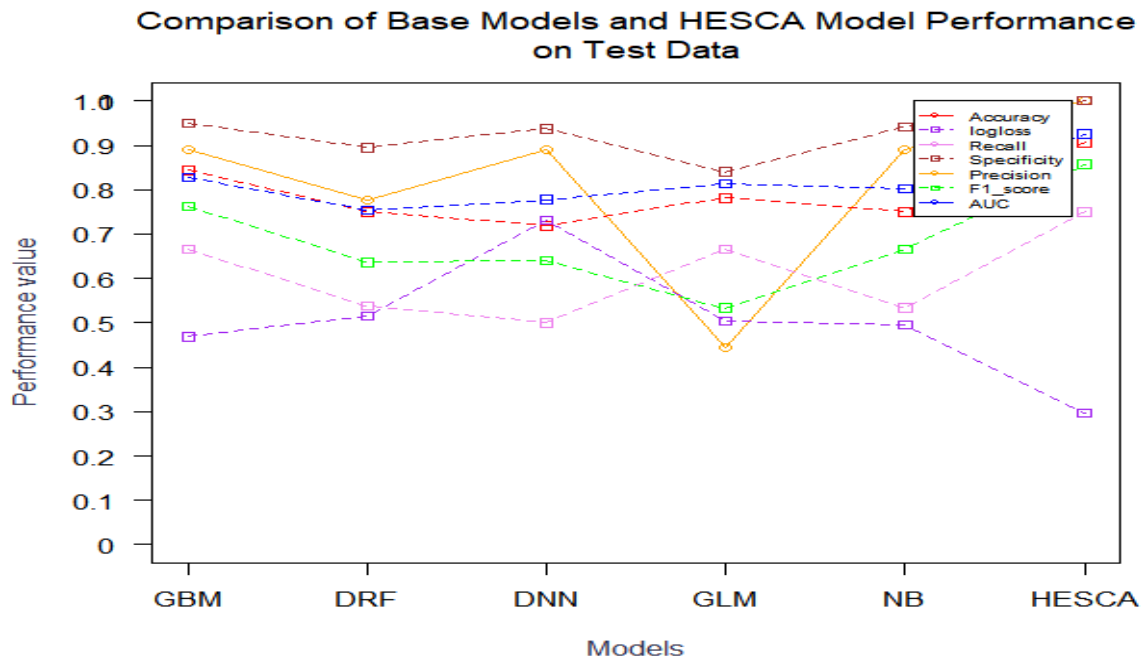


Figure 5.11 Plot of Base Models compared with HESCA Model on Test Set

Table 5.23 Comparison of Baseline Stacked Ensemble Models and HESCA Model Performance on Training and Test Sets

Metrics	Training Set			Test Set		
	Model-GLM1	Model-GLM2	HESCA Model	Model-GLM1	Model-GLM2	HESCA Model
Accuracy	0.9140	0.9247	0.9677	0.5313	0.9063	0.9063
Log-loss	0.3450	0.3075	0.1172	0.7880	0.4267	0.2959
Recall	0.8333	0.9474	0.9000	0.3750	0.8750	0.7500
Specificity	0.9420	0.9189	1.0000	1.0000	0.9167	1.0000
Precision	0.8333	0.7500	1.0000	1.0000	0.7778	1.0000
F1-Score	0.8333	0.8372	0.9474	0.5455	0.8235	0.8571
AUC	0.9073	0.9043	0.9952	0.4493	0.8623	0.9251

Table 5.23 compares the training and evaluation metrics of baseline stacked ensemble models and the HESCA model based on the training and the testing data used in this study. Considering the performance metrics of stacked ensemble models obtained on the training data, it can be observed that, the HESCA model has the best accuracy (96.77%) with the least log-loss value (0.1172), specificity (100%), precision (100%), F1-Score (94.74%), and AUC of (0.9952) as compared to the baseline stacked ensemble models. However, the best recall value (94.74%) was obtained for the baseline stacked ensemble model (Model-GLM2) having three base classifiers with the GLM meta-classifier.

Then, considering the performance metrics obtained on the test data, it can be observed that the HESCA model has the best F1-Score (85.71%) and AUC (0.9251) with the least log-loss value (0.2959) as compared to the baseline stacked ensemble models. Meanwhile, the best specificity value of (100%) and precision (100%) were obtained for both the HESCA model and the baseline stacked ensemble model (Model-GLM1) having two base classifiers with the GLM meta-classifier. The best recall value of (87.50%) was obtained for the baseline stacked ensemble model (Model-GLM2) having three base classifiers with the GLM meta-classifier. Considering the accuracy value for the HESCA model and the baseline stacked ensemble model (Model-GLM2), both obtained the best accuracy of (90.63%), but the HESCA model had the least log-loss value of (0.2959) as compared to the high log-loss value (0.4267) for the Model-GLM2. The selection of the best model is based on the test error (model with the least test error). Thus, based on the log-loss values of all the models in the Table 5.23, the HESCA model that had the least log-loss is considered the best model.

In effect, using the GBM as a meta-classifier in a stacking ensemble having five base classifiers provides the best accuracy (90.63%) with least log-loss value (0.2959) on the test set as used in the study. The graph of this is shown in Figure 5.12 below in red line.

Table 5.24 Comparison of State-of-the-art (SA) Stacked Ensemble Model and HESCA Model Performance on Training and Test Sets

Metrics	Training Set		Testing Set	
	State-of-the-art (SA)	HESCA Model	State-of-the-art (SA)	HESCA Model
Accuracy	0.9355	0.9677	0.8438	0.9063
Log-loss	0.2667	0.1172	0.4406	0.2959
Recall	0.9091	0.9000	0.6667	0.7500
Specificity	0.9437	1.0000	0.9500	1.0000
Precision	0.8333	1.0000	0.8889	1.0000
F1-Score	0.8696	0.9474	0.7619	0.8571
AUC	0.9816	0.9952	0.9179	0.9251

Table 5.24 compares the training and evaluation metrics of the state-of-the-art (SA) stacked ensemble model and HESCA model based on the training and test data used in this study. Here, the SA stacked ensemble model is a stacked ensemble model having four base classifiers including GBM, DRF, DNN, and GLM, with GBM meta-classifier whereas HESCA model is a stacked ensemble model having five base classifiers including GBM, DRF, DNN, GLM, and NB with the GBM meta-classifier. Considering the performance metrics obtained for each model on the training data, it can be observed that the HESCA model had the best accuracy (96.77%) with the least log-loss value (0.1172), specificity (100%), precision (100%), F1-Score (94.74%), and AUC analysis of (0.9952) as compared to the SA stacked ensemble model having the accuracy (93.55%) with a high log-loss value (0.2667), specificity (94.37%), precision (83.33%), F1-Score (86.96%), and AUC (0.9816). Meanwhile, the best recall value (90.91%) was obtained for the SA stacked ensemble model as compared to the HESCA model with (90.00%). Based on the accuracy and log-loss metrics obtained for each model on the training set, HESCA model outperforms the SA stacked ensemble model.

On the other hand, considering the performance metrics obtained for each model on the test data, it can be observed that, the HESCA model had the best accuracy (90.63%) with the least log-loss value (0.2959) compared to the SA stacked ensemble model whose accuracy is (84.38%) with a high log-loss value (0.4406). The best recall (75.00%), specificity (100%), precision (100%), F1-Score (85.71%), and AUC (0.9251) were obtained for the HESCA model as compared to those of the SA stacked ensemble model with the recall

(66.67%), specificity (95.00%), precision (88.89%), F1-Score (76.19%), and AUC of (0.9179).

In effect, using the GBM as a meta-classifier in a stacking ensemble having five base classifiers provides the best accuracy (90.63%) with the least log-loss value (0.2959) on the test set as used in the study. The graph of this is shown in the Figure 5.13 below.

Table 5.25 Summary of the comparison of Baseline Stacked Ensemble Models and State-of-the-Art Model with HESCA Model on Test Set

Metrics	Model-GLM1	Model-GLM2	State-of-the-Art Model	HESCA Model
Accuracy	0.5313	0.9063	0.8438	0.9063
Log-loss	0.7880	0.4267	0.4406	0.2959
Recall	0.3750	0.8750	0.6667	0.7500
Specificity	1.0000	0.9167	0.9500	1.0000
Precision	1.0000	0.7778	0.8889	1.0000
F1-Score	0.5455	0.8235	0.7619	0.8571
AUC	0.4493	0.8623	0.9179	0.9251

Table 5.25 shows the summary of the comparison of baseline stacked ensemble models and the state-of-the-art model with the HESCA model on the testing set. It can be observed that the performance metrics in terms of the accuracy (90.63%) with the least log loss value (0.2959) and AUC of (0.9251) were obtained for the HESCA model, indicating its outperformance compared to other stacked ensemble models considered in the study. The information in the Table 5.25 is presented graphically in the Figure 5.13 below.

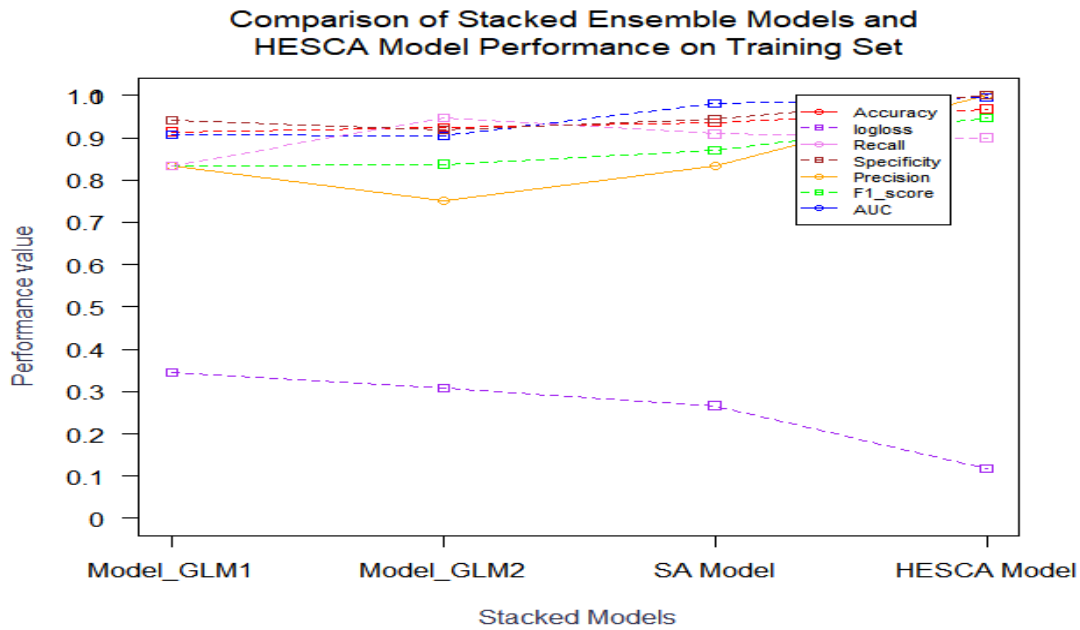


Figure 5.12 Graph of Stacked Ensemble Models compared with HESCA Model on Training Set

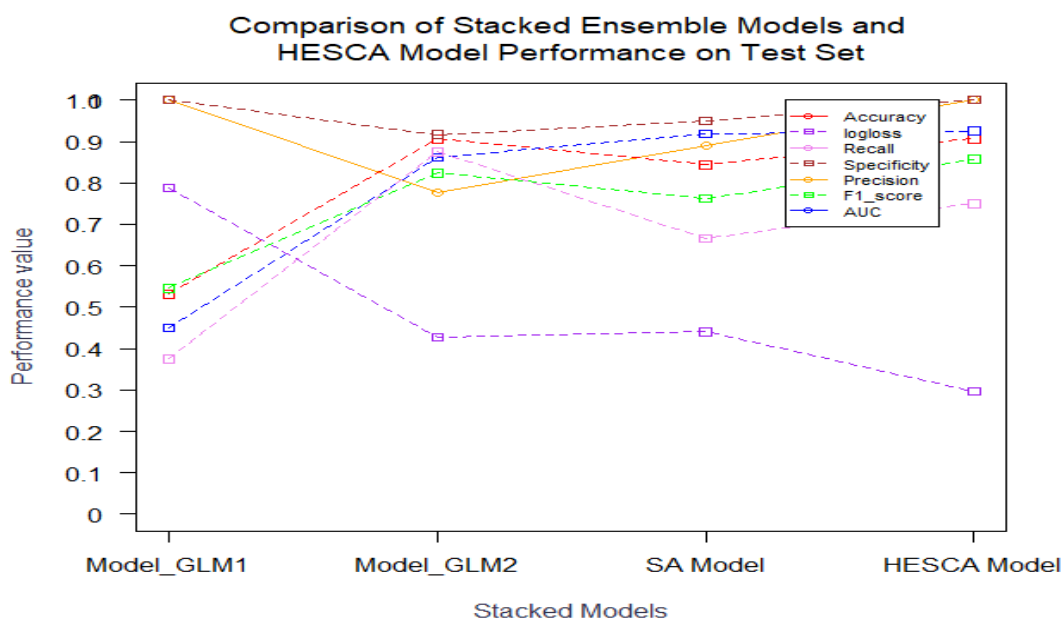


Figure 5.13 Graph of Stacked Ensemble Models compared with HESCA Model on Test Set

5.7 HESCA Classification Model Prediction

The classification model's predictions based on the optimal feature subset are compared with the actual predictions of the dataset. This is shown in the Table 5.33 below, and the goodness of fit-test of the model shown in the Figure 5.14.

Table 5.26 HESCA Classification Model Prediction

Recurrence (1)	Non-Recurrence (0)	Label	max_prob
0.6831017	0.3168983	1	1
0.8927578	0.1072422	1	1
0.9091983	0.0908017	1	1
0.7803982	0.2196018	0	1
0.9188790	0.0811210	1	1
0.9086598	0.0913402	1	1

Table 5.26 shows the predictions made by the HESCA model; where the nonrecurrence column and recurrence column show the probability of having nonrecurrence and the probability of having recurrence respectively. The label column is the actual class label in the test data, and the max_prob column indicates the maximum probability predicted by the model. The cut-off or threshold value is 0.5 indicating that, when the probability value is greater than the cut-off value, the model is predicting class 1 (recurrence) otherwise, it is predicting class 0 (nonrecurrence). The '0' indicated that patients have no recurrence and '1' indicates that patients have recurrence. Now, based on the data, the model finds that the maximum probability (0.6831017) is for the first class where the recurrence is '1' and that is why it takes the value '1'. That is, the model predicted for class 1 that, the patient be classified to have recurrence and actually the patient is classified to have recurrence. Similarly, the maximum probability (0.9086598) is higher for the class 1 and that is why the model predicted '1'. That is, the model predicted for class 1 that, the patient be classified to have recurrence and actually the patient is classified to have recurrence. But a clear look at the fourth prediction, where the maximum probability (0.7803982) is for the first column and that is why the value is '1', which means the model predicted that patients be classified to have recurrence when actually the patient is classified to have nonrecurrence. So, this in the way, is a misclassification.

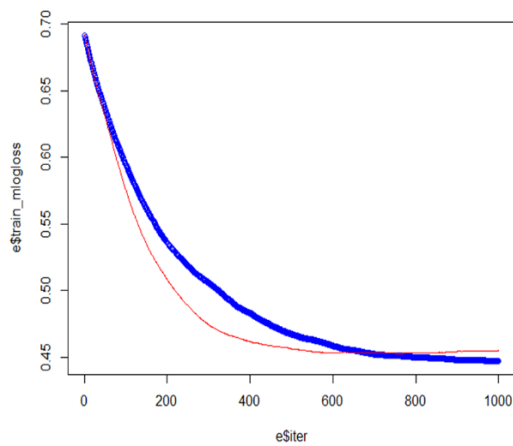


Figure (a)

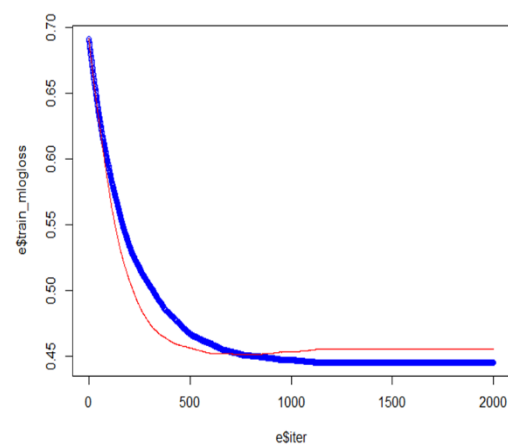


Figure (b)

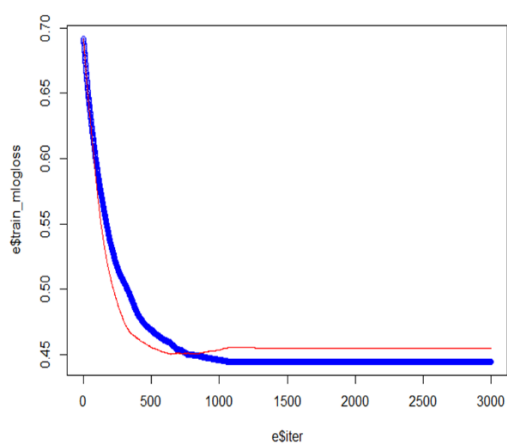


Figure (c)

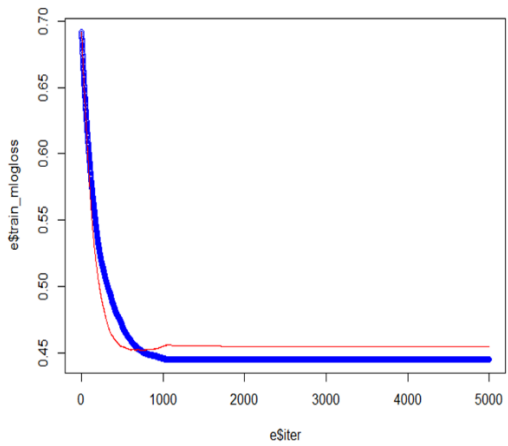


Figure (d)

Figure 5.14 A Plot of Good Fit Learning Curves

Figure 5.14 shows the plot of good-fit learning curves of the HESCA model. It can be observed that both the training loss (blue curve) and validation (red curve) loss gradually decrease to a point of stability (upon adding training examples) with a minimal gap called the generalisation gap between the two final loss curves. This suggests that addition of more training examples does not improve the (HESCA) model's performance on the training data (for training loss) and on the unseen data (for validation loss). Thus, the HESCA model achieves a good fit. Figure (a) is a learning curve with 1 000 trees indicating that addition of more training examples can improve the performance of the model both on training set and testing set. Figures (b), (c), and (d) are learning curves with maximum of 2 000, 3 000, and 5 000 trees respectively, indicating that addition of more training examples does not improve the performance of the model both on training set and testing set. Both losses on training set and testing set had attained the level of stability with minimal error.

5.7.1 Partial Dependence Plot and Individual Conditional Expectations

The Partial Dependence Plot (PDP) is similar to Individual Conditional Expectation (ICE), and shows the marginal effect a feature has on the predicted outcome (binary classification in this case) of a machine learning model (Friedman, 2001). A PDP or ICE can show whether the relationship between the target and a feature is linear, monotonic or more complex. The yellow curves indicate the PDP while the black curves represent the ICE. The PDP shows how the average prediction of all instances are associated with the feature while the ICE shows the prediction of each instance is associated with the feature. Having fitted the HESCA classification model, where GBM is a meta classifier; to predict the recurrent HNSCC prognosis, the partial dependence plot and ICE are used to visualise the relationships the model has learned. The influence of prognostic features on the predicted recurrent binary class is visualised in the Figures 5.15, 5.16, 5.17, 5.18, 5.19, and 5.20 below.

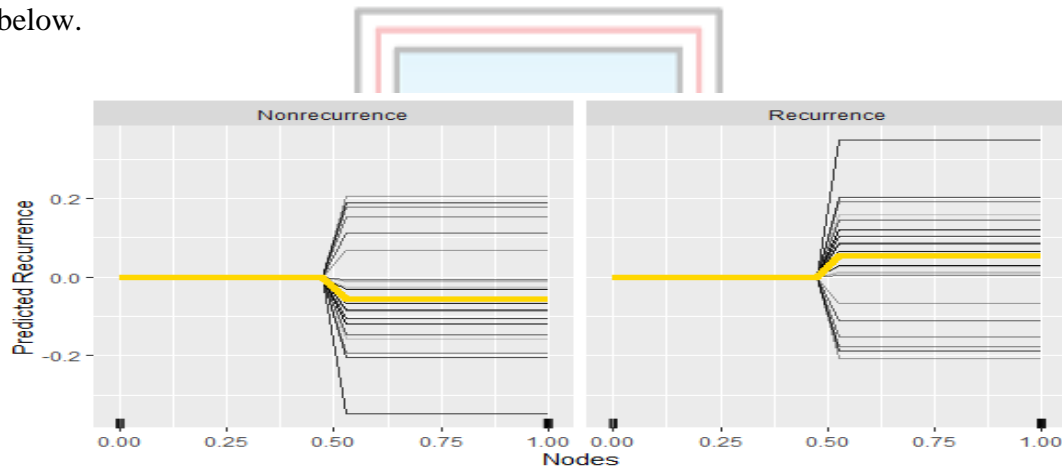


Figure 5.15 Individual Conditional Expectations on Feature Nodes

Figure 5.15 shows the PDP and ICE marginal effect of the feature Nodes on recurrent HNSCC prognosis. The PDP shows that, the recurrent HNSCC probability increases when the number of neck or cervical nodes exceeds 50% than when it is less than 50%. That is, the prediction of recurrent HNSCC is centered at “0” until the number of neck or cervical lymph nodes exceeds 50%. It can be observed that recurrent HNSCC probability increases around 50% of the presence of lymph nodes, but does this apply to every patient (instance) in the dataset? The ICE plot reveals that for most patients the cervical lymph nodes effect follows the average pattern of an increase at lymph nodes 50%, but there are some exceptions: For some patients that have a high predicted probability at a less presence of lymph nodes, the predicted recurrent HNSCC probability does not change with much presence of lymph nodes.

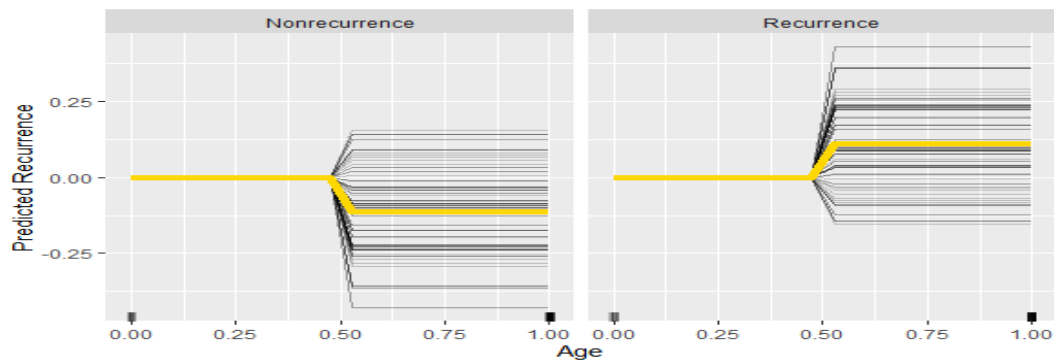


Figure 5.16 Individual Conditional Expectations on Feature Age

Figure 5.16 shows the PDP and ICE marginal effect of the feature Age on recurrent HNSCC prognosis. The PDP shows that, the recurrent HNSCC probability increases at around age of 55. That is, the prediction of recurrent HNSCC is centered at “0” until around the age of 55. It can be observed that recurrent HNSCC probability increases around the age of 55, but does this apply to every patient (instance) in the dataset? The ICE plot reveals that for most patients the age effect follows the average pattern of an increase at age 55, but there are some exceptions: For some patients that have a high predicted probability at a young age, the predicted recurrent HNSCC probability does not change with age.

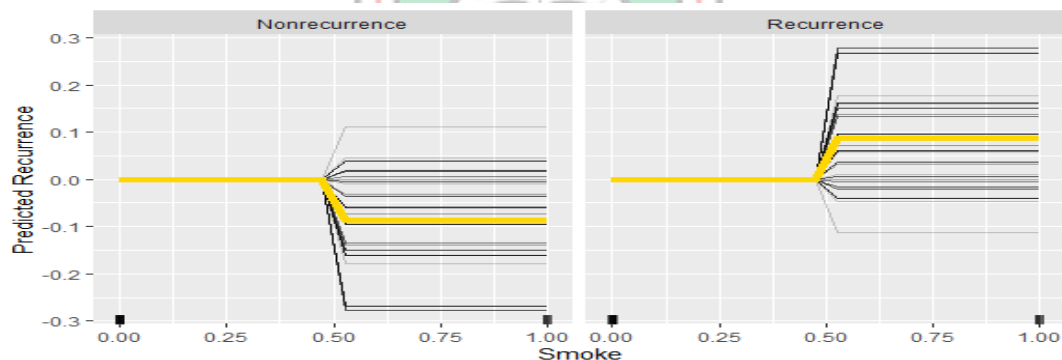


Figure 5.17 Individual Conditional Expectations on Feature Smoke

Figure 5.17 shows the PDP and ICE marginal effect of the feature Smoke on recurrent HNSCC prognosis. The PDP shows that, more recurrences are likely to occur as the rate of smoking habit of patients increases at around 52% and vice versa. The prediction of recurrent HNSCC is centered at “0” until the rate of habit of smoking goes up at around 52%. It can be observed that, the recurrent HNSCC probability increases at around high smoking habit rate of 52%, but does this apply to every instance in the dataset? The ICE plot reveals that for most patients the smoking effect follows the average pattern of an increase at the smoking habit rate of 52%, but there are some exceptions: For some patients

that have a high predicted probability at a low rate of smoking habit, the predicted recurrent HNSCC probability does not change with high rate of smoking habit.

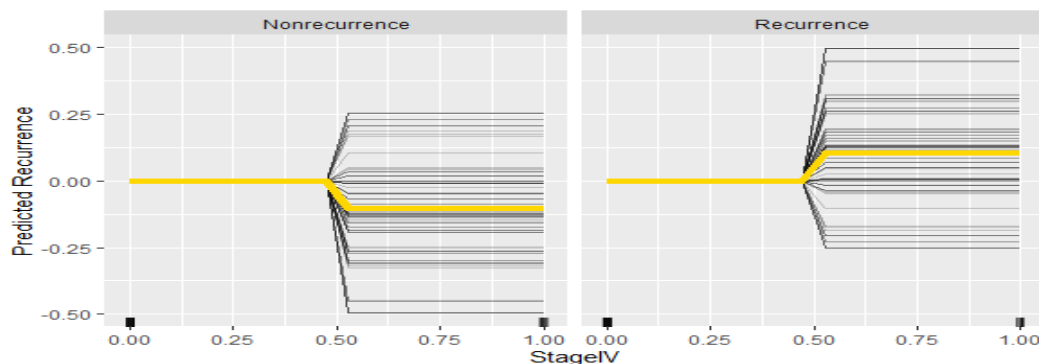


Figure 5.18 Individual Conditional Expectations on Feature StageIV

Figure 5.18 shows the PDP and ICE marginal effect of the feature Smoke on recurrent HNSCC prognosis. The PDP shows that, more recurrences are likely to occur as patients are diagnosis of HNSCC at the advanced or metastatic stage. The prediction of recurrent HNSCC is centered at “0” until the stage at diagnosis is beyond the stage II of the tumor at diagnosis. It can be observed that, the recurrent HNSCC probability increases at around stage II (around 55%) of metastasis, but does this apply to every instance in the dataset? The ICE plot reveals that for most patients the stage IV effect follows the average pattern of an increase at around 55% of metastasis, but there are some exceptions: For some patients that have a high predicted probability at a low stage (stage III), the predicted recurrent HNSCC probability does not change with stage IV.

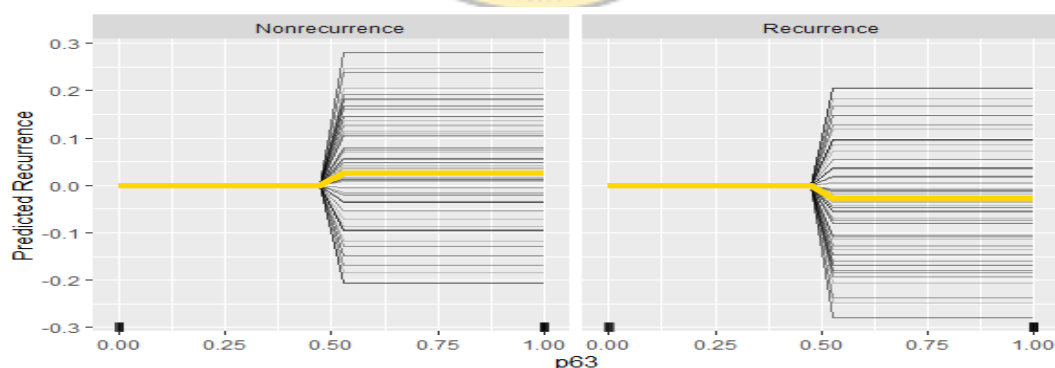


Figure 5.19 Individual Conditional Expectations on p63 Feature

Figure 5.19 shows the ICE effect of the feature p63 on recurrent HNSCC prognosis. It explains that, patient whose tumor suppressor gene p63 is either positively weak or strong is likely to experience recurrence holding all other factors or features constant.

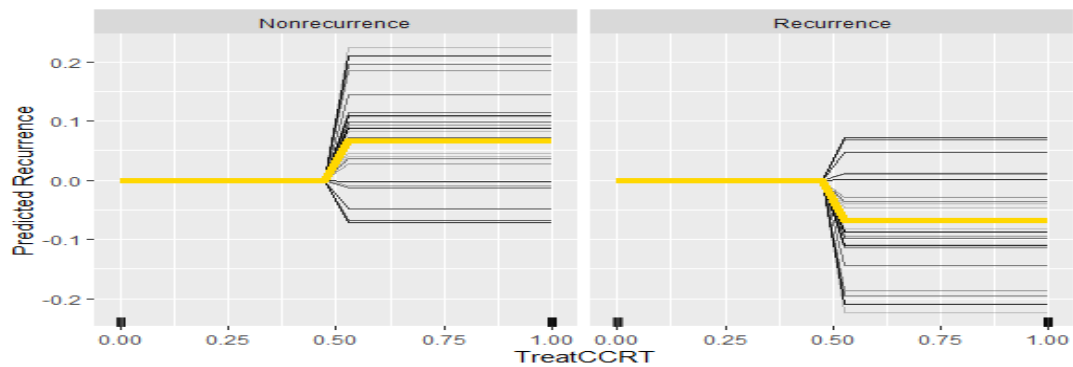


Figure 5.20 Individual Conditional Expectations on TreatCCRT Feature

Figure 5.20 shows the PDP and ICE effect of Treatment with Concurrent Chemoradiotherapy (TreatCCRT) feature on recurrent HNSCC prognosis; the binary classification (prognosis for HNSCC recurrence verses nonrecurrence). The more or higher the Concurrent Chemoradiotherapy (CCRT) treatment is administered to patients with HNSCC cases for curative intent, the less recurrences or relapses are experienced or recorded and vice versa. The PDP plot can be observed that the probability of recurrent HNSCC decreases around when the treatment process is half-way (55%) to its completion, but does this apply to every instance in the dataset? The ICE plot reveals that for most patients, the TreatCCRT effect follows the average pattern of decrease at around 55% to its completion, but there are some exceptions: For some patients that have low predicted probability at the half-way of treatment with CCRT, the predicted recurrent HNSCC probability does not change with TreatCCRT completion. This feature has a positive marginal effect on the target binary class.

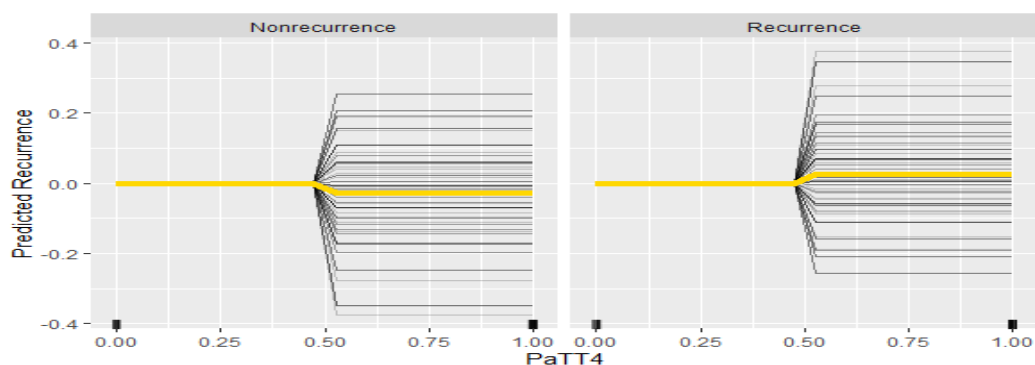


Figure 5.21 Individual Conditional Expectations on Feature PaTT4

Figure 5.21 shows the PDP and ICE marginal effect of the feature PaTT4 (pathological tumor staging at T4) on recurrent HNSCC prognosis. The PDP shows that, more recurrences are likely to occur as pathological tumor staging of patients' head and neck tumor is staged at either T3 or T4 at diagnosis. The prediction of recurrent HNSCC is centered at "0" until

the pathological tumor staging at diagnosis is beyond T2. It can be observed that, the recurrent HNSCC probability increases at around stage T2 (around 55%) of pathological tumor staging, but does this apply to every instance in the dataset? The ICE plot reveals that for most patients the PaTT4 effect follows the average pattern of an increase at around 55% of pathological tumor staging, but there are some exceptions: For some patients that have a high predicted probability at a low staging (T2), the predicted recurrent HNSCC probability does not change with PaTT4 (pathological tumor staging T4). This feature has a positive marginal effect on the target binary class.

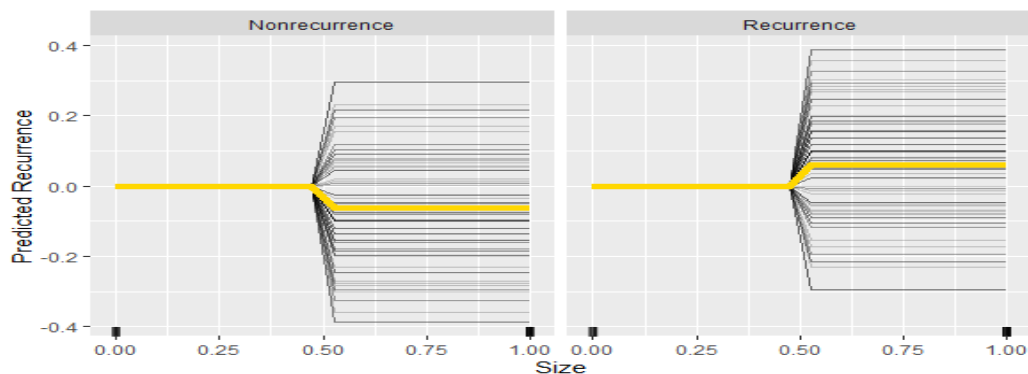


Figure 5.22 Individual Conditional Expectations on Feature Size

Figure 5.22 shows the PDP and ICE marginal effects of the feature “Size” on recurrent HNSCC prognosis. The ICE explains that, with low tumor size (that is, when the tumor size equals 2 cm or less), the possibility of experiencing recurrence is zero. But larger tumor size greater than 2 cm, the possibility of experiencing HNSCC recurrence is around 0.09. Interestingly, the predicted probability of experiencing recurrence does not fall when the size of tumor is greater than 2 cm. It can further be observed from the PDP that, the recurrent HNSCC probability increases at around 0.55 (around 2 cm) of the tumor size, but does this apply to every instance in the dataset? The ICE plot reveals that for most patients the tumor size effect follows the average pattern of an increase at around 2 cm, but there are some exceptions: For some patients that have a high predicted probability at a smaller tumor size (2 cm), the predicted recurrent HNSCC probability does not change with size (tumor size 2 cm or greater). This feature has a positive marginal effect on the target binary class.

In summary, the present study focuses on improving the classification performance on the face of accuracy, log loss, recall, precision, specificity, F1-Score, and AUC based on HNSCC prognosis dataset for recurrence. To do this, a hybrid stacked ensemble technique that identifies a robust meta-classifier model when the classifiers employed as base

classifiers are as well as employed as meta-classifiers. Here, GBM, DRF, DNN, GLM, and NB were first used as base classifiers. To find the optimal meta-classifier model consisting of the combination of these base classifiers used in this study, each base classifier was used as the meta-classifier. Meanwhile, the overall aim of the study is to formulate a hybrid ensemble super classification algorithm (HESCA) model that can be applied in recurrent HNSCC for accurate prognosis based on clinical, pathological, and genomic markers. As a result, a hybrid prognostic classification model for recurrent HNSCC using ML techniques based on optimum features has been developed with very promising results. The findings for developing the recurrent HNSCC prognostic classification model are summarised as according to the research objectives.

Chapter three (3) explains the overall methods used in this study. The study identified the most accurate prognosis for recurrent HNSCC using the GBM for ensemble FS. Chapter 4 discusses the details of this technique and Chapter 5 discusses the results and discussions. The study developed a hybrid stacked ensemble prognostic classification model for the recurrent HNSCC using stacked ensemble technique that proves this classification model is well optimally and can serve as a supportive tool for the prognosis HNSCC recurrence. For this, a stacked ensemble technique was proposed to cater for the need to improving the generalisation ability of a classification model. This is described in Chapter 4 in terms of the methods, preparations, and procedures adopted to acquire the prognosis for HNSCC data described in Chapter 3. The study also validated the developed HESCA model with the existing data and compare the performance results with two baseline stacked ensemble classification models and one SA classification model using the existing data used for the study. This is discussed in Chapter 5 where the baseline stacked ensemble techniques and the state-of-the-art technique are also applied to learn the recurrent HNSCC dataset as done for the proposed the HESCA model. Further, the study proved that the prognosis of recurrent HNSCC is more robust when gradient boosted features are used. This is discussed in Chapter 5 where various feature selection techniques considered in this study were applied to the original training data; and, where the optimality of each feature selector being verified using the HESCA model.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

This concluding chapter discusses further about the research work. The conclusions, research contributions to knowledge, recommendations, and limitation of the study are discussed in subsequent sections.

6.1 Conclusion

The proposed hybrid HESCA model is a stacked ensemble-based technique. The pre-processing techniques were implemented on recurrent HNSCC prognostic dataset consisting the markers of clinicopathological and that of genomic. From there, feature selection techniques were implemented with the aims to reduce the number of training features; to avoid overfitting, and to find out the optimal feature subset for recurrent HNSCC prognostic model. Five feature selection techniques were implemented which are GBM, DRF, DNN, GLM, and NB. The proposed hybrid HESCA model was implemented on the feature subsets selected from each FS technique. Then, the proposed hybrid HESCA model was developed on the 8-input optimal features that the GBM-FS as an ensemble FS technique generated. Thus, the proposed HESCA model was developed for the classification of whether patients had recurrence or nonrecurrence after subsequent years of diagnosis, treatment, and follow-up; here, one-year to five-year. Due to the small number of training instances, in order to learn the proposed stacked ensemble classification model, a ν -fold ($\nu=10$) CV was implemented. The generated results from the proposed HESCA model were compared with individual base ML models (GBM, DRF, DNN, GLM, and NB). Furthermore, the 8-input HESCA model was validated and tested using the existing test data and the results were compared with two baseline stacked ensemble classification models, and state-of-the-art classification model used in the study. These compared existing stacked ensemble models were learned and tested on the same existing training and test data respectively. Also, the 8-input HESCA model was compared with the HESCA model with full-input features based on training set and test set. Findings obtained from the analysis of the recurrent HNSCC prognostic dataset for the HESCA model are;

- i. The optimal feature subset for the recurrent HNSCC prognosis is a composition of *Nodes*, *Age*, *Smoke*, *StageIV*, *p63*, *TreatCCRT*, *PaTT4*, and *Size* being the most

- accurate prognosis which is in accordance with the findings in the literature of similar previous studies.
- ii. The performance of the GBM as a meta-classifier is better than other classifiers used as meta classifiers in a stacking ensemble having five base classifiers.
 - iii. A stacked ensemble classification (HESCA) model having five base classifier models achieves the best performance accuracy in recurrent HNSCC prognosis compare to the individual ML classification models and other stacked ensemble classification models considered in the study.
 - iv. The prognostic result is more accurate with the GBM-FS ensemble feature selection technique compared to the prognostic result with other base feature selection techniques considered in the study.
 - v. The HESCA prognostic model with 8-input features based on the GBM-FS optimal features performs well optimally than the HESCA model with the full-input features in a 5-year recurrent HNSCC prognostic data.

Surely, the 8-input HESCA model with the training features being; *Nodes*, *Age*, *Smoke*, *StageIV*, *p63*, *TreatCCRT*, *PaTT4*, and *Size*, achieved better performance accuracy that can be considered feasible and be adopted as a supporting tool by clinicians as the prognosis for recurrent HNSCC subtypes. The gender, alcoholism, chewing quid, tumor site, tumor grade, invasion font, pathological lymph nodes, family history, HPV, and *p16* features had no significant effect on the class label. In summary, patient with much lymph nodes, old age, high rate of smoking habit, negative *p63*, the proposed HESCA prognostic model based on the GBM-FS feature selection with 10-fold cross-validation in a stacked ensemble provides a machine learning-based approach to recurrent HNSCC prognosis based on a combination of only three medical information: clinical, pathological, and genomic markers.

6.2 Research Contributions to Knowledge

The contribution of this study to the development of science can be structured in four parts.

- i. First, the most precise and reliable features as the prognosis for recurrent HNSCC have been developed by an ensemble feature selection technique based on GBM-FS. Using the HESCA model, a comparison analysis with the other four feature selection techniques that were taken into consideration in the study was done to determine how robust this feature selection methodology is. Data from multiple feature

selection techniques was fed into the HESCA model, which was then used to train different feature subsets. The GBM-FS technique was found to be adaptable and to have the greatest outcomes across all training and test performance parameters, providing the most accurate prognosis for the recurrent HNSCC subtypes, according to the results analysis. The GBM-FS technique also has a propensity to reduce feature redundancy and dimensionality, enabling effective consideration of pertinent features for improving classification accuracy and training efficiency. The versatility of the GBM-FS gives it a further benefit in determining the most precise prognosis for data on recurrent HNSCC. With the right combination and size of feature subsets, the GBM-FS increased classification accuracy while lowering computing costs, overfitting, training time, and modeling uncertainty. But it was noted that the GBM-FS method was task-specific. The feature selections under comparison have fairly comparable prognoses. As a result, the most reliable prognostic indicators for recurrent HNSCC subtypes were found to be the GBM-FS characteristics *Nodes*, *Age*, *Smoke*, *Stage IV*, *p63*, *TreatCCRT*, *PaTT4*, and *Size*.

- ii. The HESCA model, a novel hybrid stacked ensemble technique, has been established as the second novelty for learning a classification of the prognosis for recurrent HNSCC patterns. The suggested hybrid ensemble super classification algorithm (HESCA) model's generalisability was evaluated using the good-fit learning curves. It was discovered that the validation loss (red curve) and training loss (blue curve) both steadily decreased to a point of stability (upon adding training instances), with a narrow gap between the two final loss curves known as the generalisation gap. This implies that increasing the number of training instances does not enhance the (HESCA) model's performance when applied to training data (for training loss) and test data (for validation loss). The HESCA model has good fit, which led to good generalisation. The suggested HESCA model successfully classified all recurrent HNSCC data, demonstrating its generalisability and flexibility. Instead of devoting resources to figuring out which stacked ensemble classifier to choose for a given task since the impact of the base classifiers varied with different datasets, the proposed classification technique can be adopted for medical applications due to its generalisability and adaptability.
- iii. Third, three pre-existing stacked ensemble models have been used to validate and contrast the HESCA model that has been constructed. Using the current HNSCC data, a comparison study with two baseline stacked ensemble classification models

and one state-of-the-art stacked ensemble classification model was carried out in order to determine the superiority of the proposed hybrid ensemble super classification algorithm (HESCA) model. In terms of training and test accuracy as well as other assessment criteria, it was found that the suggested HESCA model excelled all compared adaptations. It was found that the stacked ensemble model with the base classifier functioning as the meta-classifier produced superior classification accuracy than the two baseline stacked ensemble models proposed by Kabir and Ludwig (2019), where no base classifier functioned as the meta-classifier. Furthermore, it was found that the state-of-the-art stacked ensemble model with GBM meta-classifier proposed by Kwon *et al.* (2019) with a maximum of four base classifier models performed worse in terms of classification accuracy than the proposed HESCA model, which is a stacked ensemble model consisting of five base classifier models with GBM meta-classifier.

- iv. Fourthly, the most reliable predictor for recurrent HNSCC has been determined to be the gradient boosted features. The HESCA model was used in a comparison analysis using the four feature subsets taken into consideration in the study to determine the robustness of these features. The data generated by various feature selection techniques were utilised to train the HESCA model on various feature subsets. Gradient boosted features were shown to be adaptable and to have the greatest outcomes across all training and test performance parameters, providing the most accurate prognosis for the recurrent HNSCC subtypes, according to the findings analysis.

6.3 Recommendation

This study recommends that the developed hybrid prognostic model could be used as a supporting tool for classifying the prognosis for recurrent HNSCC patterns in clinical domain. Meanwhile, other tests with large dataset may be required for further verification of the outputs generated in the study. In as much as the number of the training instances is small, optimistically, the study serves as a groundwork to embark on more similar research in Ghana. However, the proposed technique still has more room for improvement, which suggests that the future work should;

- i. consider other ML classifiers like SVM, DT etc that might improve the prediction accuracy, which were not considered in this study due to limited computational resources.
- ii. Employ heterogeneous feature selection techniques for ensemble feature selection with the investigation on more diverse base classifiers and meta-classifiers for stacked generalisation.
- iii. Policy makers of cancer awareness should educate Ghanaians on HNC awareness to frequently and should visit various healthcare facilities frequently for medical checkup.
- iv. Finally, apply this technique to other real-world cancer problems, and other problem domains such as cyber security, agriculture, geographic information system, and transportation.

6.4 Limitation of the Study

Ghana's research in medical informatics is still in its infancy, hence there are not many medical databases to choose from. Due to the lack of sufficient medical samples or tissues for the study experiments, this has restricted the research activity in this area. However, it takes time to prepare the medical tissues for the goal of obtaining genomic data. Moreover, getting the genomic data came at a considerable cost. In addition, the majority of medical records now in existence are preserved in hardcopy (paper format), so converting these data into a computerised format took some time. Depending on the quantity of samples, the entire process took several months. The study only had data up to a maximum 5-year prognosis. This is because the records for instances older than five years are incomplete. For records older than five years, an additional year or two are required to gather enough data for the prognostic model that has been proposed. Hence, only 1-year to 5-year recurrence were included in the study due to time and financial constraints, and only the genetic data for *p16* and *p63* were chosen.

REFERENCES

- Abdul-Kareem, S., Baba, S., Zubairi, Y. Z., Prasad, U. and Wahid, M. I. A. (2001), “Prognostic Systems for NPC: A Comparison of the Multilayer Perceptron Model and the Recurrent Model”, Paper presented at the 9th International Conference on Neural Information Processing.
- Acuna, E. and Rodriguez, C. (2004), “The treatment of missing values and its effect in the classifier accuracy”, *Classification, Clustering and Data Mining Applications*, pp. 639-648.
- Adeyemi, O. J., Adebayo, V. O., Olaniyi, O., Olusanya, O. O. and Idowu, P. A. (2019), “A Stack Ensemble Model for the Risk of Breast Cancer Recurrence”, *International Journal of Research Studies in Computer Science and Engineering*, Vol. 6, No. 3, pp. 8-21.
- Akinbohun, F. (2021), “A Stacked Ensemble Model for Diagnosis of Craniocervical (Head and Neck) Cancer”, Post Graduate Research Unit of Albert Ilemobade Library, FUTA, pp. 1.
- Akinbohun, F., Akinbohun, A., Daniel, A. and Oyinloye, O. E. (2020), “Diagnosis of Head and Neck Cancer in Developing Countries using a Stacked Ensemble Model”, *European Journal of Engineering Research and Science*, Vol. 5, No. 9, pp. 1-5.
- Alabi, R. O., Elmusrati, M., Sawazaki-Colone, I., Kowalski, L. P., Haglund, C. and Coletta, R. D. *et al.* (2019), “Machine learning application for prediction of locoregional recurrences in early oral tongue cancer”, *A Web-based prognostic tool*, VirchowsArchiv.
- Anon. (2020), “Head and neck squamous cell carcinoma”, NIH, <https://ghr.nlm.nih.gov/condition/head-and-neck-squamous-cell-carcino...> 2020
- Anon, (2019), “The Abramson Cancer Center of the University of Pennsylvania”, <https://www.oncolink.org/about-oncolink/use-of-the-oncolink-site- and-content>

- Anon. (2017), “Head and Neck Cancers”, National Cancer Institute.
<https://www.cancer.gov/types/head-and-neck/head-neck-fact-sheet>
- Anon. (2016), “Head and Neck Cancers: Follow-up Care After Cancer Treatment”,
<https://www.cancer.gov/aboutcancer/coping/survivorship/followupcare/follow-up-fact-sheet>, National Cancer Institute.
- Anon. (2015), “Head and neck squamous cell carcinoma”, NIH,
<https://pubmed.ncbi.nlm.nih.gov/33243986/>
- Argiris, A., Karamouzis, M. V., Raben, D. and Ferris, R. L. (2008), “Head and neck cancer. *Lancet*”, Vol. 371, No. 9625, pp. 1695–1709.
- Baiju, N., T. (2019), “How to teach your kids about Machine Learning and AI”,
<https://roboticsbiz.com/author/baiju/page/7/>
- Beale, M. H. and Hagan, M. T. (2012), “Neural network toolbox™ user’s guide”, Natick, MA: *The MathWorks, Inc.*
- Bishop, C. M. (2006), “Pattern recognition and machine learning”, New York: Springer.
- Boffetta, P., Hecht, S., Gray, N. and Gupta, S. K. (2008), “Smokeless tobacco and cancer”, *The Lancet Oncology*, Vol. 9, No. 7, pp. 667-675.
- Boyle, P. and Levin, B. (2008), “World Cancer Report”, International Agency for Research on Cancer.
- Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R. L., Torre, L. A. and Jemal, A. (2018), “Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries”, *CA: A Cancer Journal for Clinicians*, No. 6, Vol. 68, pp. 394.
- Breiman, L. (2001), “Random forests: Machine Learning”, Vol. 45, No. 1, pp. 5–32.
- Brockstein, B., Haraf, D. J., Rademaker, A. W., Kies, M. S., Stenson, K. M., Rosen, F., Mittal, B. B., Pelzer, H., Fung, B. B., Witt, M-E., Wenig, B., Portugal, L.,

- Weichselbaum, R. W. and Vokes, E. E. (2004), “Patterns of failure, prognostic factors and survival in locoregionally advanced head and neck cancer treated with concomitant chemoradiotherapy: a 9 year, 337-patient, multi-institutional experience”, *Annals of oncology*, Vol.15, pp.1179–1186.
- Brown, K. F., Rungay, H., Dunlop, C. *et al.* (2018), “The fraction of cancer attributable to known risk factors in England, Wales, Scotland, Northern Ireland, and the UK overall in 2015”, *British Journal of Cancer* 2018.
- Boulesteix, A. L., Janitza, S., Kruppa, J. and König, I. R. (2012), “Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics”, *Wiley Interdiscip Rev Data Min Knowl Discov*, Vol. 2, No. 6, pp. 493–507.
- Cai, H., Pang, X., Dong, D., Ma, Y., Huang, Y., Fan, X., Wu, P., Chen, H., He, F., Cheng, Y., Liu, S., Yu, Y. Hong, M., Xiao, J., Wan, X., Lv, Y. and Zheng, J. (2019), “Molecular Decision Tree Algorithms Predict Individual Recurrence Pattern for Locally Advanced Nasopharyngeal Carcinoma”, *Journal Cancer*, No. 15, Vol. 10, pp. 3323-3332. <http://www.jcancer.org/v10p3323.htm>
- Catto, J. W. F., Abbod, M. F., Linkens, D. A. and Hamdy, F. C. (2006), “Neuro-Fuzzy Modeling: An Accurate and Interpretable Method for Predicting Bladder Cancer Progression”, *The Journal of Urology*, Vol. 175, pp. 474- 479.
- Chi-Chang, C., Huang, T-H., Shueng, P-W., Chen, S-H., Chen, C-C., Lu, C-J. and Tseng Y-J. (2021), “Developing a Stacked Ensemble-Based Classification Scheme to Predict Second Primary Cancers in Head and Neck Cancer survivors”, *International Journal of Environment Research and Public Health*, Vol. 18, No. 12499, pp. 1-10.
- Chang S-W., Abdul-Kareem S., Merican A. F. and Zain R. B. (2013), “Oral cancer prognosis based on clinicopathologic and genomic markers using a hybrid of feature selection and machine learning methods”, *BMC Bioinforma*, No. 14, pp. 170.

- Chattopadhyay, R. and Guha, A. (2004), “Artificial neural networks: applications to textiles”, *Textile Progress*. 35, 1, 1–42.
- Chen, P.-H., Shieh, T.-Y., Ho, P. S., Tsai, C.-C., Yang, Y.-H. and Lin, Y.-C. (2007), “Prognostic Factors Associated with the Survival of Oral and Pharyngeal Carcinoma in Taiwan”, *BMC Cancer*, Vol. 7, No. 101.
- Clark, T., Bradburn, M. and Love, S. A. D. G. (2003), “Survival Analysis Part 1: Basic Concepts and First Analyses”, *British Journal of Cancer*, Vol. 89, pp. 232-238.
- Cochran, W. G. (2007), “Sampling Techniques”, (3rd ed.), New York: John Wiley and Sons.
- Colozza, M., Cardoso, F. and Sotiriou, C. (2005), “Bringing molecular prognosis and prediction to the clinic”, *Clin Breast Cancer*, Vol. 6, pp. 61-76.
- Comme, E. (2019), “Cancer Cases in Ghana are not decreasing”, General News, Source: GNA.
- Cruz, J. A. and Wishart, D. S. (2006), “Review: Applications of Machine Learning in Cancer Prediction and Prognosis”, *Cancer Informatics*, No. 2, pp. 59-74.
- Dietterich, T. (2001), “Ensemble methods in machine learning”, In: *Proceedings of the 1st International Workshop on Multiple Classifier Systems*, pp. 1–15.
- Dom, R. M., Abdul-Kareem, S., Abidin, B., Jallaludin, R. L. R., Cheong, S. C. and Zain, R. B. (2008), “Oral Cancer Prediction Model for Malaysian Sample”, *Austral Asian Journal of Cancer*, Vol. 7, No. 4, pp. 209-214.
- Du, D., Feng, H., Lv, W., Ashrafinia, S., Yuan, Q., Wang, Q., Yang, W., Feng, Q., Chen, W., Rahmim, A. and Lu, L. (2019), “Machine Learning Methods for Optimal Radiomics-Based Differentiation”, *jnl of Molecular Imaging & Biology*.
- Duda, R. O., Hart, P. E. and Stork, D. G. (2001), “Pattern classification (2nd edition)”, New York: Wiley.

- Dybowski, J. N., Heider, D. and Hoffmann, D. (2010), “Structure of hiv-1 quasi- species as early indicator for switches of co-receptor tropism”, *AIDS Res Ther.*, Vol. 7, pp. 41.
- Exarchos, K. P., Goletsis, Y. and Fotiadis, D. I. (2012b), “A multiscale and multiparametric approach for modeling the progression of oral cancer”, *BMC Medical Informatics and Decision Making*, pp. 12-136.
- Exarchos, K. P., Goletsis, Y. and Fotiadis, D. I. (2011a), “Multiparametric decision support system for the prediction of oral cancer reoccurrence”, *IEEE Trans InfTechnol Biomed*, No. 16, pp. 1127–1134.
- Fielding, L. P., Fenoglio-Preiser, C. M. and Freedman, L. S. (1992), “The Future of Prognostic Factors in Outcome Prediction for Patients with Cancer”, *Cancer*, Vol. 70, pp. 2367-2377.
- Foundation, O. C. (2010), “Oral Cancer Facts”, Available from: <http://www.oralcancerfoundation.org/facts/index.htm>.
- Friedman, J., H. (2001), “Greedy function approximation: A gradient boosting machine.” *Annals of statistics*, pp. 1189-1232.
- Gandini, S., Botteri, E., Iodice, S., Boniol, M., Lowenfels, A. B., Maisonneuve, P. and Boyle, P. (2008), “Tobacco smoking and cancer: a meta-analysis. *International Journal of Cancer*”, Vol. 122, No. 1, pp. 155-164.
- Gangil, T., Shahabuddin, A. B., Rao, D. B., Palanisamy, K., Chakrabarti, B. and Sharan, K. (2022), “Predicting clinical outcomes of radiotherapy for head and neck squamous cell carcinoma patients using machine learning algorithms”, *Journal of Big Data*, pp. 1-19.
- Graupe, D. (2007), “Principles of artificial neural networks”, Singapore: World Scientific Publishing Advanced Series on Circuits and Systems. 2nd edition, Vol. 6. Co. Pte. Ltd.

- Gremmell, D. (2018), “Ensemble Learning in R with SuperLearner”, <https://www.datacamp.com/community/tutorials/ensemble-r-machine-learning>.
- Hagerty, R. G., Butow, P. N., Ellis, P. M. (2005), “Communicating prognosis in cancer care: a systematic review of the literature”, *Ann Oncol*, No. 16, pp. 1005-1053.
- Hand, D. J. and Yu, K. (2001), "Idiot's Bayes-not so stupid after all?", *International Statistical Review*, Vol. 69, No. 3, pp. 385–399.
- Hashibe, M., Brennan, P., Chuang, S. C. Boccia, S., Castellsague, X., Curado, M. P., Maso, L. D. and Daudt, A. W. (2009), “Interaction between tobacco and alcohol use and the risk of head and neck cancer: pooled analysis in the international Head and Neck Cancer Epidemiology Consortium”, *Journal of National Cancer Institute*, Vol. 18, No. 2, pp. 541-550.
- Hashibe, M., Brennan, P., Benhamou, S., Castellsague, X. and Chen, C. (2007), “Alcohol drinking in never use of tobacco, cigarette smoking in never drinkers, and the risk of head and neck cancer: pooled analysis in the international Head and Neck Cancer Epidemiology Consortium”, *Journal of National Cancer Institute*, Vol. 99, No. 10, pp. 777-789.
- Hastie, T., Tibshirani, R. and Friedman, J. (2009), “The elements of statistical learning”, In Vol. 1 of Bayesian forecasting and dynamic models, New York: Springer, pp. 1–694.
- He, Z. and Yu, W. (2010), “Stable feature selection for biomarker discovery”, *Comput Biol Chem*, Vol. 34, pp. 215–25.
- Jemal, A., Siegel, R., Ward, E., Murray, T., Xu, J. and Thun, M. J. (2007), “Cancer statistics”, *CA Cancer Journal for Clin.*, Vol. 57, No. 1, pp. 43-66.
- Kabir, F. and Ludwig, S. A. (2019), “Enhancing the Performance of Classification Using Super Learning”, *Data-Enabled Discov. Appl.*, Vol. 3, No. 5, pp. 2-13.
- Kan, T., Shimada, Y. and Sato, F. Tetsuo Ito, Kondo, K., Watanabe, G., Maeda, M., Yamasaki, S., Meltzer, S. J. and Imamura, M. (2004), “Prediction of lymph node

metastasis with use of artificial neural networks based on gene expression profiles in esophageal squamous cell carcinoma”, *Ann Surg Oncol*, No. 11, pp. 1070-1078.

Karparthy, A. (2016), “Neural Networks 1”, <http://cs231n.github.io/neural-networks-1/>, Online.

Kennedy, B. K., Berger, S. L., Brunet, A., Campisi, J., Cuervo, A. M., Epel, E. S., Franceschi, C., Lithgow, G. J., Morimoto, R. I., Pessin, J. E., Rando, T. A., Richardson, A., Schadt, E. E., Wyss-Coray, T., Sierra, F. (2014), “Geroscience: linking aging to chronic disease *Cell*”, Vol. 159, No. 4, pp. 709-13.

Kourou, K., Exarchosa, T. P., Exarchos, K. P., Karamouzis, M. V. and Fotiadis, D. I. (2015), “Review: Machine learning applications in cancer prognosis and prediction”, *Computational and Structural Biotechnology Journal*, No. 13, pp.8-17.

Kwon, H., Park, J. and Lee, Y. (2019), “Stacking Ensemble Technique for Classifying Breast Cancer”, *Jnl of Health Informatics Research*, Vol. 25, No. 4, pp. 283-288.

Lavanya, L. and Chandra, J. (2019), “Oral Cancer Analysis Using Machine Learning Techniques”, *International Journal of Engineering Research and Technology*, Vol. 12, No. 5, pp. 596-601.

LeDell, E. (2016), “Scalable super learning”, *Handbook of Big Data*, PP. 339.

Llorca, J. and Delgado-Rodríguez, M. (2002), “Visualising exposure-disease association: the lorenz curve and the gini index”, *Med Sci Monit*. Vol. 8, No. 10, pp. 193–7.

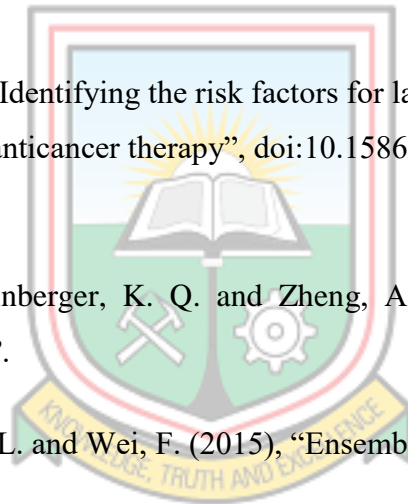
Li, S., Wang, K., Hou, Z., Yang, J., Ren, W., Gao, S., Meng, F., Wu, P. Liu, B. Liu, J. and Yan, J. (2018), “Use of Radiomics Combined with Machine Learning Method in the Recurrence Patterns After Intensity-Modulated Radiotherapy for Nasopharyngeal Carcinoma: A Preliminary Study”, *Journal Frontiers in Oncology*,

- Liu, J., Wyatt, J. C. and Altman, D. G. (2013), “Decision tools in health care: focus on the problem, not the solution”, *BMC Medical Informatics and Decision Making*, Vol. 6, No. 4.
- Megha, S. (2020), “Binary Cross Entropy aka Log Loss-The cost function used in Logistic Regression,<https://www.analyticsvidhya.com/blog/2020/11/binary-cross-entropy-aka-log-loss-the-cost-function-used-in-logistic-regression/>”
- Mehrotra, R. and Yadav, S. (2006), “Oral Squamous cell carcinoma: Etiology, pathogenesis and prognostic value of genomic alterations”, *Indian Journal of Cancer*, Vol. 43, No. 2, pp. 60-66.
- Mitchell, T. M. (2006), “The discipline of machine learning”, Carnegie Mellon University, School of Computer Science, Machine Learning Department.
- Mitchell, T. M. (1997), “Machine Learning”, New York: McGraw Hill.
- Mucke, T. S., Wagenpfeil, S., Kesting, M. R., Hölzle, F., Wolff, K-D. (2009), "Recurrence interval affects survival after local relapse of oral cancer," *Oral Oncol*, Vol. 45, pp. 687- 691.
- Munakata, T. (2008), “Fundamentals of the new artificial intelligence; neural, evolutionary, fuzzy and more”, 2nd edition, London: Springer-Verlag London Limited.
- Narasimha M. M. and Susheela, D. V. (2011), “Pattern Recognition: An Algorithmic Approach”, ISBN 978-0857294944.
- Neumann, U., Genze, N. and Heider, D. (2017), “EFS: an ensemble feature selection tool implemented as R-package and web-application”, *Journal of Bio Data Mining*, pp. 1-9.
- Okut, H., Wu, X. L., Rosa, J. M. G., Bauck, S., Woodward, B., Schnabel, D. R., Taylor, F. J. and Gianola, D. (2014), “Predicting expected progeny difference for marbling score in Angus cattle using artificial neural networks and Bayesian regression models”, *Genetics Selection Evolution*. Vol. 45, pp.34.

- Oliveira, L. R., Ribeiro-Silve, A., Costa, J. P. O., Simoes, A. L., Di Matteo, M. A. S. and Zucoloto, S. (2008), “Prognostic factors and survival analysis in a sample of oral squamous cell carcinoma patients”, *Oral Surgery, Oral Medicine, Oral Pathology, Oral Radiology, and Endodontology*, Vol. 106, No. 5, pp. 685-695.
- Owusu-Afriyie, O., Owiredu, W., Owusu-Danquah, K., Komarck, C., Foltin, S. K. and Larsen-Reindorf, R. (2020), “Expression of immunohistochemical markers in non oropharyngeal head and neck squamous cell carcinoma in Ghana”, *PLoS ONE*. Vol. 13, No. 8, pp. e0202790.
- Piryonesi, S. M. and El-Diraby, T. E. (2020), “Role of Data Analytics in Infrastructure Asset Management: Overcoming Data Size and Quality Problems”, *Journal of Transportation Engineering*, PP. 1-17.
- Pyka, M., Hahn, T., Heider, D., Krug, A., Sommer, J., Kircher, T. and Jansen, A. (2013), “Baseline activity predicts working memory load of preceding task condition”, *Hum Brain Mapp*, Vol. 34, No. 11, pp. 3010–22.
- Ragunthar, T. and Selvakumar, S. (2019), “Classification of gene expression data with optimized feature selection”, *Int. J. Recent Technol. Eng*, Vol. 8, No. 2, pp. 4763-4769. <https://doi.org/10.35940/ijrte.B1845.078219> (2019).
- Razak, A. A., Saddki, N., Naing, N. N. and Abdullah, N. (2010), “Oral cancer survival among Malay patients in Hospital Universiti Sains Malaysia, Kelantan”, *Asian Pacific Journal of Cancer Prevention*, Vol. 11, No. 2, pp. 187-191.
- Reichart, P. A. (2001), “Identification of Risk Groups for Oral Precancer and Cancer and Preventive Measures”, *Clin. Oral Invest*, Vol. 5, pp. 207-213.
- Ribeiro, I. P., Caramelo, F., Esteves, L., Menoita, J., Marques, F., Barroso, L. Migueis, J., Melo, J. B., Carreira, I. M., (2017), “Genomic predictive model for recurrence and metastasis development in head and neck squamous cell carcinoma patients”, *Scientific Report*, pp. 1-8.
- Rokach, L. (2010), “Pattern Classification Using Ensemble Methods”, *World Scientific Publishing Company*, Singapore.

- Rosado, P., Lequerica-Fernández, P., Villallaín, L., Peña, I., Sanchez-Lasheras, F. and de Vicente, J. C. (2013), “Survival model in oral squamous cell carcinoma based on clinicopathological parameters, molecular markers and support vector machines”, *Expert SystAppl*, Vol. 40, pp. 4770–4776.
- Saeys, Y., Inza, I. and Larrañaga, P. (2007), “A review of feature selection techniques in bioinformatics”, *Bioinformatics*, Vol. 23, No. 19, pp. 2507– 2517.
- Sandri, M. and Zuccolotto, P. (2008), “A bias correction algorithm for the gini variable importance measure in classification trees”, *J Comput Graph Stat*, Vol. 17, No. 3, pp. 611–28.
- Siegel, R. L., Miller K. D. and Jemal, A. (2017), “Cancer Statistics”, CA: *Cancer Journal for Clinicians*, Vol. 67, No. 1, pp. 7-30.
- Singh, S., Yassine, A. and Benlamri, R. (2020), “Internet of Energy: Ensemble Learning through Multilevel Stacking for Load Forecasting”, *IEEE Intl Conf on Dependable*, pp. 664-668.
- Singh, A., Goyal, S., Rao, Y. J. and Loew, M. (2019), “Tumor Heterogeneity and Genomics to Predict Radiation Therapy Outcome for Head-and-Neck Cancer: A Machine Learning Approach”, *Radiation Oncology*, Vol. 105, No. 1, pp. 232-233.
- Stewart, B. and Wild, C. P. (2016), “World cancer report 2014”, World 2016.
- Su, J., Zhang, Y., Su, H., Zang, C. and Li, W. (2017), “A recurrence model for laryngeal cancer based on SVM and gene function clustering”, *Acta Otolaryngol*, Vol. 137, No. 5, pp. 557-562.
- Tang, Z., Wei, G., Zhang, L. and Xu, Z. (2019), “Signature microRNAs and long noncoding RNAs in laryngeal cancer recurrence identified using a competing endogenous RNA network”, *Molecular Medicine Reports, Spandidos publications*, pp. 4806-4818.

- Upadhyay, D., Manero, J., Zaman, M. and Sampalli, S. (2021), “Gradient Boosting Feature Selection with Machine Learning Classifiers for Intrusion Detection on Power Grids”, *IEEE*, Vol. 18, No. 1, pp. 1104-1115.
- Wang, J., Xu, J., Zhao, C., Peng, Y. and Wang, H. (2019), “An ensemble feature selection method for high-dimensional data based on sort aggregation”, *Systems Science & Control Engineering*, No. 7, Vol. 2, 32-39.
- Warsinske, H., Vashisht, R. and Khatri, P. (2019), “Host-response-based gene signatures for tuberculosis diagnosis: A systematic comparison of 16 signatures”, *PLoS Med* Vol. 16, No. 4, pp. 1–19.
- Wolpert, D. H. (1992), “Stacked generalization”, *Neural Networks*, Vol. 5, No. 2, pp. 241-259.
- Worsham, M. J. (2011), “Identifying the risk factors for late-stage head and neck cancer: Expert review of anticancer therapy”, doi:10.1586/era.11.135, No. 11, pp. 1321-1325.
- Xu, Z., Huang, G., Weinberger, K. Q. and Zheng, A. X. (2019), “Gradient Boosted Feature Selection”.
- Yaliang, L., Jing, G., Qi, L. and Wei, F. (2015), “Ensemble Learning”, pp. 484-504.
- Yang, B., Guo, Q., Wang, F., Cai, K., Bao, X. and Chu, J. A. (2017), “80-gene set potentially predicts the relapse in laryngeal carcinoma optimized by support vector machine”, *Cancer Biomarkers*, Vol. 19, No. 1, pp. 65-73.
- Yarney J., Aryeetey, N. A., Mensah, A., Kitcher, E. D., Vanderpuye, V., Aidoo, C. Baidoo, K. (2017), “Cancers of the Head & Neck: Does concurrent chemoradiotherapy preceded by chemotherapy improve survival in locally advanced nasopharyngeal cancer patients? Experience from Ghana”, *BioMed Central*, No. 2, Vol. 4, pp. 1-7.
- Zang, W., Zhang, P., Zhou, C. and Guo, L. (2014), “Comparative study between incremental and ensemble learning on data streams: Case study”, *Journal of Big Data*, Vol. 1, No. 1, pp. 5.



Zheng, L., Zheng, W., Li, G. and Xu, Y. (2022), “Evaluating the significance of samples in deep learning-based transient stability assessment”, *Frontiers in Energy Research*, pp. 1-11.

Zhou, Z. H. (2012), “Ensemble Methods: Foundations and Algorithms”, Chapman and Hall/CRC Machine Learning & Pattern Recognition Series.



APPENDICES

APPENDIX A GRAPHS FOR PROGNOSTIC FEATURES R CODE

```
# Define vectors
Drink <- c(48, 77, 0)
Smoke <- c(38, 87, 0)
Chew <- c(24, 93, 8)
History <- c(16, 84, 25)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, Drink, Smoke, Chew, History)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(Drink, type = "o", col="orange", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:3,
     lab=c("Yes", "No", "NA"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=0.1*10:g_range[2])

# Create box around plot
box()

lines(Smoke, type = "o", pch=22, lty=2, col="red")
lines(Chew, type = "o", pch=22, lty=2, col="green")
lines(History, type = "o", pch=22, lty=2, col="pink")
lines(Recurrence, type = "o", pch=22, lty=2, col="blue")

title(main = "Features:Drink, Smoke, Chew, History",
      col.main="black", font.main=1)

title(xlab = "Categories", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "No. of Instances", col.lab=rgb(0.2,0.2,0.3))

legend(2.5, g_range[2], c("Drink", "Smoke", "Chew", "History"),
      cex = 0.7, col = c("orange", "red", "green", "pink"), pch = 21:22, lty = 1:2);

#####
# Define vectors
p16 <- c(74, 36, 15)
p63 <- c(60, 56, 9)
```

```

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, p16, p63)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(p16, type = "o", col="red", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:3,
     lab=c("Yes", "No", "NA"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=0.1*10:g_range[2])

# Create box around plot
box()

lines(p63, type = "o", pch=22, lty=2, col="blue")
lines(Chew, type = "o", pch=22, lty=2, col="green")
lines(History, type = "o", pch=22, lty=2, col="pink")
lines(Recurrence, type = "o", pch=22, lty=2, col="blue")

title(main = "Features:p16, p63",
      col.main="black", font.main=1)

title(xlab = "Categories", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "No. of Instances", col.lab=rgb(0.2,0.2,0.3))

legend(2.5, g_range[2], c("p16", "p63"),
      cex = 0.7, col = c("red", "blue"), pch = 21:22, lty = 1:2);

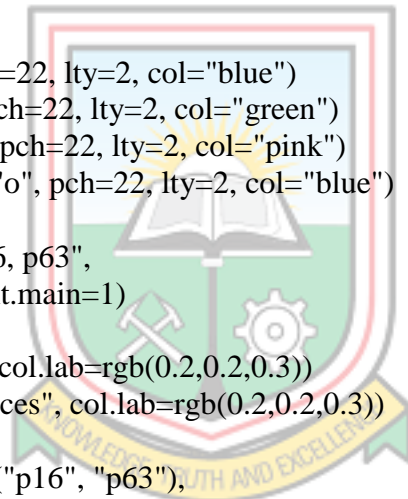
#####
# Define vectors
p16 <- c(74, 36, 15)
p63 <- c(60, 56, 9)
Nodes <- c(61, 44, 20)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, p16, p63, Nodes)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(p16, type = "o", col="red", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:3,

```



```

lab=c("Positive", "Negative", "NA"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=1*10:g_range[2])

# Create box around plot
box()

lines(p63, type = "o", pch=22, lty=2, col="blue")
lines(Nodes, type = "o", pch=22, lty=2, col="green")
lines(History, type = "o", pch=22, lty=2, col="pink")
lines(Recurrence, type = "o", pch=22, lty=2, col="blue")

title(main = "Features:p16, p63, Nodes",
      col.main="black", font.main=1)

title(xlab = "Categories", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "No. of Instances", col.lab=rgb(0.2,0.2,0.3))

legend(2.5, g_range[2], c("p16", "p63", "Nodes"),
      cex = 0.7, col = c("red", "blue", "green"), pch = 21:22, lty = 1:2);

#####
# Define vectors
Site <- c(3, 40, 70, 12)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, Site)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(Site, type = "o", col="red", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:4,
     lab=c("HPC", "Larynx", "NPC", "OPC"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=1*10:g_range[2])

# Create box around plot
box()

lines(p63, type = "o", pch=22, lty=2, col="blue")
lines(Nodes, type = "o", pch=22, lty=2, col="green")

```



```

lines(History, type = "o", pch=22, lty=2, col="pink")
lines(Recurrence, type = "o", pch=22, lty=2, col="blue")

title(main = "Feature:Site",
      col.main="black", font.main=1)

title(xlab = "Categories", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "No. of Instances", col.lab=rgb(0.2,0.2,0.3))

legend(2.5, g_range[2], c("Site"),
      cex = 0.7, col = c("red"), pch = 21:22, lty = 1:2);

#####
# Define vectors
Stage <- c(7, 23, 33, 62)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, Stage)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(Stage, type = "o", col="red", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:4,
     lab=c("I", "II", "III", "IV"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=1*10:g_range[2])

# Create box around plot
box()

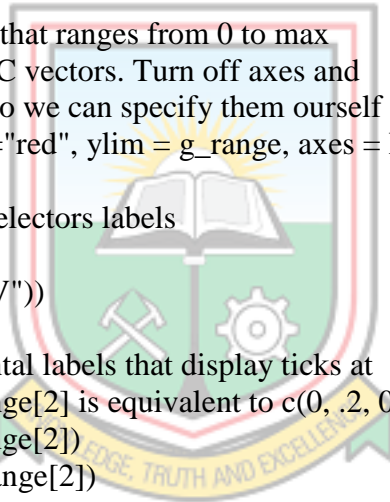
lines(p63, type = "o", pch=22, lty=2, col="blue")
lines(Nodes, type = "o", pch=22, lty=2, col="green")
lines(History, type = "o", pch=22, lty=2, col="pink")
lines(Recurrence, type = "o", pch=22, lty=2, col="blue")

title(main = "Feature:Stage",
      col.main="black", font.main=1)

title(xlab = "Categories", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "No. of Instances", col.lab=rgb(0.2,0.2,0.3))

legend(2.5, g_range[2], c("Site"),
      cex = 0.7, col = c("red"), pch = 21:22, lty = 1:2);

```



```
#####
# Define vectors
Grade <- c(18, 31, 76)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, Grade)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(Grade, type = "o", col="red", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:3,
     lab=c("G1", "G2", "G3"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=1*10:g_range[2])

# Create box around plot
box()

lines(p63, type = "o", pch=22, lty=2, col="blue")
lines(Nodes, type = "o", pch=22, lty=2, col="green")
lines(History, type = "o", pch=22, lty=2, col="pink")
lines(Recurrence, type = "o", pch=22, lty=2, col="blue")

title(main = "Feature:Tumor Grade",
      col.main="black", font.main=1)

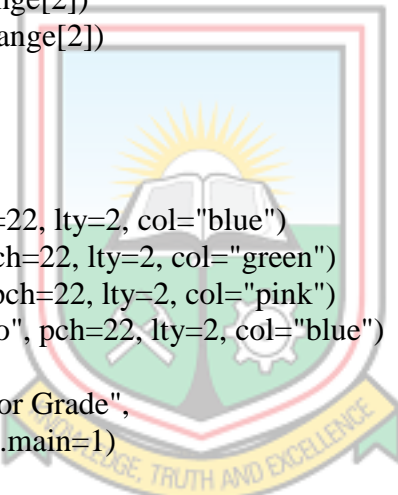
title(xlab = "Categories", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "No. of Instances", col.lab=rgb(0.2,0.2,0.3))

legend(2.5, g_range[2], c("Site"),
      cex = 0.7, col = c("red"), pch = 21:22, lty = 1:2);

#####
# Define 2 vectors
PIN <- c(34, 14, 43, 34)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, PIN)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(PIN, type = "o", col="red", ylim = g_range, axes = FALSE, ann = FALSE)
```



```

# Make x axis using base selectors labels
axis(1, at=1:4,
     lab=c("N0", "N1", "N2", "N3"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=1*10:g_range[2])

# Create box around plot
box()

lines(p63, type = "o", pch=22, lty=2, col="blue")
lines(Nodes, type = "o", pch=22, lty=2, col="green")
lines(History, type = "o", pch=22, lty=2, col="pink")
lines(Recurrence, type = "o", pch=22, lty=2, col="blue")

title(main = "Feature:PIN",
      col.main="black", font.main=1)

title(xlab = "Categories", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "No. of Instances", col.lab=rgb(0.2,0.2,0.3))

legend(2.5, g_range[2], c("Site"),
      cex = 0.7, col = c("red"), pch = 21:22, lty = 1:2);
#####
# Define vectors
HPV <- c(13, 27, 11, 2, 72)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, HPV)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(HPV, type = "o", col="red", ylim = g_range, axes = FALSE, ann = FALSE)

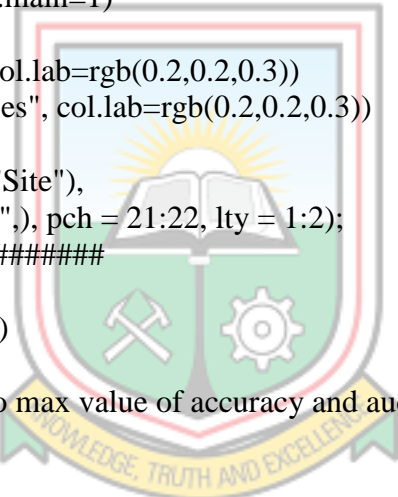
# Make x axis using base selectors labels
axis(1, at=1:5,
     lab=c("HPV11", "HPV16", "HPV18", "HPV6", "NA"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=1*10:g_range[2])

# Create box around plot
box()

lines(p63, type = "o", pch=22, lty=2, col="blue")

```



```

lines(Nodes, type = "o", pch=22, lty=2, col="green")
lines(History, type = "o", pch=22, lty=2, col="pink")
lines(Recurrence, type = "o", pch=22, lty=2, col="blue")

title(main = "Feature:HPV",
      col.main="black", font.main=1)

title(xlab = "Categories", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "No. of Instances", col.lab=rgb(0.2,0.2,0.3))

legend(2.5, g_range[2], c("Site"),
      cex = 0.7, col = c("red"), pch = 21:22, lty = 1:2);

#####
# Define vectors
Treat <- c(26, 9, 47, 33, 4, 2, 4)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, Treat)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(Treat, type = "o", col="red", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:7,
     lab=c("CCRT", "Chemo", "CRT", "RT", "Surg+CCRT", "Surg+CRT", "Surg+RT"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=1*10:g_range[2])

# Create box around plot
box()

lines(p63, type = "o", pch=22, lty=2, col="blue")
lines(Nodes, type = "o", pch=22, lty=2, col="green")
lines(History, type = "o", pch=22, lty=2, col="pink")
lines(Recurrence, type = "o", pch=22, lty=2, col="blue")

title(main = "Feature:Treat",
      col.main="black", font.main=1)

title(xlab = "Categories", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "No. of Instances", col.lab=rgb(0.2,0.2,0.3))

legend(2.5, g_range[2], c("Site"),
      cex = 0.7, col = c("red"), pch = 21:22, lty = 1:2);

```

```
#####
# Define vectors
Gender <- c(92, 33)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, Gender)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(Gender, type = "o", col="red", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:2,
     lab=c("Male", "Female"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=1*10:g_range[2])

# Create box around plot
box()

lines(p63, type = "o", pch=22, lty=2, col="blue")
lines(Nodes, type = "o", pch=22, lty=2, col="green")
lines(History, type = "o", pch=22, lty=2, col="pink")
lines(Recurrence, type = "o", pch=22, lty=2, col="blue")

title(main = "Feature:Gender",
      col.main="black", font.main=1)

title(xlab = "Categories", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "No. of Instances", col.lab=rgb(0.2,0.2,0.3))

legend(2.5, g_range[2], c("Site"),
      cex = 0.7, col = c("red"), pch = 21:22, lty = 1:2);

#####
# Define vectors
Invasion <- c(47, 74, 4)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, Invasion)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(Invasion, type = "o", col="red", ylim = g_range, axes = FALSE, ann = FALSE)
```

```

# Make x axis using base selectors labels
axis(1, at=1:3,
     lab=c("Cohesion", "Non-cohesion", "NA"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=1*10:g_range[2])

# Create box around plot
box()

lines(p63, type = "o", pch=22, lty=2, col="blue")
lines(Nodes, type = "o", pch=22, lty=2, col="green")
lines(History, type = "o", pch=22, lty=2, col="pink")
lines(Recurrence, type = "o", pch=22, lty=2, col="blue")

title(main = "Feature:Invasion",
      col.main="black", font.main=1)

title(xlab = "Categories", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "No. of Instances", col.lab=rgb(0.2,0.2,0.3))

legend(2.5, g_range[2], c("Site"),
      cex = 0.7, col = c("red"), pch = 21:22, lty = 1:2);

#####
# Define vectors
Age <- c(51, 74)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, Age)

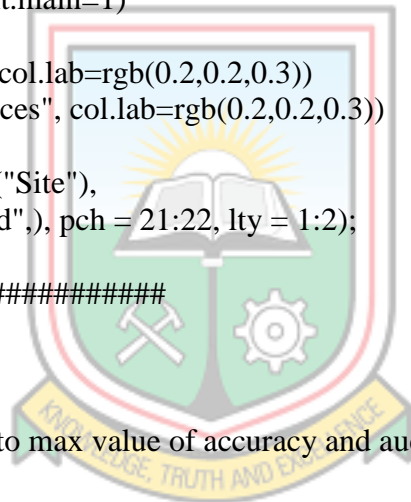
# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(Age, type = "o", col="red", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:2,
     lab=c("15-45 yrs", ">45 yrs"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=1*10:g_range[2])

# Create box around plot
box()

```



```

lines(p63, type = "o", pch=22, lty=2, col="blue")
lines(Nodes, type = "o", pch=22, lty=2, col="green")
lines(History, type = "o", pch=22, lty=2, col="pink")
lines(Recurrence, type = "o", pch=22, lty=2, col="blue")

title(main = "Feature:Age",
      col.main="black", font.main=1)

title(xlab = "Categories", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "No. of Instances", col.lab=rgb(0.2,0.2,0.3))

legend(2.5, g_range[2], c("Site"),
      cex = 0.7, col = c("red"), pch = 21:22, lty = 1:2);

#####
# Define vectors
Size <- c(75, 46, 4)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, Size)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(Size, type = "o", col="red", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:3,
     lab=c("0-4cm", ">4cm", "NA"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=1*10:g_range[2])

# Create box around plot
box()

lines(p63, type = "o", pch=22, lty=2, col="blue")
lines(Nodes, type = "o", pch=22, lty=2, col="green")
lines(History, type = "o", pch=22, lty=2, col="pink")
lines(Recurrence, type = "o", pch=22, lty=2, col="blue")

title(main = "Feature:Size",
      col.main="black", font.main=1)

title(xlab = "Categories", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "No. of Instances", col.lab=rgb(0.2,0.2,0.3))

legend(2.5, g_range[2], c("Site"),

```



```

    cex = 0.7, col = c("red"), pch = 21:22, lty = 1:2);

#####
# Define vectors
Recurrence <- c(61, 60, 4)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, Recurrence)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(Recurrence, type = "o", col="blue", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:3,
     lab=c("Yes", "No", "NA"))
# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=1*10:g_range[2])

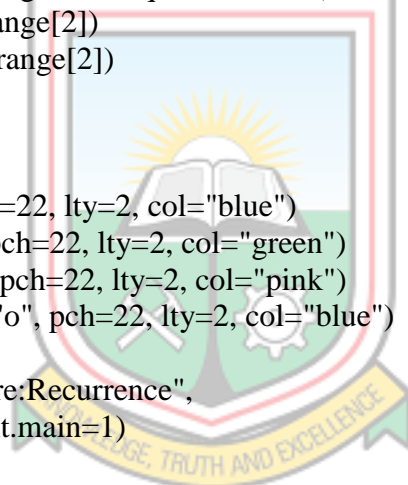
# Create box around plot
box()
lines(p63, type = "o", pch=22, lty=2, col="blue")
lines(Nodes, type = "o", pch=22, lty=2, col="green")
lines(History, type = "o", pch=22, lty=2, col="pink")
lines(Recurrence, type = "o", pch=22, lty=2, col="blue")

title(main = "Label Feature:Recurrence",
      col.main="black", font.main=1)

title(xlab = "Categories", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "No. of Instances", col.lab=rgb(0.2,0.2,0.3))

legend(2.5, g_range[2], c("Site"),
      cex = 0.7, col = c("red"), pch = 21:22, lty = 1:2);

```



APPENDIX B DATA CLEANSING R CODE

```
library(dummies)
library(Hmisc)

# Data
library(data.table)
data <- fread(file.choose(), data.table = T)
str(data)
View(data)
print(data)

# Single dummy Features
data$Drink[data$Drink == "No"] = "0"
data$Drink[data$Drink == "Yes"] = "1"

data$Smoke[data$Smoke == "No"] = "0"
data$Smoke[data$Smoke == "Yes"] = "1"

data$Chew[data$Chew == "No"] = "0"
data$Chew[data$Chew == "Yes"] = "1"

data$Invasion[data$Invasion=="Non cohesive"]="0"
data$Invasion[data$Invasion=="Cohesive"]="1"

data$Gender[data$Gender=="F"]="0"
data$Gender[data$Gender=="M"]="1"

data$Nodes[data$Nodes=="neg"]="0"
data$Nodes[data$Nodes=="pos"]="1"

data$History[data$History=="No"]="0"
data$History[data$History=="Yes"]="1"

data$HPV[data$HPV=="HPV11"]="0"
data$HPV[data$HPV=="HPV16"]="1"

data$p16[data$p16=="neg"]="0"
data$p16[data$p16=="pos"]="1"

data$p63[data$p63=="neg"]="0"
data$p63[data$p63=="pos"]="1"

data$Recurrence[data$Recurrence=="No"]="0"
data$Recurrence[data$Recurrence=="Yes"]="1"

data$Age <- ifelse(data$Age < 45, 0, 1)

data$Size <- ifelse(data$Size < 4, 0, 1)

# Mode imputation: Impute the missing values
```

```
data$Drink <- impute(data$Drink, 0)
data$Smoke <- impute(data$Smoke, 0)
data$Chew <- impute(data$Chew, 0)
data$Nodes <- impute(data$Nodes, 1)
data$Size <- impute(data$Size, 0)
data$Invasion <- impute(data$Invasion, 0)
data$History <- impute(data$History, 0)
data$HPV <- impute(data$HPV, 1)
data$p16 <- impute(data$p16, 1)
data$p63 <- impute(data$p63, 1)
data$TreatCCRT <- impute(data$TreatCCRT, 0)
data$TreatChemo <- impute(data$TreatChemo, 0)
data$TreatCRT <- impute(data$TreatCRT, 0)
data$TreatRT <- impute(data$TreatRT, 0)
data$`TreatSurg+CCRT` <- impute(data$`TreatSurg+CCRT`, 0)
data$`TreatSurg+CRT` <- impute(data$`TreatSurg+CRT`, 0)
data$`TreatSurg+RT` <- impute(data$`TreatSurg+RT`, 0)
data$Recurrence <- impute(data$Recurrence, 0)
```



APPENDIX C BASE MODELS SELECTION R CODE

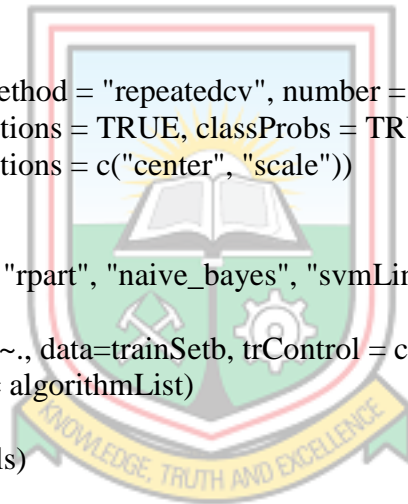
```
# Load and split the data
library(data.table)
data <- fread(file.choose(), data.table = T)
str(data)
summary(data)
#####
# Splitting training set into two parts based on outcome: 70% and 30%
indexb <- createDataPartition(datab$Class, p=0.70, list = FALSE)
trainSetb <- datab[indexb,]
testSetb <- datab[-indexb,]
dim(trainSetb)
dim(testSetb)

# Number of CV folds (to generate level-one data for stacking)
nfolds <- 10

# Example of stacking algorithms
# Create sub_models
seed <- 7
control <- trainControl(method = "repeatedcv", number = 10, repeats = 1,
  savePredictions = TRUE, classProbs = TRUE,
  preProcOptions = c("center", "scale"))

set.seed(seed)
algorithmList <- c("knn", "rpart", "naive_bayes", "svmLinear")

models <- caretList(Class~., data=trainSetb, trControl = control,
  methodList = algorithmList)
summary(models)
result <- resamples(models)
summary(result)
dotplot(result)
```



APPENDIX D MODELS LEARNING R CODE

```
library(tidyverse)
library(h2o) # for fitting stacked models
library(rsample) # for creating our train-test splits
library(recipes) # for minor feature engineering tasks
library(xgboost)
library(dplyr)
library(plotROC)
library(pROC)
library(lime)
library(plotly)
library(devtools)
library(h2oEnsemble)
library(ggthemes)
library(rio)
library(knitr)
library(viridis)
library(ggplot2)
library(ROCR)
library(Hmisc)
library(MASS)
library(hier.part)
library(car)

# Load and split the data
library(data.table)
data <- fread(file.choose(), data.table = T)
str(data)
summary(data)

# Single dummy Features
data$Age <- ifelse(data$Age < 45, 0, 1)
data$Size <- ifelse(data$Size < 4, 0, 1)
data$Recurrence <- as.factor(data$Recurrence)
str(data)

#####
split <- initial_split(data, strata = "Recurrence")
data_train <- training(split)
data_test <- testing(split)

# Make sure we have consistent categorical levels
blueprint <- recipe(Recurrence ~., data = data_train) %>%
  step_other(all_nominal(), threshold = 0.05)

h2o.init()
#####
# Convert data to H2OFrame
data_h2o <- as.h2o(data)
```



```

# Create training & test sets for h2o
train_h2o <- prep(blueprint, training = data_train, retain = TRUE) %>%
  juice() %>%
  as.h2o()
test_h2o <- prep(blueprint, training = data_train) %>%
  bake(new_data = data_test) %>%
  as.h2o()

# Get response column and feature names
Y <- "Recurrence"
X <- setdiff(names(data_train), Y)

#####
# Learn base learners
# Train & cross-validate a GBM model
best_gbm <- h2o.gbm(
  x = X, y = Y, training_frame = train_h2o, ntrees = 5000, learn_rate = 0.01,
  max_depth = 10, min_rows = 3, sample_rate = 0.9, nfolds = 10,
  fold_assignment = "Modulo", keep_cross_validation_predictions = TRUE,
  seed = 123, stopping_rounds = 50, stopping_metric = "logloss",
  stopping_tolerance = 0.0001
)
#####
# Train & cross-validate a RF model
best_rf <- h2o.randomForest(
  x = X, y = Y, training_frame = train_h2o, ntrees = 5000, mtries = 3,
  max_depth = 10, min_rows = 3, sample_rate = 0.8, nfolds = 10,
  fold_assignment = "Modulo", keep_cross_validation_predictions = TRUE,
  seed = 123, stopping_rounds = 50, stopping_metric = "logloss",
  stopping_tolerance = 0.00001
)
#####
# Train & cross-validate Deep Neural Network model
best_dl <- h2o.deeplearning(
  x = X, y = Y, training_frame = train_h2o, nfolds = 10, activation = "rectifier",
  fold_assignment = "Modulo", keep_cross_validation_predictions = TRUE,
  seed = 123, stopping_rounds = 50, tweedie_power = 1.5, stopping_metric = "logloss",
  stopping_tolerance = 0.0001
)
#####
# Train & cross-validate naiveBayes model
best_naiv <- h2o.naiveBayes(
  x = X, y = Y, training_frame = train_h2o, nfolds = 10, laplace = 0.2,
  fold_assignment = "Modulo", keep_cross_validation_predictions = TRUE,
  seed = 123
)
#####
# Train & cross-validate a GLM model
best_glm <- h2o.glm(
  x = X, y = Y, training_frame = train_h2o, alpha = 0.1,

```

```

remove_collinear_columns = TRUE, nfolds = 10, fold_assignment = "Modulo",
keep_cross_validation_predictions = TRUE, seed = 123
)
#####
# Train a stacked GBM ensemble
metalearner_gbm <- h2o.stackedEnsemble(
  x = X, y = Y, training_frame = train_h2o, model_id = "HESCA Model",
  base_models = list(best_glm, best_rf, best_gbm, best_dl, best_naiv),
  keep_levelone_frame = TRUE, metalearner_algorithm = "gbm")
#####
# Train a stacked randomForest ensemble
metalearner_rf <- h2o.stackedEnsemble(
  x = X, y = Y, training_frame = train_h2o, model_id = "my_rf_ensemble",
  base_models = list(best_glm, best_rf, best_gbm, best_dl, best_naiv),
  keep_levelone_frame = TRUE, metalearner_algorithm = "drf")
#####
# Train a stacked Deep Neural Network ensemble
metalearner_dl <- h2o.stackedEnsemble(
  x = X, y = Y, training_frame = train_h2o, model_id = "my_dl_ensemble",
  base_models = list(best_glm, best_rf, best_gbm, best_dl, best_naiv),
  keep_levelone_frame = TRUE, metalearner_algorithm = "deeplearning")
#####
# Train a stacked naive_Bayes ensemble
metalearner_naiv <- h2o.stackedEnsemble(
  x = X, y = Y, training_frame = train_h2o, model_id = "my_naiv_ensemble",
  base_models = list(best_glm, best_rf, best_gbm, best_dl, best_naiv),
  keep_levelone_frame = TRUE, metalearner_algorithm = "naiveBayes")
#####
# Train a stacked Logit ensemble
metalearner_glm <- h2o.stackedEnsemble(
  x = X, y = Y, training_frame = train_h2o, model_id = "my_glm_ensemble",
  base_models = list(best_glm, best_rf, best_gbm, best_dl, best_naiv),
  keep_levelone_frame = TRUE, metalearner_algorithm = "glm")
#####
##Train baseline stacked enesmble
# stack GBM & RF using GLM
meta_glm1 <- h2o.stackedEnsemble(
  x = X, y = Y, training_frame = train_h2o, model_id = "my_glm1_ensemble",
  base_models = list(best_gbm, best_rf),
  keep_levelone_frame = TRUE, metalearner_algorithm = "gbm")
#####
# Train a stacked GBM, DRF & DNN using GLM
meta_glm2 <- h2o.stackedEnsemble(
  x = X, y = Y, training_frame = train_h2o, model_id = "my_glm2_ensemble",
  base_models = list(best_gbm, best_rf, best_dl),
  keep_levelone_frame = TRUE, metalearner_algorithm = "drf")
#####
#Train State-of-the-art stacked enesmble
# stack GBM, RF, DNN & GLM using GBM
meta_gbm <- h2o.stackedEnsemble(

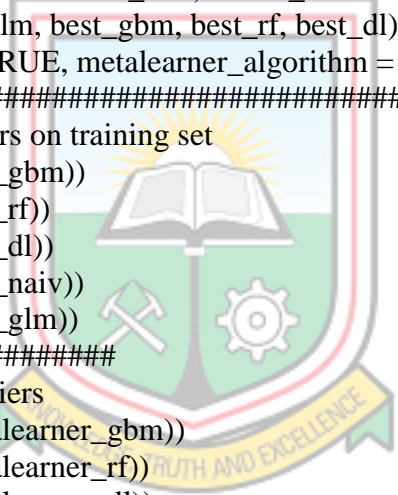
```



```

x = X, y = Y, training_frame = train_h2o, model_id = "my_gbm_ensemble",
base_models = list(best_gbm, best_rf, best_dl, best_glm),
keep_levelone_frame = TRUE, metalearner_algorithm = "gbm")
#####
# Train a stacked randomForest ensemble
meta_rf <- h2o.stackedEnsemble(
  x = X, y = Y, training_frame = train_h2o, model_id = "my_rf_ensemble",
  base_models = list(best_glm, best_gbm, best_rf, best_dl),
  keep_levelone_frame = TRUE, metalearner_algorithm = "drf")
#####
# Train a stacked Deep Neural Network ensemble
meta_dl <- h2o.stackedEnsemble(
  x = X, y = Y, training_frame = train_h2o, model_id = "my_dl_ensemble",
  base_models = list(best_glm, best_gbm, best_rf, best_dl),
  keep_levelone_frame = TRUE, metalearner_algorithm = "deeplearning")
#####
# Train a stacked Logit ensemble
meta_glm <- h2o.stackedEnsemble(
  x = X, y = Y, training_frame = train_h2o, model_id = "my_glm_ensemble",
  base_models = list(best_glm, best_gbm, best_rf, best_dl),
  keep_levelone_frame = TRUE, metalearner_algorithm = "glm")
#####
# Plot AUC for base learners on training set
plot(h2o.performance(best_gbm))
plot(h2o.performance(best_rf))
plot(h2o.performance(best_dl))
plot(h2o.performance(best_naiv))
plot(h2o.performance(best_glm))
#####
#plot AUC for meta classifiers
plot(h2o.performance(metalearner_gbm))
plot(h2o.performance(metalearner_rf))
plot(h2o.performance(metalearner_dl))
plot(h2o.performance(metalearner_naiv))
plot(h2o.performance(metalearner_glm))
#####
# plot AUC for baseline stacked ensemble
plot(h2o.performance(meta_glm1))
plot(h2o.performance(meta_glm2))
#####
# plot AUC for state-of-the-art
plot(h2o.performance(meta_gbm))
plot(h2o.performance(meta_rf))
plot(h2o.performance(meta_dl))
plot(h2o.performance(meta_glm))
#####
# Check model correlation
data.frame(

```



```

GBM_pred =
as.vector(h2o.getFrame(best_gbm@model$cross_validation_holdout_predictions_frame_id$name)),
DRF_pred =
as.vector(h2o.getFrame(best_rf@model$cross_validation_holdout_predictions_frame_id$name)),
DNN_pred =
as.vector(h2o.getFrame(best_dl@model$cross_validation_holdout_predictions_frame_id$name)),
Naive_pred =
as.vector(h2o.getFrame(best_naiv@model$cross_validation_holdout_predictions_frame_id$name)),
GLM_pred =
as.vector(h2o.getFrame(best_glm@model$cross_validation_holdout_predictions_frame_id$name))
) %>% cor()
#####
# Compare base learner performance on the test set
perf_gbm_test <- h2o.performance(best_gbm, newdata = test_h2o)
perf_rf_test <- h2o.performance(best_rf, newdata = test_h2o)
perf_dl_test <- h2o.performance(best_dl, newdata = test_h2o)
perf_glm_test <- h2o.performance(best_glm, newdata = test_h2o)
perf_naiv_test <- h2o.performance(best_naiv, newdata = test_h2o)
#####
# Compare metalearner performance on the test set
perf_mgbm_test <- h2o.performance(metalearner_gbm, newdata = test_h2o)
perf_mrf_test <- h2o.performance(metalearner_rf, newdata = test_h2o)
perf_md1_test <- h2o.performance(metalearner_dl, newdata = test_h2o)
perf_mnaiv_test <- h2o.performance(metalearner_naiv, newdata = test_h2o)
perf_mglm_test <- h2o.performance(metalearner_glm, newdata = test_h2o)
#####
# Performance of baseline stacked ensemble on test set
perf_mglm1_test <- h2o.performance(meta_glm1, newdata = test_h2o)
perf_mglm2_test <- h2o.performance(meta_glm2, newdata = test_h2o)
#####
# Performance of state-of-the-art stacked ensemble on test set
perf_mgbms_test <- h2o.performance(meta_gbm, newdata = test_h2o)
perf_mrfs_test <- h2o.performance(meta_rf, newdata = test_h2o)
perf_mdls_test <- h2o.performance(meta_dl, newdata = test_h2o)
perf_mglms_test <- h2o.performance(meta_glm, newdata = test_h2o)
#####
# Plot ROC Curve for base learner on test set
perf_gbm_roc <- h2o.performance(best_gbm, test_h2o)
plot(perf_gbm_roc, type = "roc")
perf_rf_roc <- h2o.performance(best_rf, test_h2o)
plot(perf_rf_roc, type = "roc")
perf_dl_roc <- h2o.performance(best_dl, test_h2o)
plot(perf_dl_roc, type = "roc")
perf_naiv_roc <- h2o.performance(best_naiv, test_h2o)
plot(perf_naiv_roc, type = "roc")

```

```

perf_glm_rc <- h2o.performance(best_glm, test_h2o)
plot(perf_glm_rc, type = "roc")
# Plot ROC Curve for meta learner on test set
perf_mgbm_rc <- h2o.performance(metalearner_gbm, test_h2o)
plot(perf_mgbm_rc, type = "roc")
perf_mrf_rc <- h2o.performance(metalearner_rf, test_h2o)
plot(perf_mrf_rc, type = "roc")
perf_mdل_rc <- h2o.performance(metalearner_dl, test_h2o)
plot(perf_mdل_rc, type = "roc")
perf_mnaiv_rc <- h2o.performance(metalearner_naiv, test_h2o)
plot(perf_mnaiv_rc, type = "roc")
perf_mglm_rc <- h2o.performance(metalearner_glm, test_h2o)
plot(perf_mglm_rc, type = "roc")
#####
# Plot ROC Curve for baseline stacked ensemble on test set
perf_mglm1_rc <- h2o.performance(meta_glm1, test_h2o)
plot(perf_mglm1_rc, type = "roc")
perf_mglm2_rc <- h2o.performance(meta_glm2, test_h2o)
plot(perf_mglm2_rc, type = "roc")
#####
# Plot ROC Curve for state-of-the-art on test set
perf_mgbms_rc <- h2o.performance(meta_gbm, test_h2o)
plot(perf_mgbms_rc, type = "roc")
perf_mrfs_rc <- h2o.performance(meta_rf, test_h2o)
plot(perf_mrfs_rc, type = "roc")
perf_mdls_rc <- h2o.performance(meta_dl, test_h2o)
plot(perf_mdls_rc, type = "roc")
perf_mglms_rc <- h2o.performance(meta_glm, test_h2o)
plot(perf_mglms_rc, type = "roc")
#####
# Get results from base classifiers
get_logloss <- function(model){
  results <- h2o.performance(model, newdata = test_h2o)
  results@metrics$logloss
}
list(best_gbm, best_rf, best_dl, best_naiv, best_glm) %>%
  purrr::map_dbl(get_logloss)

# stacked results
h2o.performance(metalearner_gbm, newdata = test_h2o)@metrics$logloss
#####
# Define GBM hyperparameter grid
hyper_grid_gbm <- list(
  max_depth = c(7, 9),
  min_rows = c(1, 3, 5),
  learn_rate = c(0.01, 0.1),
  sample_rate = c(0.5, 0.75, 1),
  col_sample_rate = c(0.8, 0.9, 1)
)

```

```

# Define random grid search criteria
search_criteria <- list(
  strategy = "RandomDiscrete",
  max_models = 25,
  seed = 123
)

# Build random grid search
random_grid <- h2o.grid(
  algorithm = "gbm", grid_id = "gbm_grid", x = X, y = Y,
  training_frame = train_h2o, hyper_params = hyper_grid_gbm,
  search_criteria = search_criteria, ntrees = 5000,
  stopping_metric = "logloss", stopping_rounds = 10, stopping_tolerance = 0,
  nfolds = 10, fold_assignment = "Modulo", keep_cross_validation_predictions = TRUE,
  seed = 123
)

# Sort results by Logloss
random_grid_perf <- h2o.getGrid(
  grid_id = "gbm_grid",
  sort_by = "logloss",
  decreasing = TRUE
)

# Grab the model_id for the top model, chosen by validation error
best_model_id <- random_grid_perf@model_ids[[1]]
best_model <- h2o.getModel(best_model_id)
h2o.performance(best_model, newdata = test_h2o)
#####
# Train a stacked ensemble using the GBM grid
ensemble_gbm <- h2o.stackedEnsemble(
  x = X, y = Y, training_frame = train_h2o, model_id = "ensemble_gbm_grid",
  base_models = random_grid@model_ids, metalearner_algorithm = "gbm"
)
#####
# Evaluate performance
perf_gbm <- h2o.performance(ensemble_gbm, newdata = test_h2o)
#####
# Look at hyperparameters for best model
print(best_model@model[["model_summary"]])
#####
# Generate a random grid of models and stack them together
# GBM Hyperparameters
learn_rate_opt <- c(0.01, 0.02, 0.03)
max_depth_opt <- c(3, 4, 5, 6, 9)
sample_rate_opt <- c(0.7, 0.8, 0.9, 1.0)
col_sample_rate_opt <- c(0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8)
hyper_params <- list(learn_rate = learn_rate_opt,
  max_depth = max_depth_opt,
  sample_rate = sample_rate_opt,

```

```

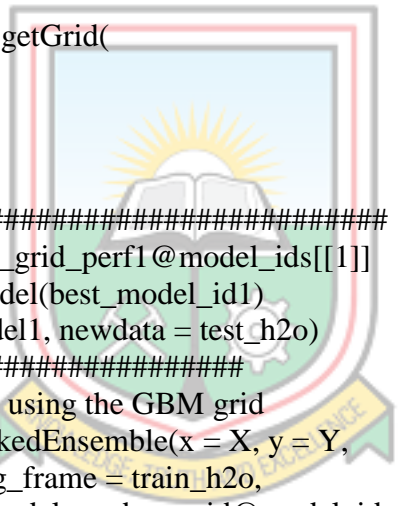
col_sample_rate = col_sample_rate_opt)

search_crit <- list(strategy = "RandomDiscrete",
  max_models = 25,
  seed = 123)

gbm_grid <- h2o.grid(algorithm = "gbm",
  grid_id = "gbm_grid_binomial",
  x = X,
  y = Y,
  training_frame = train_h2o,
  ntrees = 50,
  seed = 1,
  nfolds = nfolds,
  keep_cross_validation_predictions = TRUE,
  hyper_params = hyper_params,
  search_criteria = search_crit)
#####
# Sort results by Logloss
random_grid_perf1 <- h2o.getGrid(
  grid_id = "gbm_grid",
  sort_by = "logloss",
  decreasing = TRUE
)
#####
best_model_id1 <- random_grid_perf1@model_ids[[1]]
best_model1 <- h2o.getModel(best_model_id1)
h2o.performance(best_model1, newdata = test_h2o)
#####
# Train a stacked ensemble using the GBM grid
ensemble_gbm <- h2o.stackedEnsemble(x = X, y = Y,
  training_frame = train_h2o,
  base_models = gbm_grid@model_ids,
  metalearner_algorithm = "gbm")
#####
# Evaluate performance
per <- h2o.performance(ensemble_gbm, newdata = test_h2o)
#####
# Hyperparameters for DRF
# set hyperparameter grid
# Define DRF hyperparameter grid
hyper_grid_drf <- list(
  max_depth = c(9, 30),
  min_rows = c(1, 5, 10),
  sample_rate = c(0.5, 0.75, 1),
  col_sample_rate_per_tree = c(0.8, 0.9, 1)
)

# Define random grid search criteria
search_criteria_drf <- list(

```



```

strategy = "RandomDiscrete",
max_models = 25,
seed = 123
)

# Build random grid search
random_grid_drf <- h2o.grid(
  algorithm = "randomForest", grid_id = "drf_grid", x = X, y = Y,
  training_frame = train_h2o, hyper_params = hyper_grid_drf,
  search_criteria = search_criteria_drf, ntrees = 5000,
  stopping_metric = "logloss", stopping_rounds = 10, stopping_tolerance = 0,
  nfolds = 10, fold_assignment = "Modulo", keep_cross_validation_predictions = TRUE,
  seed = 123
)

# Sort results by Logloss
random_grid_perf_drf <- h2o.getGrid(
  grid_id = "drf_grid",
  sort_by = "logloss",
  decreasing = TRUE
)

# Grab the model_id for the top model, chosen by validation error
best_model_id_drf <- random_grid_perf_drf@model_ids[[25]]
best_model_drf <- h2o.getModel(best_model_id_drf)
h2o.performance(best_model_drf, newdata = test_h2o)
#####
# Train a stacked ensemble using the GBM grid
ensemble_drf <- h2o.stackedEnsemble(
  x = X, y = Y, training_frame = train_h2o, model_id = "ensemble_drf_grid",
  base_models = random_grid_drf@model_ids, metalearner_algorithm = "drf"
)
#####
# Evaluate performance
perf_drf <- h2o.performance(ensemble_drf, newdata = test_h2o)
#####
# Hyperparameters for DNN
# Define DRF hyperparameter grid
hyper_grid_dl <- list(
  activation = c("Rectifier", "Maxout", "Tanh"),
  hidden = list(c(5, 5, 5, 5), c(10, 10, 10, 10), c(50, 50, 50, 50), c(100, 100, 100, 100)),
  epochs = c(50, 100, 200),
  l1 = c(0, 1e-3, 1e-5),
  l2 = c(0, 1e-3, 1e-5),
  rate = c(0, 0.1, 0.005, 0.001),
  rate_annealing = c(1e-8, 1e-7, 1e-6),
  rho = c(0.9, 0.95, 0.99, 0.999),
  epsilon = c(1e-10, 1e-8, 1e-6, 1e-4),
  momentum_start = c(0, 0.5),
  momentum_stable = c(0.99, 0.5, 0),

```

```

input_dropout_ratio = c(0, 0.1, 0.2),
max_w2 = c(10, 100, 1000, 3.4028235e+38)
)

# Define random grid search criteria
search_criteria_dl <- list(
  strategy = "RandomDiscrete",
  max_models = 25,
  max_runtime_secs = 900,
  stopping_tolerance = 0.001,
  stopping_rounds = 15,
  seed = 123
)

# Build random grid search
random_grid_dl <- h2o.grid(algorithm = "deeplearning",
  x = X, y = Y, training_frame = train_h2o,
  hyper_params = hyper_grid_dl, validation_frame = test_h2o,
  search_criteria = search_criteria_dl,
  nfolds = 10, fold_assignment = "Modulo",
  seed = 123,
)

# Sort results by Logloss
random_grid_perf_dl <- h2o.getGrid(
  grid_id = "dl_grid",
  sort_by = "logloss",
  decreasing = TRUE
)

# Grab the model_id for the top model, chosen by validation error
best_model_id_dl <- random_grid_perf_dl@model_ids[[1]]
best_model_dl <- h2o.getModel(best_model_id_dl)
h2o.performance(best_model_dl, newdata = test_h2o)
#####
# Grid search hyperparameters for Naive Bayes
hyper_param_naiv <- list(
  laplace = c(0, 0.5, 1, 1.5, 2)
)

threshold = c(0.001, 0.00001, 0.0000001)

# performs the grid search
grid_id_naiv <- "dl_grid"
model_grid_naiv <- h2o.grid(
  algorithm = "naivebayes",
  grid_id = grid_id_naiv,
  training_frame = train_h2o,
  x = X, y = Y,
  hyper_params = hyper_param_naiv
)

```



```

# find the best model and eval its perf
stopping_metric <- 'accuracy'
sorted_models <- h2o.getGrid(
  grid_id = grid_id_naiv,
  sort_by = stopping_metric,
  decreasing = TRUE
)

h2o.confusionMatrix(best_grid_naiv, valid = TRUE, metrics = 'accuracy')

auc <- h2o.auc(best_grid_naiv, valid = TRUE)
fpr <- h2o.fpr( h2o.performance(best_grid_naiv, valid = TRUE) )[['fpr']]
tpr <- h2o.fpr( h2o.performance(best_grid_naiv, valid = TRUE) )[['tpr']]
#####
hyper_grid_rf <- list(nmtree = seq(50, 500, by = 50),
  mtries = seq(3, 5, by = 1),
  max_depth = seq(10, 30, by = 10),
  min_rows = seq(1, 3, by = 1),
  nbins = seq(20, 30, by = 10),
  sample_rate = c(0.55, 0.632, 0.75))

# the number of models is 1620
sapply(hyper_grid_rf, length) %>% prod()

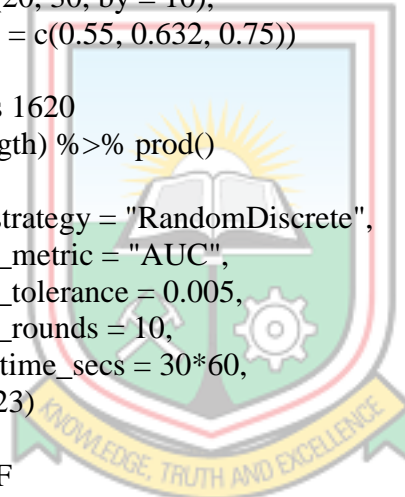
search_criteria_rf <- list(strategy = "RandomDiscrete",
  stopping_metric = "AUC",
  stopping_tolerance = 0.005,
  stopping_rounds = 10,
  max_runtime_secs = 30*60,
  seed = 123)

#Turn parameters for DRF
random_grid <- h2o.grid(algorithm = "randomForest",
  grid_id = "rf_grid",
  x = X, y = Y,
  seed = 123,
  nfolds = nfolds,
  training_frame = train_h2o,
  hyper_params = hyper_grid_rf,
  search_criteria = search_criteria_rf)

# collect the results and sort by my models
grid_perf_rf <- h2o.getGrid(grid_id = "rf_grid",
  sort_by = "logloss",
  decreasing = TRUE)

# Best DRF
best_model_grid <- grid_perf_rf@model_ids[[1]]
best_model_rf <- h2o.getModel(best_model_grid)
h2o.performance(best_model, newdata = test_h2o)

```



APPENDIX E GRAPHS OF MODELS PERFORMOMANCE R CODE

```

# Define vectors
Accuracy <- c(0.91398, 0.82796, 0.83871, 0.79570, 0.79570, 0.96774)
Logloss <- c(0.28381, 0.50211, 0.72000, 0.48505, 0.59259, 0.11718)
Recall <- c(0.90000, 0.72222, 0.65517, 0.60000, 0.60000, 0.90000)
Specificity <- c(0.91781, 0.85333, 0.92188, 0.86765, 0.87879, 1.00000)
Precision <- c(0.75000, 0.54167, 0.79167, 0.62500, 0.66667, 1.00000)
F1_score <- c(0.81000, 0.61905, 0.71698, 0.61224, 0.63156, 0.94737)
AUC <- c(0.93297, 0.74155, 0.87953, 0.77687, 0.72977, 0.99523)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, Precision)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(Precision, type = "o", col="orange", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:6,
     lab=c("GBM", "DRF", "DNN", "GLM", "NB", "HESCA"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=0.1*10:g_range[2])

# Create box around plot
box()

lines(Accuracy, type = "o", pch=22, lty=2, col="red")
lines(AUC, type = "o", pch=22, lty=2, col="blue")
lines(Logloss, type = "o", pch=22, lty=2, col="purple")
lines(F1_score, type = "o", pch=22, lty=2, col="green")
lines(Recall, type = "o", pch=22, lty=2, col="violet")
lines(Precision, type = "o", pch=22, lty=2, col="orange")
lines(Specificity, type = "o", pch=22, lty=2, col="brown")

title(main = "Comparison of Base Models and HESCA Model Performance
on Training Data", col.main="black", font.main=1)

title(xlab = "Models", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "Performance value", col.lab=rgb(0.2,0.2,0.3))

legend(5.05, g_range[2], c("Accuracy", "logloss", "Recall", "Specificity", "Precision",
"F1_score", "AUC"),
     cex = 0.7, col = c("red", "purple", "violet", "brown", "orange", "green", "blue"), pch = 21:22,
     lty = 1:2);
#####

```

```

# Define vectors
Accuracy <- c(0.84375, 0.75000, 0.71875, 0.78125, 0.75000, 0.90625)
Logloss <- c(0.46861, 0.51558, 0.73101, 0.50379, 0.49484, 0.29591)
Recall <- c(0.66667, 0.53846, 0.50000, 0.66667, 0.53333, 0.75000)
Specificity <- c(0.95000, 0.89474, 0.9375, 0.84000, 0.94118, 1.00000)
Precision <- c(0.88889, 0.77778, 0.88889, 0.44444, 0.88889, 1.00000)
F1_score <- c(0.76191, 0.63636, 0.64000, 0.53333, 0.66666, 0.85714)
AUC <- c(0.82850, 0.75362, 0.77778, 0.81401, 0.80193, 0.92512)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, Precision)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(Precision, type = "o", col="orange", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:6,
     lab=c("GBM", "DRF", "DNN", "GLM", "NB", "HESCA"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=0.1*10:g_range[2])

# Create box around plot
box()

lines(Accuracy, type = "o", pch=22, lty=2, col="red")
lines(AUC, type = "o", pch=22, lty=2, col="blue")
lines(Logloss, type = "o", pch=22, lty=2, col="purple")
lines(F1_score, type = "o", pch=22, lty=2, col="green")
lines(Recall, type = "o", pch=22, lty=2, col="violet")
lines(Precision, type = "o", pch=22, lty=2, col="orange")
lines(Specificity, type = "o", pch=22, lty=2, col="brown")

title(main = "Comparison of Base Models and HESCA Model Performance
on Test Data", col.main="black", font.main=1)

title(xlab = "Models", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "Performance value", col.lab=rgb(0.2,0.2,0.3))

legend(5.05, g_range[2], c("Accuracy", "logloss", "Recall", "Specificity", "Precision",
"F1_score", "AUC"),
     cex = 0.6, col = c("red", "purple", "violet", "brown", "orange", "green", "blue"), pch = 21:22,
     lty = 1:2);
#####
# Define vectors
Accuracy <- c(0.91398, 0.92473, 0.9355, 0.96774)

```

```

Logloss <- c(0.34495, 0.30749, 0.2667, 0.11718)
Recall <- c(0.83333, 0.94737, 0.9091, 0.90000)
Specificity <- c(0.94203, 0.91892, 0.9437, 1.00000)
Precision <- c(0.83333, 0.75000, 0.8333, 1.00000)
F1_score <- c(0.83333, 0.83721, 0.86955, 0.94737)
AUC <- c(0.90731, 0.90429, 0.9816, 0.99523)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, Precision)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(Precision, type = "o", col="orange", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:4,
     lab=c("Model_GLM1", "Model_GLM2", "SA Model", "HESCA Model"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=0.1*10:g_range[2])

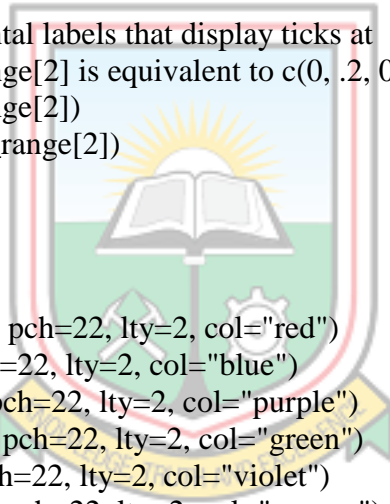
# Create box around plot
box()

lines(Accuracy, type = "o", pch=22, lty=2, col="red")
lines(AUC, type = "o", pch=22, lty=2, col="blue")
lines(Logloss, type = "o", pch=22, lty=2, col="purple")
lines(F1_score, type = "o", pch=22, lty=2, col="green")
lines(Recall, type = "o", pch=22, lty=2, col="violet")
lines(Precision, type = "o", pch=22, lty=2, col="orange")
lines(Specificity, type = "o", pch=22, lty=2, col="brown")

title(main = "Comparison of Stacked Ensemble Models and
HESCA Model Performance on Training Set", col.main="black", font.main=1)

title(xlab = "Stacked Models", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "Performance value", col.lab=rgb(0.2,0.2,0.3))
legend(3.3, g_range[2], c("Accuracy", "logloss", "Recall", "Specificity", "Precision",
"F1_score", "AUC"),
     cex = 0.7, col = c("red", "purple", "violet", "brown", "orange", "green", "blue"), pch = 21:22,
     lty = 1:2);
#####
# Define vectors
Accuracy <- c(0.53125, 0.90625, 0.84375, 0.90625)
Logloss <- c(0.78799, 0.42667, 0.4406, 0.29591)
Recall <- c(0.37500, 0.87500, 0.6667, 0.75000)
Specificity <- c(1.00000, 0.91667, 0.9500, 1.00000)

```



```

Precision <- c(1.00000, 0.77778, 0.8889, 1.00000)
F1_score <- c(0.54545, 0.82350, 0.76193, 0.85714)
AUC <- c(0.44928, 0.86232, 0.9179, 0.92512)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, Precision)

# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(Precision, type = "o", col="orange", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:4,
     lab=c("Model_GLM1", "Model_GLM2", "SA Model", "HESCA Model"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=0.1*10:g_range[2])

# Create box around plot
box()

lines(Accuracy, type = "o", pch=22, lty=2, col="red")
lines(AUC, type = "o", pch=22, lty=2, col="blue")
lines(Logloss, type = "o", pch=22, lty=2, col="purple")
lines(F1_score, type = "o", pch=22, lty=2, col="green")
lines(Recall, type = "o", pch=22, lty=2, col="violet")
lines(Precision, type = "o", pch=22, lty=2, col="orange")
lines(Specificity, type = "o", pch=22, lty=2, col="brown")

title(main = "Comparison of Stacked Ensemble Models and
HESCA Model Performance on Test Set", col.main="black", font.main=1)

title(xlab = "Stacked Models", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "Performance value", col.lab=rgb(0.2,0.2,0.3))
legend(3.3, g_range[2], c("Accuracy", "logloss", "Recall", "Specificity", "Precision",
"F1_score", "AUC"),
     cex = 0.7, col = c("red", "purple", "violet", "brown", "orange", "green", "blue"), pch = 21:22,
     lty = 1:2);
#####
Accuracy <- c(0.9677, 0.9140, 0.9032, 0.9032, 0.9032)
AUC <- c(0.9952, 0.9677, 0.9164, 0.9164, 0.9164)
Logloss <- c(0.1172, 0.2854, 0.2864, 0.2864, 0.2864)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, AUC)

# Graph autos using y axis that ranges from 0 to max

```

```

# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(AUC, type = "o", col="blue", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:5,
     lab=c("GBM-FS", "DRF-FS", "DNN-FS", "GLM-FS", "NB-FS"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=0.1*10:g_range[2])

# Create box around plot
box()

lines(Accuracy, type = "o", pch=22, lty=2, col="red")
lines(AUC, type = "o", pch=22, lty=2, col="blue")
lines(Logloss, type = "o", pch=22, lty=2, col="purple")

title(main = "Performance Metrics of Stacked Ensemble Model
on various Feature Subsets on Training Set", col.main="black", font.main=1)

title(xlab = "Feature Selectors", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "Performance value", col.lab=rgb(0.2,0.2,0.3))

legend(4.1, g_range[2], c("Accuracy", "AUC", "logloss"),
      cex = 0.7, col = c("red", "blue", "purple"), pch = 23:23, lty = 1:2);
#####
Accuracy <- c(0.9063, 0.7813, 0.7500, 0.7500, 0.7500)
AUC <- c(0.9251, 0.7536, 0.7319, 0.7319, 0.7319)
Logloss <- c(0.2959, 0.4046, 0.5246, 0.5246, 0.5246)

# Calculate range from 0 to max value of accuracy and auc
g_range <- range(0, AUC)
# Graph autos using y axis that ranges from 0 to max
# Value in accuracy or AUC vectors. Turn off axes and
# annotations (axis labels) so we can specify them ourself
plot(AUC, type = "o", col="blue", ylim = g_range, axes = FALSE, ann = FALSE)

# Make x axis using base selectors labels
axis(1, at=1:5,
     lab=c("GBM-FS", "DRF-FS", "DNN-FS", "GLM-FS", "NB-FS"))

# Make y axis with horizontal labels that display ticks at
# every 4 marks. 4*0:g_range[2] is equivalent to c(0, .2, 0.4, 0.6, 0.8, 1, 1.2)
axis(2, las=1, at=1*0:g_range[2])
axis(2, las=1, at=0.1*10:g_range[2])

```

```

# Create box around plot
box()

lines(Accuracy, type = "o", pch=22, lty=2, col="red")
lines(AUC, type = "o", pch=22, lty=2, col="blue")
lines(Logloss, type = "o", pch=22, lty=2, col="purple")

title(main = "Performance Metrics of Stacked Ensemble Model
on various Feature Subsets on Test Set", col.main="black", font.main=1)

title(xlab = "Feature Selectors", col.lab=rgb(0.2,0.2,0.3))
title(ylab = "Performance value", col.lab=rgb(0.2,0.2,0.3))

legend(4.1, g_range[2], c("Accuracy", "AUC", "logloss"),
      cex = 0.7, col = c("red", "blue", "purple"), pch = 23:23, lty = 1:2);
#####

```



APPENDIX F PDP AND ICE R CODES

```
library(caret)
library(gridExtra)
library(grid)
library(ggribes)
library(ggthemes)
library(iml)
library(partykit)
library(rpart)
library(tidyverse)
library(data.table)

theme_set(theme_minimal())
set.seed(88)

kfolds <- 5

load_data <- function() {
  dataset <- fread(file.choose(), data.table = T) %>%
    mutate(Recurrence=as.factor(ifelse(Recurrence== 1, "Yes", "No")))
  X = dataset[, 1:6]
  Y = dataset$Recurrence
  return(list(dataset, X, Y))
}

str(dataset)

compute_rf_model <- function(dataset) {
  index <- createDataPartition(dataset$Recurrence,
    p=0.8,
    list = FALSE,
    times = 1)
  dataset_train <- dataset[index,]
  dataset_test <- dataset[-index,]

  fit_control <- trainControl(method = "repeatedcv",
    number = kfolds,
    repeats = 1,
    classProbs = TRUE,
    savePredictions = TRUE,
    verboseIter = FALSE,
    allowParallel = FALSE,
    summaryFunction = defaultSummary)

  rf_model <- train(Recurrence~.,
    data=dataset_train,
    method="gbm",
    preProcess=c("center", "scale"),
    trControl=fit_control,
```



```

        metric="Accuracy",
        verbose=FALSE)
return(list(rf_model, dataset_train, dataset_test))
}

main <- function() {
  data <- load_data()
  dataset <- data[[1]]
  X <- data[[2]]
  Y <- data[[3]]

  rf_model_data <- compute_rf_model(dataset)
  rf_model <- rf_model_data[[1]]
  dataset_train <- rf_model_data[[2]]
  dataset_test <- rf_model_data[[3]]

  X <- dataset_train %>%
    select(-Recurrence) %>%
    as.data.frame()

  predictor <- Predictor$new(rf_model, data = X, Y = dataset_train$Recurrence,
    type = "prob")

  ice <- FeatureEffect$new(predictor, feature = "TreatCCRT",
    center.at = min(x$TreatCCRT), method = "pdp+ice")
  ice_plot_TreatCCRT <- ice$plot() +
    scale_color_discrete(guide = "none") +
    scale_y_continuous("Predicted Recurrence")

  ice <- FeatureEffect$new(predictor, feature = "PaTT4",
    center.at = min(x$PaTT4), method = "pdp+ice")
  ice_plot_paTT4 <- ice$plot() +
    scale_color_discrete(guide = "none") +
    scale_y_continuous("Predicted Recurrence")

  ice <- FeatureEffect$new(predictor, feature = "p63",
    center.at = min(x$p63), method = "pdp+ice")
  ice_plot_p63 <- ice$plot() +
    scale_color_discrete(guide = "none") +
    scale_y_continuous("Predicted Recurrence")

  ice <- FeatureEffect$new(predictor, feature = "Nodes",
    center.at = min(x$Nodes), method = "pdp+ice")
  ice_plot_nodes <- ice$plot() +
    scale_color_discrete(guide = "none") +
    scale_y_continuous("Predicted Recurrence")

  ice <- FeatureEffect$new(predictor, feature = "Smoke",
    center.at = min(x$Smoke), method = "pdp+ice")
  ice_plot_smoke <- ice$plot() +

```

```

scale_color_discrete(guide = "none") +
scale_y_continuous("Predicted Recurrence")

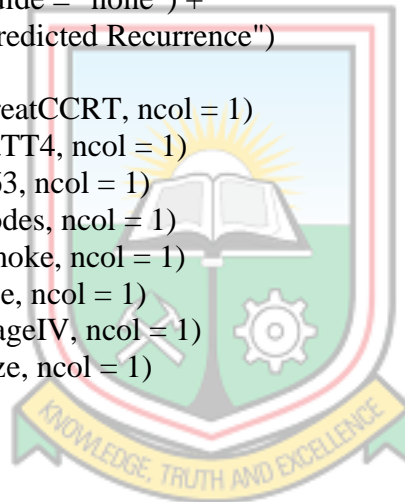
ice <- FeatureEffect$new(predictor, feature = "Age",
                        center.at = min(x$Age), method = "pdp+ice")
ice_plot_age <- ice$plot() +
  scale_color_discrete(guide = "none") +
  scale_y_continuous("Predicted Recurrence")

ice <- FeatureEffect$new(predictor, feature = "StageIV",
                        center.at = min(x$StageIV), method = "pdp+ice")
ice_plot_stageIV <- ice$plot() +
  scale_color_discrete(guide = "none") +
  scale_y_continuous("Predicted Recurrence")

ice <- FeatureEffect$new(predictor, feature = "Size",
                        center.at = min(x$Size), method = "pdp+ice")
ice_plot_size <- ice$plot() +
  scale_color_discrete(guide = "none") +
  scale_y_continuous("Predicted Recurrence")

grid.arrange(ice_plot_TreatCCRT, ncol = 1)
grid.arrange(ice_plot_paTT4, ncol = 1)
grid.arrange(ice_plot_p63, ncol = 1)
grid.arrange(ice_plot_nodes, ncol = 1)
grid.arrange(ice_plot_smoke, ncol = 1)
grid.arrange(ice_plot_age, ncol = 1)
grid.arrange(ice_plot_stageIV, ncol = 1)
grid.arrange(ice_plot_size, ncol = 1)
}

```



APPENDIX G HNSCC PROGNOSIS DATASET

S/N	Gender	Age	Drink	Smoke	Chew	Site	Stage	Grade	Size	Invasion
1	F	44	No	No	Yes	NPC	III	G3	10	Cohesive
2	F	56	No	No	No	NPC	III	G3	5.5	on cohesiv
3	M	52	No	No	Yes	NPC	III	G2	5.8	Cohesive
4	F	45	Yes	No	Yes	NPC	II	G2	1	on cohesiv
5	F	28	No	No	Yes	NPC	III	G2	9.4	on cohesiv
6	M	55	No	No	No	OPC	IV	G2	10.5	Cohesive
7	F	45	No	No	No	NPC	IV	G3	1.2	Cohesive
8	M	37	No	No	No	NPC	III	G3	1	on cohesiv
9	M	15	No	No	No	NPC	III	G3	2	on cohesiv
10	M	29	No	No	No	NPC	IV	G2	0.5	on cohesiv
11	M	72	No	No	No	NPC	IV	G3	1	Cohesive
12	M	24	Yes	Yes	No	NPC	III	G3	7.7	Cohesive
13	M	47	No	No	No	NPC	III	G2	1.5	Cohesive
14	M	56	Yes	Yes	No	NPC	IV	G2	3	Cohesive
15	M	47	No	No	No	OPC	III	G3	1.3	on cohesiv
16	F	55	No	No	No	Larynx	II	G1	1.5	Cohesive
17	M	28	No	No	No	NPC	II	G2	10	Cohesive
18	M	64	No	Yes	No	Larynx	IV	G3	2	on cohesiv
19	M	68	No	No	No	NPC	II	G3	4	on cohesiv
20	M	56	Yes	No	No	OPC	IV	G3	1.5	on cohesiv
21	M	59	Yes	Yes	No	OPC	III	G2	1	on cohesiv
22	F	63	No	No	Yes	NPC	III	G3	0.9	on cohesiv
23	M	51	No	No	No	NPC	I	G3	2	on cohesiv
24	M	16	No	No	No	NPC	III	G3	2	on cohesiv
25	F	27	No	No	No	NPC	IV	G3	2	on cohesiv
26	M	57	Yes	Yes	No	NPC	IV	G2	8	Cohesive
27	M	59	Yes	Yes	No	NPC	I	G2	4.4	Cohesive
28	M	45	No	No	No	NPC	I	G3	1.2	Cohesive
29	M	20	No	No	NA	NPC	IV	G3	1.8	on cohesiv
30	M	36	Yes	No	No	NPC	IV	G2	10	Cohesive
31	F	23	No	No	NA	NPC	III	G3	1.5	on cohesiv
32	F	39	No	No	No	OPC	IV	G3	1.3	on cohesiv
33	M	60	Yes	Yes	No	Larynx	II	G2	13	Cohesive
34	M	52	Yes	Yes	No	NPC	III	G2	1.3	on cohesiv
35	F	60	Yes	Yes	No	NPC	IV	G2	1	on cohesiv
36	M	37	Yes	Yes	No	HPC	IV	G3	10	Cohesive
37	M	35	Yes	No	No	NPC	IV	G3	NA	NA
38	F	18	No	No	No	NPC	II	G3	0.8	on cohesiv
39	M	18	No	No	No	NPC	II	G3	5.7	Cohesive
40	F	44	Yes	No	No	NPC	III	G3	1	on cohesiv
41	M	52	Yes	Yes	No	NPC	IV	G2	1.5	on cohesiv
42	F	58	Yes	No	No	OPC	III	G1	1.2	on cohesiv
43	M	71	Yes	Yes	No	NPC	III	G2	1	on cohesiv
44	M	65	Yes	Yes	No	NPC	III	G3	1	on cohesiv
45	F	48	No	No	No	NPC	IV	G3	7	Cohesive
46	F	56	No	No	No	NPC	III	G3	1.5	on cohesiv
47	F	17	No	No	No	NPC	III	G3	0.7	on cohesiv
48	M	62	Yes	Yes	No	Larynx	IV	G1	NA	NA
49	F	39	No	No	No	Larynx	III	G2	8.5	Cohesive
50	M	60	No	No	No	Larynx	IV	G1	6	Cohesive
51	M	59	Yes	No	No	Larynx	I	G3	0.7	on cohesiv
52	M	49	No	Yes	No	Larynx	IV	G1	2.6	on cohesiv
53	M	48	Yes	Yes	Yes	OPC	IV	G3	5.5	Cohesive
54	F	40	No	No	NA	NPC	III	G3	1	on cohesiv
55	M	73	No	No	Yes	Larynx	IV	G1	4	on cohesiv
56	F	38	No	No	NA	Larynx	IV	G3	10	Cohesive
57	M	38	Yes	No	No	OPC	III	G3	5	Cohesive
58	F	58	No	No	Yes	OPC	IV	G3	1.8	on cohesiv
59	M	49	No	No	Yes	Larynx	III	G1	11	Cohesive
60	M	74	Yes	No	Yes	Larynx	IV	G3	NA	NA
61	F	55	No	No	Yes	NPC	II	G3	5	Cohesive

S/N	Gender	Age	Drink	Smoke	Chew	Site	Stage	Grade	Size	Invasion
62	M	57	Yes	Yes	No	Larynx	II	G3	4	on cohesiv
63	M	20	No	No	No	NPC	IV	G3	1	on cohesiv
64	M	14	No	No	No	NPC	III	G3	NA	NA
65	M	13	No	No	No	NPC	III	G3	3	on cohesiv
66	M	66	Yes	Yes	No	HPC	II	G3	7.3	Cohesive
67	M	13	No	No	No	NPC	IV	G2	2	Cohesive
68	F	61	Yes	Yes	No	Larynx	IV	G1	3	on cohesiv
69	M	36	No	No	No	NPC	IV	G3	2	on cohesiv
70	F	19	No	No	No	NPC	III	G3	0.6	on cohesiv
71	M	54	Yes	Yes	No	NPC	IV	G3	2.3	on cohesiv
72	M	24	No	No	No	NPC	III	G3	1.2	on cohesiv
73	M	19	No	No	No	NPC	IV	G3	2.5	on cohesiv
74	M	73	Yes	Yes	No	Larynx	IV	G3	10	Cohesive
75	M	54	No	Yes	No	NPC	IV	G3	1.5	on cohesiv
76	M	72	No	No	No	Larynx	II	G1	3	on cohesiv
77	M	50	Yes	No	No	Larynx	II	G1	4.5	Cohesive
78	M	21	Yes	No	No	OPC	II	G3	2.4	on cohesiv
79	M	32	No	No	No	Larynx	IV	G3	2.5	on cohesiv
80	M	21	No	No	No	NPC	IV	G3	7.6	Cohesive
81	M	53	No	No	No	Larynx	IV	G3	4	on cohesiv
82	F	76	Yes	Yes	No	Larynx	II	G3	3.5	on cohesiv
83	M	17	No	No	No	NPC	II	G3	0.9	on cohesiv
84	M	53	Yes	Yes	No	Larynx	I	G1	1.7	on cohesiv
85	M	76	Yes	Yes	No	Larynx	IV	G1	1	on cohesiv
86	M	44	No	No	No	NPC	IV	G3	4	on cohesiv
87	M	64	Yes	Yes	No	Larynx	IV	G3	6.7	Cohesive
88	F	22	No	No	No	NPC	III	G3	1.2	on cohesiv
89	M	52	No	No	No	Larynx	IV	G3	9	Cohesive
90	M	18	No	No	No	NPC	II	G2	5	Cohesive
91	M	18	No	No	No	NPC	II	G3	1.2	on cohesiv
92	F	58	No	No	No	Larynx	II	G2	1	on cohesiv
93	M	62	Yes	Yes	No	Larynx	IV	G3	5	Cohesive
94	M	81	No	Yes	NA	Larynx	I	G3	1.2	on cohesiv
95	M	36	No	No	Yes	NPC	III	G3	3.5	on cohesiv
96	M	51	Yes	No	No	Larynx	IV	G3	1	on cohesiv
97	M	83	No	No	Yes	Larynx	III	G2	5	Cohesive
98	M	70	No	No	No	Larynx	IV	G2	3	on cohesiv
99	M	66	Yes	Yes	Yes	Larynx	III	G2	5	Cohesive
100	F	44	No	No	Yes	NPC	IV	G3	10	Cohesive
101	M	32	Yes	No	NA	NPC	IV	G3	6	Cohesive
102	M	56	Yes	Yes	Yes	NPC	II	G2	1	on cohesiv
103	F	32	No	No	No	NPC	IV	G2	16	Cohesive
104	M	61	No	No	Yes	Larynx	II	G1	4	on cohesiv
105	M	42	Yes	Yes	Yes	NPC	III	G2	4	Cohesive
106	M	57	Yes	Yes	No	OPC	IV	G2	0.5	on cohesiv
107	M	67	No	No	No	Larynx	IV	G1	1.4	on cohesiv
108	M	67	No	Yes	No	Larynx	II	G2	2	on cohesiv
109	M	59	Yes	Yes	No	NPC	IV	G3	0.7	on cohesiv
110	M	78	Yes	Yes	Yes	Larynx	II	G3	0.63	on cohesiv
111	M	40	Yes	Yes	Yes	HPC	IV	G1	5.6	Cohesive
112	M	64	Yes	No	No	Larynx	IV	G2	10	Cohesive
113	M	48	No	No	NA	Larynx	IV	G3	10	Cohesive
114	M	63	No	Yes	No	Larynx	IV	G3	11	Cohesive
115	M	59	Yes	No	No	Larynx	IV	G1	3	Cohesive
116	F	63	No	No	Yes	OPC	IV	G3	1	on cohesiv
117	M	86	Yes	Yes	No	NPC	II	G1	8.2	Cohesive
118	M	52	Yes	No	Yes	Larynx	IV	G2	0.1	on cohesiv
119	M	44	No	No	No	NPC	IV	G3	1	on cohesiv
120	M	15	No	No	NA	NPC	IV	G3	2.4	Cohesive
121	F	25	No	No	Yes	NPC	IV	G3	1	on cohesiv
122	M	26	No	No	No	NPC	I	G3	2	on cohesiv
123	M	24	No	No	No	NPC	IV	G1	2.5	on cohesiv
124	F	11	No	No	Yes	NPC	IV	G3	1	on cohesiv
125	F	56	No	No	No	NPC	IV	G3	1	on cohesiv

S/N	Nodes	PaT	PIN	History	HPV	p16	p63	Treat	Recurrence
1	pos	T4	N3	No	HPV16	pos	pos	Chemo	No
2	pos	T1	N2	No	HPV16	pos	pos	CRT	Yes
3	pos	T4	N1	Yes	HPV16	pos	pos	RT	No
4	neg	T2	N2	No	NA	neg	pos	CRT	No
5	pos	T2	N3	No	NA	NA	neg	CRT	No
6	pos	T4	N3	No	NA	pos	pos	Chemo	No
7	pos	T3	N2	No	HPV16	pos	neg	RT	Yes
8	pos	T3	N2	No	NA	pos	pos	CRT	Yes
9	NA	T4	N3	No	HPV11	pos	pos	CRT	Yes
10	neg	T4	N0	No	NA	pos	neg	CRT	Yes
11	pos	T4	N2	No	HPV11	pos	pos	CRT	Yes
12	NA	T4	N3	No	HPV6	pos	pos	CCRT	No
13	pos	T3	N3	No	NA	pos	pos	CCRT	No
14	pos	T4	N1	NA	NA	pos	neg	CCRT	No
15	pos	T3	N1	Yes	HPV16	pos	neg	CRT	Yes
16	neg	T2	N2	No	HPV11	pos	pos	CRT	No
17	pos	T2	N2	Yes	HPV18	pos	pos	CCRT	No
18	NA	T4	N0	NA	HPV6	pos	neg	CRT	No
19	pos	T2	N2	No	HPV11	pos	neg	CCRT	Yes
20	pos	T4	N0	No	HPV18	neg	pos	CRT	Yes
21	pos	T4	N2	No	NA	pos	pos	Chemo	Yes
22	pos	T4	N2	No	HPV16	pos	neg	CRT	Yes
23	neg	T2	N1	No	HPV16	pos	pos	CCRT	Yes
24	pos	T3	N3	No	NA	NA	neg	CCRT	No
25	NA	T4	N3	No	NA	neg	pos	CRT	Yes
26	pos	T2	N3	No	NA	neg	pos	CRT	Yes
27	neg	T2	N0	No	HPV18	NA	neg	CRT	Yes
28	neg	T3	N0	No	NA	pos	pos	CRT	No
29	neg	T4	N2	No	HPV18	NA	neg	CCRT	No
30	pos	T4	N2	No	NA	NA	NA	Surg+RT	No
31	pos	T3	N1	No	NA	pos	pos	CCRT	No
32	NA	T4	N2	No	NA	pos	pos	RT	Yes
33	neg	T3	N2	No	NA	pos	pos	CRT	No
34	pos	T3	N3	No	NA	pos	pos	CCRT	No
35	pos	T4	N2	No	HPV18	pos	neg	Chemo	NA
36	pos	T4	N0	No	NA	pos	neg	Surg+CCRT	No
37	pos	T4	N2	No	NA	pos	pos	CCRT	No
38	pos	T3	N2	No	HPV11	pos	neg	CCRT	No
39	neg	T3	N2	No	NA	NA	pos	CCRT	No
40	pos	T1	N3	No	HPV18	pos	pos	Chemo	Yes
41	NA	T4	N2	Yes	NA	neg	NA	CRT	Yes
42	NA	T4	N2	No	NA	neg	pos	CRT	No
43	NA	T4	N1	NA	HPV16	neg	pos	RT	Yes
44	NA	T4	N0	No	NA	NA	neg	CCRT	Yes
45	neg	T4	N0	NA	NA	neg	neg	RT	Yes
46	pos	T4	N3	No	HPV18	NA	neg	RT	Yes
47	pos	T4	N2	No	NA	pos	neg	CRT	No
48	pos	T4	N2	No	NA	pos	pos	CRT	Yes
49	neg	T4	N0	No	HPV11	pos	pos	CCRT	No
50	pos	T4	N2	NA	NA	pos	neg	CRT	Yes
51	neg	T2	N0	No	NA	pos	neg	RT	No
52	neg	T4	N0	NA	NA	pos	pos	Surg+CCRT	Yes
53	pos	T4	N2	NA	HPV16	pos	pos	CRT	Yes
54	pos	T4	N3	Yes	NA	pos	pos	Chemo	Yes
55	pos	T4	N0	No	NA	pos	neg	RT	Yes
56	neg	T4	N0	No	NA	pos	neg	RT	No
57	pos	T3	N3	NA	HPV16	pos	pos	CCRT	Yes
58	pos	T3	N3	NA	NA	pos	neg	CRT	No
59	pos	T4	N2	NA	NA	pos	pos	RT	NA
60	pos	T4	N3	No	NA	pos	pos	RT	Yes
61	pos	T3	N0	Yes	HPV16	pos	NA	Surg+CCRT	Yes

S/N	Nodes	PaT	PIN	History	HPV	p16	p63	Treat	Recurrence
62	NA	T3	N0	Yes	NA	pos	neg	Surg+RT	No
63	pos	T4	N3	Yes	NA	neg	neg	CRT	Yes
64	neg	T4	N2	Yes	HPV16	neg	pos	CCRT	No
65	neg	T4	N2	Yes	NA	neg	pos	CCRT	No
66	neg	T4	N0	Yes	NA	neg	pos	CCRT	No
67	pos	T4	N3	No	HPV16	neg	neg	CCRT	Yes
68	neg	T4	N0	Yes	NA	neg	pos	RT	No
69	NA	T4	N3	NA	NA	pos	neg	CCRT	No
70	pos	T3	N2	NA	HPV16	neg	neg	CCRT	No
71	pos	T3	N3	Yes	NA	neg	pos	CRT	No
72	NA	T4	N2	No	NA	pos	neg	CCRT	Yes
73	pos	T3	N3	No	NA	neg	pos	CRT	No
74	neg	T4	N0	No	HPV11	neg	pos	RT	Yes
75	neg	T4	N3	No	NA	pos	neg	RT	Yes
76	neg	T3	N0	No	NA	pos	neg	RT	Yes
77	pos	T3	N0	No	NA	pos	pos	RT	Yes
78	pos	T4	N2	No	HPV16	pos	neg	CRT	Yes
79	neg	T4	N0	No	NA	pos	neg	CRT	Yes
80	pos	T4	N2	NA	NA	pos	pos	CCRT	Yes
81	neg	T4	N1	No	NA	pos	pos	CCRT	No
82	neg	T2	N2	No	NA	NA	neg	RT	No
83	neg	T1	N0	NA	HPV16	NA	neg	RT	Yes
84	pos	T4	N3	NA	NA	NA	neg	Surg+CCRT	Yes
85	NA	T4	N0	NA	NA	neg	pos	Chemo	NA
86	pos	T4	N2	No	NA	pos	neg	RT	No
87	neg	T4	N0	No	HPV16	neg	pos	RT	Yes
88	pos	T3	N2	No	NA	pos	neg	CCRT	No
89	NA	T4	N0	No	HPV18	neg	pos	CRT	Yes
90	pos	T4	N3	No	NA	pos	neg	CRT	No
91	neg	T3	N3	No	NA	neg	pos	CRT	Yes
92	neg	T4	N2	No	HPV16	pos	neg	RT	No
93	neg	T4	N0	No	NA	neg	NA	RT	No
94	neg	T1	N0	No	HPV11	NA	neg	RT	No
95	pos	T4	N2	No	HPV16	pos	NA	RT	Yes
96	pos	T4	N0	No	NA	NA	pos	RT	Yes
97	neg	T3	N0	No	HPV11	pos	neg	RT	No
98	neg	T3	N2	NA	NA	pos	neg	Surg+RT	Yes
99	pos	T4	N2	NA	NA	neg	NA	CRT	No
100	neg	T4	N3	No	HPV16	pos	pos	CRT	No
101	neg	T4	N0	NA	NA	neg	NA	CRT	Yes
102	pos	T2	N1	NA	NA	neg	pos	CRT	No
103	pos	T4	N3	No	HPV11	pos	pos	Chemo	Yes
104	pos	T3	N1	No	HPV18	neg	neg	Surg+RT	Yes
105	pos	T2	N1	No	HPV16	neg	pos	Surg+CRT	Yes
106	NA	T1	N2	No	NA	pos	neg	CRT	No
107	neg	T4	N1	No	NA	neg	neg	RT	No
108	neg	T3	N0	No	HPV18	pos	pos	RT	Yes
109	NA	T4	N2	NA	HPV11	neg	neg	RT	No
110	NA	T3	N1	Yes	NA	pos	neg	CRT	No
111	NA	T4	N3	NA	HPV16	neg	pos	CRT	No
112	neg	T2	N0	No	NA	pos	neg	CRT	No
113	neg	T4	N0	NA	HPV11	pos	neg	RT	No
114	neg	T4	N2	No	NA	NA	neg	Surg+CRT	Yes
115	neg	T4	N2	No	HPV16	pos	NA	CRT	No
116	neg	T4	N2	No	NA	pos	neg	Chemo	Yes
117	neg	T4	N1	Yes	HPV16	neg	pos	RT	Yes
118	pos	T4	N1	NA	NA	pos	pos	RT	Yes
119	NA	T4	N3	Yes	HPV11	neg	neg	CRT	No
120	pos	T3	N3	No	NA	neg	pos	CRT	No
121	neg	T4	N3	NA	HPV16	pos	neg	RT	Yes
122	neg	T2	N0	No	HPV18	neg	neg	CRT	Yes
123	pos	T4	N3	No	HPV16	neg	pos	CRT	NA
124	NA	T4	N3	No	NA	pos	NA	CRT	No
125	pos	T4	N3	No	HPV16	NA	neg	CRT	Yes

APPENDIX H MEAN ACCURACY FOR BASE MODELS SELECTION

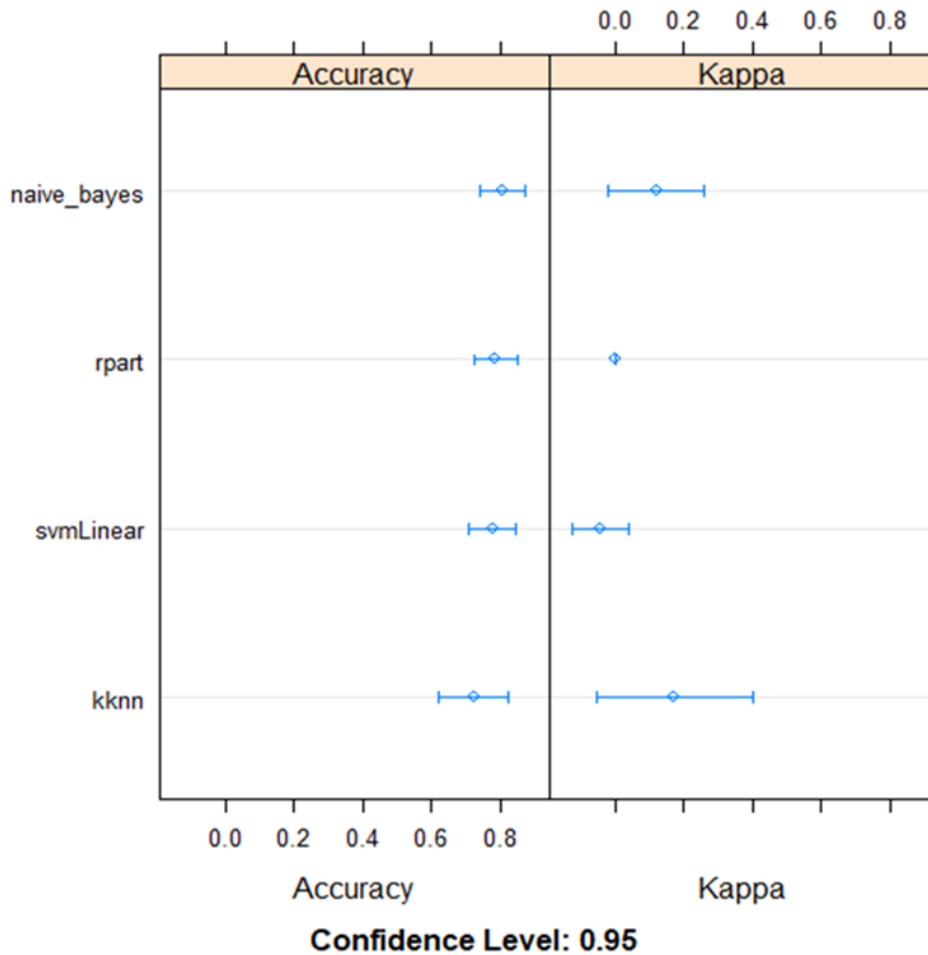
Models: kkn, svmLinear, rpart, naive_bayes
 Number of resamples: 50

Accuracy

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
kkn	0.0	0.5	1.0000000	0.7233333	1	1	0
svmLinear	0.0	0.5	0.8333333	0.7766667	1	1	0
rpart	0.5	0.5	0.8333333	0.7866667	1	1	0
naive_bayes	0.5	0.5	1.0000000	0.8066667	1	1	0

Kappa

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
kkn	-1	0	0	0.1741935	1	1	19
svmLinear	-1	0	0	-0.0400000	0	0	25
rpart	0	0	0	0.0000000	0	0	25
naive_bayes	0	0	0	0.1200000	0	1	25



APPENDIX I FRAMEWORK OF SAMPLE SIZE DETERMINATION

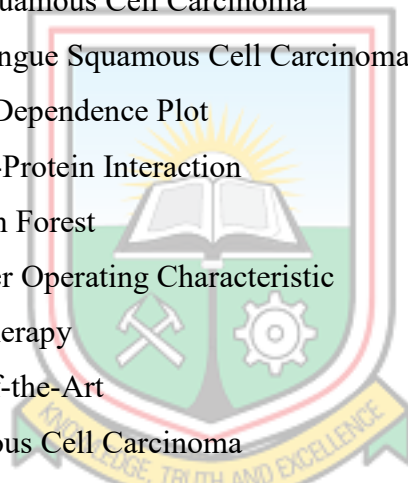
		$=(D2*(D3^2)*D5*(1-D5)/(D4^2)/(D2-1+((D3^2)*D5*(1-D5)/(D4^2))))$									
	A	B	C	D	E	F	G	H	I	J	
1	Particular			Value							
2	Population Size (N)			185							
3	z			1.96	125.087						
4	ϵ			0.05							
5	p (uncertain)			0.5							
6	q			0.5							
7											
8	Sample Size (n)				125						



APPENDIX J LIST OF ACRONYMS AND ABBREVIATIONS

Acronym	Meaning
10-CV	10-fold Cross-Validation
AdaBoost	Adaptive Boosting
AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC	Area Under the Curve
AU-ROC	Area Under Receiver Operating Characteristic
BNN	Bayesian Neural Network
BT	Bagged Tree
CCRT	Concurrent Chemo Radiotherapy
Chemo	Chemotherapy
CRT	Chemo Radiotherapy
CV	Cross Validation
DBN	Dynamic Bayesian Network
DNA	Deoxyribonucleic Acid
DNN	Deep Neural Network
DNN-FS	Deep Neural Network Feature Selection
DRF	Distributed Random Forest
DRF-FS	Distributed Random Forest Feature Selection
DSS	Decision Support System
DT	Decision Tree
EGFR	Epidermal Growth Factor Receptor
EFS	Ensemble Feature Selection
FN	False Negative
FP	False Positive
FS	Feature Selection
GBM	Gradient Boosting Machine
GBM-FS	Gradient Boosting Machine Feature Selection
GLM	Generalised Linear Model
GLM-FS	Generalised Linear Model Feature Selection
HESCA	Hybrid Ensemble Super Classification Algorithm
HNC	Head and Neck Cancer

HNSCC	Head and Neck Squamous Cell Carcinoma
HPV	Human Papilloma Virus
ICE	Individual Conditional Expectations
KATH	Komfo Anokye Teaching Hospital
KBTH	Korle-Bu Teaching Hospital
KNNs	K-Nearest Neighbours
LC	Laryngeal Carcinoma
LOGLOSS	Logarithmic Loss
LR	Logistic Regression
ML	Machine Learning
NB	Naïve Bayes
NB-FS	Naïve Bayes Feature Selection
OSCC	Oral Squamous Cell Carcinoma
OTSCC	Oral Tongue Squamous Cell Carcinoma
PDP	Partial Dependence Plot
PPI	Protein-Protein Interaction
RF	Random Forest
ROC	Receiver Operating Characteristic
RT	Radiotherapy
SA	State-of-the-Art
SCC	Squamous Cell Carcinoma
SCCHN	Squamous Cell Carcinoma of Head and Neck
SVM	Support Vector Machine
TCGA	The Cancer Genome Atlas
TCIA	The Cancer Imaging Archive
TNM	Tumor, Node, Metastasis
TN	True Negative
TP	True Positive
TreatCCRT	Treatment with Concurrent Chemotherapy
WHO	World Health Organisation
UCC	University of Cape Coast
UMaT	University of Mines and Technology
VIF	Variance Inflation Factor



WFI Weighted Feature Importance
XGBoost Extreme Gradient Boosting



APPENDIX K LIST OF PUBLICATIONS

PAPER 1



International Journal of
Applied Science Research (IJASR)
Research Article | Volume 1, Issue 2 | Pages 56-71

A Hybrid Ensemble Feature Selection Technique for Recurrent Head and Neck Squamous Cell Carcinoma Prognosis

Damianus Kofi Owusu¹, Christiana Cynthia Nyarko², Joseph Acquah³ and Joel Yarney⁴

^{1,2,3}Department of Mathematical Sciences, Faculty of Engineering, University of Mines and Technology, Tarkwa, Ghana, dkowusu@st.umat.edu.gh¹, ccnyarko@umat.edu.gh², jacquah@umat.edu.gh³

⁴Department of Radiotherapy and Oncology, National centre for radiotherapy & nuclear medicine, Korle-Bu, Accra, Ghana, kodwoahen@gmail.com

*Correspondence: Damianus Kofi Owusu; Email: dkowusu@st.umat.edu.gh

ABSTRACT- With the rapid development of Head and Neck Squamous Cell Carcinomas (HNSCC) cases with their recurrences presents medical challenges to all involved in their management worldwide, particularly in developing countries like Ghana; where the recurrent and mortality rates are very high, due to poor prognosis. Several attempts to identifying the most accurate set of prognoses associated with HNSCC recurrence proved futile because high dimensional cancer datasets usually contain redundant and or irrelevant features. This way, feature selection is particularly important. In that, the training features used to learn a classification model have a huge influence on the performance of the model. Because using a single feature selection technique to obtain feature subset may be biased, unstable, or unreliable. An ensemble feature selection technique dubbed Subsets Summation Frequency Ensemble Feature Selection (SSF-EFS) based on subsets summation frequency was proposed in this study, and this technique is oriented to classification task. For most accurate prognosis for recurrent HNSCC dataset, the results (feature subsets) of five single feature selection techniques; GBM, RF, DNN, NB, and GLM were aggregated by subsets summation specific technique. The integration effect of a given threshold value induced the acquisition of optimal feature subset. In order to evaluate the robustness of the prognostic performance of feature subsets, three most effective classifiers; GBM, RF, and NB with excellent performance were tested. The experimental results showed that the proposed ensemble feature selection technique effectively improved the classification using accuracy and AUC compared to single feature selection techniques used in this study.

Keywords: HNSCCs, Recurrence, Machine learning, Ensemble learning, Feature selection, Prognosis.

ARTICLE INFORMATION

Author(s): Damianus Kofi Owusu, Christiana Cynthia Nyarko, Joseph Acquah and Joel Yarney

Received: 22/05/2022; **Accepted:** 09/08/2022; **Published:** 10/09/2022;

ISSN: 2788-2788

larynx and oral cavity formed the largest group of HNSCCs, and most patients present with late-stage disease [33] [27]. It has also been observed that HNC in general is the third most common malignant tumor reported at the Centre for National Radiotherapy and Nuclear Medicine (CNRNM) at Korle-Bu

PAPER 2



Article Acceptance Certificate

This certificate confirms that the following paper has been accepted for publication in
Journal of AI and Data Mining

Title: Application of Stacked Ensemble Techniques in Head and Neck Squamous Cell Carcinoma
Prognostic Feature Subsets
ID: JADM-2211-2388 (R1)

Authors: DAMIANUS KOFI OWUSU, Christiana Cynthia Nyarko, Joseph Acquah, Joel Yarney

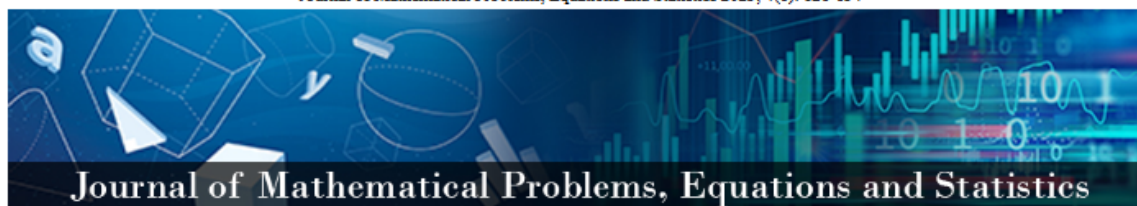
Submit Date: 20 November 2022

Accept Date: 10 July 2023

Hamid Hassanpour

Editor-in-Chief of Journal of AI and Data Mining





E-ISSN: 2709-9407
 P-ISSN: 2709-9393
 JMPES 2023; 4(1): 121-134
 © 2023 JMPES
www.mathematicaljournal.com
 Received: 03-04-2023
 Accepted: 04-05-2023

Damianus Kofi Owusu
 Ph.D., Department of
 Mathematical Sciences, Faculty
 of Engineering, University of
 Mines and Technology, Tarkwa,
 Ghana

Peter Kwesi Nyarko
 Ph.D., Department of
 Mathematical Sciences, Faculty
 of Engineering, University of
 Mines and Technology, Tarkwa,
 Ghana

Stacked ensemble model for recurrent head and neck squamous cell carcinoma prognosis based on clinicopathologic and genomic markers

Damianus Kofi Owusu and Peter Kwesi Nyarko

Abstract

The prevalence of head and neck squamous cell carcinoma (HNSCC) and its recurrences is not declining in Ghana as a result of the disease's delayed diagnosis and dismal prognosis. Early detection and treatment are crucial since HNSCC recurrence and tumor stage at diagnosis are significantly correlated. This study looked at the best meta-classifier model where the same ML classifiers for base classifiers and meta classifiers are employed in order to determine the most reliable prediction and robust prognostic model for recurrent HNSCC. Based on gradient boosted features (GBF), the suggested model was an ensemble of ML models that were stacked. Each of these models served as a meta-classifier and as a building block for the base classifiers. To find the optimal meta-classifier model, the performances of different meta-models were compared. The findings demonstrated that utilising the GBM as a meta-classifier produced superior accuracy with the least log loss compared to that produced by any other model of recurrent HNSCC prognostic data. This gave a stacked ensemble model termed as a HESCA model, consisted of five base models and GBM meta-model. 8-input HESCA model was compared with full-input model, and 8-input HESCA model was also compared with 8-input models. The results of the study demonstrated that using a GBM classifier as a meta-classifier in a stacking ensemble with five base classifiers based on GBF or GBM input features outperformed standalone models and any full-input model. Additionally, using a GBM as a meta-classifier is appropriate as a supporting tool for identifying, classifying, and predicting recurrent HNSCC prognosis data.

Keywords: Recurrent HNSCC prognosis, ensemble learning, stacked ensemble, classification



APPENDIX L ETHICAL ISSUES

In case of reply the number
And the date of this
Letter should be quoted

My Ref. No. KBTH/MD/K3/21
Your Ref. No.....



KORLE BU TEACHING HOSPITAL
P. O. BOX KB 77,
KORLE BU, ACCRA.

Tel: +233 302 667759/673034-6
Fax: +233 302 667759
Email: Info@kbth.gov.gh
pr@kbth.gov.gh
Website: www.kbth.gov.gh

29th March, 2021

DAMIANUS KOFI OWUSU
DEPARTMENT OF MATHEMATICAL SCIENCES
FACULTY OF ENGINEERING
UNIVERSITY OF MINES AND TECHNOLOGY

A MACHINE LEARNING TECHNIQUE IN MODELLING RECURRENT HEAD AND NECK SQUAMOUS CELL CARCINOMA PROGNOSIS IN GHANA

KBTH-IRB /000114/2020

Investigator: Damianus Kofi Owusu

The Korle Bu Teaching Hospital Institutional Review Board (KBTH IRB) reviewed and granted approval to the study entitled: "A Machine Learning Technique in Modelling Recurrent Head and Neck Squamous Cell Carcinoma Prognosis in Ghana"

Please note that the Board requires you to submit a final review report on completion of this study to the KBTH-IRB.

Kindly, note that, any modification/amendment to the approved study protocol without approval from KBTH-IRB renders this certificate invalid.

Please report all serious adverse events related to this study to KBTH-IRB within seven days verbally and fourteen days in writing.

This IRB approval is valid till 29th February, 2022. You are to submit annual report for continuing review.

Sincere regards,

DR. DANIEL ANKRAH
VICE CHAIR (KBTH-IRB)
FOR: CHAIR (KBTH-IRB)

Cc: The Chief Executive Officer, KBTH
The Director of Medical Affairs, KBTH

APPENDIX M INDEX

A

accuracy, 23, 25–26, 60, 81–82, 86, 88, 90–91, 93–94, 96–97, 99–108, 110–11, 138–48, 162–68, 170

best training, 86 model's, 64 prediction, 37, 125 robust prognostic, 10

accuracy value, 108–9 highest, 88, 90, 104 accurate number of prognostic features, 36 accurate prognosis, 4–6, 18, 23, 25, 120, 122–24 activation functions, 54–56 Akinbohun, 4–5, 25, 27, 31, 126 alcohol, 17, 80

algorithm for V-fold cross-validation, 58 algorithm for V-fold cross-validation in

Figure, 58

algorithms, 1, 8, 23, 28, 31, 43, 45–47, 57–58, 66, 72, 130, 154–55, 158–62

Anon, 2, 10–11, 17–19, 21–22, 126–27

Architecture of HESCA Model, 78

Architecture of HESCA Model for

Recurrent HNSCC Prognosis, 74 Area

Under the Curve. See AUC

assignment, 76, 153–54, 158, 160–61

AUC (Area Under the Curve), 26, 28–29,

61, 81, 86, 88, 90–91, 93–94, 96–97, 99–111, 138–48, 162–68

AUC on training and test data, 86 automatic model complexity optimisation

technique, popular, 58

B

bagging, 4, 43–45

base, 11, 43, 45, 85, 106, 119, 154–55, 158–60

base classifier models, 74–75, 124

base classifiers, 4–5, 7, 25–26, 44–45, 66, 68–71, 77, 85, 88–91, 93–96, 98–102, 104–5, 109–10, 120, 123–25

Base Classifiers on Training Set, 88–89 base learners, 4, 26, 42–45, 66, 155–56

base learning algorithms, 44, 69 baseline, 5, 20, 42, 79, 98–100, 105, 109,

111, 120–21, 123–24, 155–57 baseline ensemble models, 26

Baseline Stacked Ensemble Classification Techniques, 98

Baseline Stacked Ensemble Models, 108, 111

base models, 66, 105–8, 163–64 five, 66, 79

best accuracy, 91, 104, 106, 109–11 best feature selection model, 77

best performance accuracy in recurrent HNSCC prognosis, 122

boosting, 4, 43–44 Breast, 14–15

breast cancer, 5, 26–27 recurrent, 26

C

calendar period, 33–34, 71

cancer, 1–2, 4, 8–13, 16–22, 27–29, 32, 52, 68, 126–30, 133, 136

human, 17–18

laryngeal, 11–12, 17, 21, 135 oral, 21, 30, 130, 133 original, 10, 22

pharyngeal, 11–12, 17 recurrent, 22

cancer cases, 3, 10, 12, 17, 32 Cancer Cases Recorded, 15 cancer cells, 21–22

The Cancer Genome Atlas (TCGA), 24, 179 The Cancer Imaging Archive

(TCIA), 24,

179

cancer management, 7–8, 19 cancerous, 2, 18

Cancer Prediction and Prognosis, 9, 129 cancer prognosis, modelling, 1

cancer prognosis and prediction, 9, 132
 cancer recurrence, 2, 9–10, 21–22, 34
 laryngeal, 135
 cancer studies, 4, 19, 23, 25, 31, 63 cancer
 susceptibility, 9
 cancer susceptibility prediction, 2, 9
 cancer types, 2, 9, 13, 22, 25, 27
 Categories, 39, 42, 48, 63, 83, 138–48
 categorising breast cancer data, 26
 CCRT (Concurrent ChemoRT), 38,
 118,
 144, 174–75, 178 change, 2, 115–19
 chemotherapy, 3, 20–21, 32, 38, 136 Chi-
 Chang, 5, 31, 128
 classes, 37, 41, 47, 49–50, 52, 60, 113,
 151 model-based feature selection, 58
 classification accuracy, 24, 60, 124
 classification algorithm, super, 120,
 123–24 classification algorithm model,
 super, 71 classification model, 5–6,
 41–43, 65–66, 69,
 81, 88, 92, 95, 99, 101, 103, 120, 122
 developed hybrid stacked ensemble, 7
 hybrid prognostic, 120
 hybrid stacked ensemble, 6–7, 63 hybrid
 stacked ensemble prognostic, 120
 recurrent HNSCC prognostic, 120
 classification model data, 69
 Classification model for recurrent
 HNSCC, 77
 classification model for recurrent HNSCC
 prognosis, 81
 classification prediction, 93, 96 classifier
 models, 37, 60, 74–75, 79
 five base, 75, 122, 124
 classifiers, second-level, 44–45, 67–68,
 73, 96
 clinicopathologic, 6, 22–23, 31, 128
 clinicopathologic and genomic
 markers, 31,
 128 Colorectal, 14–15
 combination, accurate, 7 combination of
 genomic and
 clinicopathologic makers for recurrent
 HNSCC prognosis, 5
 comparative performance metrics of base
 models, 107
 comparison of baseline stacked ensemble
 models, 105, 111
 Comparison of Baseline Stacked
 Ensemble Models and HESCA Model
 Performance, 108
 comparison of Baseline Stacked
 Ensemble Models and state-of-the-art
 model, 111
 Comparison of Base Models and HESCA
 Model Performance, 106–7
 Comparison of HESCA model, 105
 components of stacked ensemble
 models, 77 computers, 8–9, 36
 Concurrent ChemoRT. See CCRT
 control, 151, 169
 cross, 29–30, 70, 76, 153–54, 158–60
 cross-validated predictions, 70–71, 74,
 89–90, 95
 Cross Validation. See CV
 curative intent treatment, 33–34, 93, 96
 CV (Cross Validation), 6, 27, 67–68,
 70,
 121, 178

D

Decision Trees. See DT
 Deep Neural Network. See DNN depth,
 19, 76, 153, 157–59, 162
 depth of invasion, 19, 23–24, 28, 30, 35
 diagnosis, 2, 9, 13, 20, 22, 25, 34–35,
 71,
 117–19, 121, 126 disease, 2, 8, 13, 22
 Distributed Random Forest. See DRF
 DNN (Deep Neural Network), 1, 26,
 52,
 63–64, 66, 69, 76–77, 79, 82, 88–97, 100–
 104, 106–7, 110, 120–21, 163–64

DRF (Distributed Random Forest), 2, 26–27, 63–64, 66, 69, 76–77, 79, 88–97, 99–104, 106–7, 110, 120–21, 154–56, 159–60, 162–64

DT (Decision Trees), 2, 10, 24–28, 30, 42, 45, 57–58, 125

E

EFS. See Ensemble Feature Selection ensemble, 1, 4, 6, 42–44, 46, 63–64, 67–70,

72, 85, 154–55, 158–60 ensemble classifier, 67–68, 73

ensemble feature selection, 57, 63–64, 125 ensemble feature selection technique, 6,

38–39, 63, 122

ensemble feature technique, 36

ensemble learning, 4, 7, 42–44, 131, 135–36

stacked, 4, 66 ensemble model

best stacked, 66 best-stacked, 77

equation, 47–51, 54, 56, 59–62

evaluation metrics, 93, 96, 99–102, 109–10 Exarchos, 3, 17, 22–23, 30, 130, 132

exceptions, 115–19

expression, 60–61, 134

F

false negative (FN), 60, 178 false positive.

See FP Feature Age, 116

feature Age on recurrent HNSCC prognosis, 116

feature importance, 57–58, 64 feature Nodes on recurrent HNSCC

prognosis, 115

feature on recurrent HNSCC prognosis, 118 features, 4, 6, 35, 37–39, 47–50, 54, 56–58,

64, 77, 79–85, 115, 117–19, 122, 124, 170–71

boosted, 6, 63, 65, 74, 120, 124 gene expression, 24

texture, 24

feature selection, 4, 23, 43, 56, 63–64, 72, 77–78, 82, 123, 128, 178

feature selection technique for recurrent HNSCC prognosis, 86

feature selection techniques, 8, 36, 38, 57, 63–65, 77, 79, 82, 84–86, 120–22, 124

Feature Selection Techniques on Test Sets, 87

Feature Selection Techniques on Train Sets, 87

feature selectors (FS), 57, 64, 72, 84–85, 120, 167–68, 178

feature selectors, single, 64 Feature Smoke, 116–17

feature Smoke on recurrent HNSCC prognosis, 116–17

features of genomic data, 79

feature subsets, 57–58, 64, 72, 74, 77, 82, 84–86, 121, 123–24

Feature Subsets Feature Selectors Dataset, 86

final predictions, accurate, 42

first-level classifiers, 44–45, 66–68, 70–73, 89–91

five base classifiers, 5, 63, 77, 88–91, 94, 96–97, 109–11, 122

Five feature selection techniques, 64, 121 FN (false negative), 60, 178

focus of cancer prediction and prognosis, 2 focus of cancer prognosis and prediction, 9 fold, 70, 76, 153–54, 158, 160–61 Foundation, 13, 20–21, 130, 137

FP (false positive), 60, 178 FS. See feature selectors FS techniques, 57, 84, 121

full-input features, 64, 77–79, 81–82, 105–6, 121–22

G

GBFs (Gradient Boosted Features), 77, 79
 gbm, 6, 26–27, 46–47, 63–64, 66, 72, 74,
 76–77, 88–97, 99–102, 106–7, 109–11,
 120–22, 153–59, 163–64
 GBM ensemble feature selection
 technique, 6, 86
 GBM-FS, 57, 63–64, 66, 77, 79, 84–86,
 88, 105–7, 121–23, 167
 ensemble feature selection technique, 63
 GBM-FS ensemble feature selection
 technique, 79, 122 GBM-FS features, 86,
 105
 GBM-FS Optimal Feature Subset Base
 Classifiers, 88
 GBM-FS technique, 69, 85–86, 88, 92,
 123 GBM meta-classifier, 91, 104, 110,
 124 GBM meta-model, 63, 66, 77, 102
 Generalised Linear Model, 56
 generalised prognostic model for
 recurrent HNSCC prognosis, 31
 genomic, 3–6, 10, 22, 28–30, 121, 134–35
 genomic features, 23–24, 36
 genomic markers, 3, 18, 31, 120, 122, 128
 giving correct classification, 93, 96–
 97, 99,
 101–3
 glm, 25–27, 56, 63–64, 66, 69, 76–77, 79,
 82, 88–97, 99–104, 106–7, 110, 120–
 21, 153–57, 163–64
 glm1, 99–100, 108, 154–57, 165–66
 glm2, 101, 108, 154–57, 165–66
 GLM2 on Training and Test Sets, 101
 GLM classifiers, 90, 94
 GLM-FS, 84–86, 167
 GLM meta-classifier, 91, 99–100, 104,
 109 GLOBOCAN, 12, 14–17
 GradeG2, 84–85
 GradeG3, 84–85
 Gradient Boosted Features (GBFs), 77, 79
 Graph of Stacked Ensemble Models,
 112 Graphs for Prognostic Features,
 39–40 Gynecological, 14–15

H

Head and neck cancer, 2, 19, 126–28, 178
 Head and Neck Cancer. See HNCs
 Head and Neck Squamous Cell
 Carcinoma. See HNSCC
 Hematological, 14–15
 HESCA (Hybrid Ensemble Super
 Classification Algorithm), 4, 31, 63,
 65, 72, 105, 108, 110, 114, 120, 122–
 24, 163–64
 HESCA Classification Model Prediction,
 112
 HESCA model, 63–65, 69, 71–72, 74–75,
 77–79, 81–82, 85–86, 105–7, 109–11,
 113–14, 120–24, 165–66
 input, 121–22 developed, 120 full-input,
 77 hybrid, 121 testing set, 105
 HESCA model analyses, 88, 92
 HESCA Model Evaluation on Test Set, 92
 HESCA model for better performance,
 68 HESCA model for training and
 testing data,
 86
 HESCA Model HESCA Model Metrics,
 105 HESCA Model Hyperparameters,
 76
 HESCA Model Hyperparameters for
 Recurrent HNSCC Prognosis
 Classifiers Hyperparameters, 76
 HESCA Model Hyper-parameters
 Identification, 75
 HESCA model learning technique, 63
 HESCA model on GBM-FS features,
 86 HESCA Model on Original Test
 Training
 Set, 81
 HESCA model on test set, 82, 108, 111–
 12 HESCA Model on Training Set,
 107, 112 HESCA model outperforms,
 108, 110 HESCA model outperforms
 base models,
 107

HESCA model parameters, 78
 HESCA Model Performance on Test Data, 107, 164
 HESCA Model Performance on Test Set, 166
 HESCA Model Performance on Training and Test Sets Training Set, 108, 110
 HESCA Model Performance on Training Data, 106, 163
 HESCA model to learn, 64
 HESCA prognostic model, 79, 122
 HESCA prognostic model for recurrent HNSCC prognosis, 79
 history, 37, 138–48, 174–75
 HNSCC (Head and Neck Squamous Cell Carcinoma), 1–4, 7–8, 10, 12, 16, 18–19, 22, 24, 29, 33, 35, 49, 126–27, 134, 179
 HNSCC patients, recurrent, 5, 25
 HNSCC recurrence, 1–4, 60, 90, 95, 119
 HNSCC recurrence verses nonrecurrence, 118
 HNSCC subtypes, recurrent, 31, 122–24
 HPV (Human Papilloma Virus), 18, 35, 37–38, 79–80, 84, 122, 143–44, 174–75
 Human Papilloma Virus. See HPV
 hybrid ensemble classification system, 63
 Hybrid Ensemble Super Classification Algorithm. See HESCA
 hybrid stacked ensemble-based model, 6

I

imaging, 23, 29–30
 importance, 57, 64, 77, 82, 84
 indicating, 12, 82–83, 90, 96, 104, 111, 113–14
 input features, 36, 38, 54, 63, 86, 89, 105
 input features and HESCA model, 105
 inputs, 44–47, 53–55, 67–68, 71–72, 161
 instability, unstable, 57

instances, 10, 34, 36–38, 41, 64, 69, 79, 115–19, 125, 138–48
 International Agency for Research on Cancer (IARC), 17, 127
 invasion, 19, 23–24, 28, 30, 35, 37, 80, 84–85, 145–46, 172–73

K

Kabir and Ludwig, 4–5, 25–27, 66, 68–69, 124
 KNN (K-Nearest Neighbours), 2, 24–25, 27–29, 151
 Kwon, 4–5, 25–27, 66, 68–69, 124, 132

L

Large training feature sizes, 36
 larynx, 2, 11, 15, 17, 34, 38, 140, 172–73
 layers, 53, 55
 hidden, 53–55
 least log-loss value, 86, 88, 90, 94, 97, 106, 108–11
 locoregional recurrences, 23, 126
 logloss, 76, 100, 105, 153, 157–68
 loss functions, 46, 66
 LR (Logistic Regression), 23, 28, 56, 133, 179
 Ludwig, 4–5, 25–27, 66, 68–69, 124, 131
 lymph nodes, 10, 20, 22, 35, 115, 122, 131

M

machine learning, 1, 8–9, 66, 127, 129–30, 133, 179
 marginal effect, 115–19
 max, 49, 76, 113, 138–48, 153, 157–67
 Mehrotra, 18–19, 133
 meta-classifier models best, 74
 robust, 66, 119
 meta classifiers, 26, 70–71, 77, 90–91, 96–97, 101, 115, 122

meta-classifiers, 26, 66–69, 71–72, 74, 85, 89–91, 96–97, 99–100, 102, 104, 109, 111, 120, 122, 124–25

Meta Classifiers on Test set, 95, 97–98

meta-features, 70, 74

meta-learning algorithm, 4, 66

methods for feature selection and training, 4

methods for feature selection and training of

prognostic models, 4

metrics, 57, 60–61, 76, 85–86, 88, 99–101, 153, 158, 160, 162, 170

misclassification, 94, 97, 99, 101, 103, 113

model for recurrent HNSCC, 65

Model-GLM2, 109, 111

model on test set, 100, 102

model on training and testing sets, 77

model over-fitting, 36

model performance, 105

experimental, 79

Modulo keep, 76

Most Frequent Cancers in Ghana, 14

N

Nasopharyngeal, 25, 28–29

Nasopharyngeal Carcinoma. See NPC

neck, 8, 14–15, 18, 21, 115, 126

Neural Networks (NNs), 52–54, 132, 136

neurons, 53–55

NNs. See Neural Networks

nodes, 35, 37–38, 53, 58, 80, 84–85, 121–23, 139–48, 170–71, 174–75

noncancerous, 18, 21

Non-Recurrence, 49, 51, 60, 93, 96, 99, 101–2, 113

novel approach for recurrent HNSCC

prognosis in cancer, 68

NPC (Nasopharyngeal Carcinoma), 11, 24, 126, 128, 132, 140, 172–73

O

optimal features for recurrent HNSCC

prognosis, 6

optimal feature subset, 63–66, 69, 74, 77, 79, 82, 85–86, 88, 92, 106, 112, 121

optimal feature subset for recurrent HNSCC prognostic model, 121

Oral Squamous Cell Carcinoma (OSCC), 23, 133–35, 179

output layers, 53, 55–56

P

Pancreas, 14–15

patients, 1–3, 7, 10, 18–25, 33–37, 54, 60–61, 71–72, 93–94, 96–97, 99, 101–3, 113, 115–19, 121–22

advanced nasopharyngeal cancer, 136

recurrent, 93, 96

Patterns

average, 115–19

patient recurrence, 24

good, 41–42

model's, 114, 123

model training, 105

performance accuracy, 44

achieved better, 122

best, 26, 122

performance metrics, 62, 82, 85, 88, 90–91, 105, 109–11

comparative, 106–7

Performance Metrics of Base Classifiers on 10-Fold Cross-Validation Set, 90

Performance Metrics of Base Classifiers on Test Set, 94

performance metrics of baseline stacked ensemble models, 105

Performance Metrics of HESCA Model, 81, 86

performance metrics of meta classifiers, 91

Performance Metrics of Meta Classifiers on Test Set, 97

Performance Metrics of Stacked Ensemble, 100–101

Performance Metrics of Stacked Ensemble Model, 109, 167–68

Performance Metrics of State-of-the-Art, 103–4
 pharynx, 2, 11
 plot, 83, 114, 155–57, 170–71 Plot of Base Models, 108
 Plot of Performance of Feature Selection Techniques, 87
 precision, 34, 60–61, 81, 88, 90–91, 94, 97, 100–111, 119, 163–66
 precision value, 60, 88, 106 prediction functions, 70–71 prediction of HNSCC patients, 93
 prediction of therapy response for recurrent HNSCC patients, 25
 predictions, 8–9, 24–25, 42–44, 61, 65–66, 76, 91, 96, 99, 101–3, 113, 129–30, 153–54, 156, 158–60
 cancer recurrence, 2 classification model's, 112 correct, 60, 93 incorrect, 101, 103 predictor, 24, 59, 170–71
 prognosis, 1–6, 9, 18, 20, 22–25, 31, 49–50, 83, 85, 118, 120, 122–25, 129, 131 recurrent HNSCC, 7, 86
 prognosis for recurrent HNSCC, 6, 28, 31, 122
 prognosis for recurrent HNSCC patterns, 123–24
 prognosis for recurrent HNSCC subtypes, 31, 122
 prognosis of HNSCC recurrence, 4, 60, 90, 95
 prognostic accuracy result, improved, 18 prognostic features, 6, 36, 39–40, 115 prognostic model, 1, 4, 24–25, 125 generalised, 31 robust, 4, 31, 64 unstable, 5
 prognostic model accuracy, 24 prognostic results, 10, 23, 122
 unstable, 4

R

radiation, 20–21

radiotherapy, 3, 20–21, 32–33, 38, 79, 130, 179
 Radiotherapy and Oncology Department (ROD), 32, 79
 range, 47, 138–48, 163–68 training features, 64
 rate, 12, 16, 76, 116, 153, 157–60, 162 true negative, 60–61
 true positive, 61
 recall, 60–61, 81, 88, 90–91, 94, 97, 100–108, 110–11, 119, 163–66
 recall value, 106
 best, 91, 107, 109–10
 recurrence and nonrecurrence of breast cancer prognosis, 26
 recurrence verses nonrecurrence, 63
 recurrent HNSCC, 6–7, 24, 28, 31, 34–35, 65, 74, 77, 115–18, 120, 122–24 recurrent HNSCC probability, 115–19
 predicted, 115–19
 recurrent HNSCC prognosis, 5–6, 22, 31, 41, 63–64, 68, 74, 79, 81, 108, 115–22
 Recurrent HNSCC Prognosis Classifiers Hyperparameters, 76
 Recurrent HNSCC Prognosis Classifiers Hyperparameters in grid search, 76
 recurrent HNSCC prognosis datasets, 86 recurrent HNSCC prognosis in Ghana, 5
 recurrent HNSCC prognosis research, 7–8 recurrent HNSCC prognostic model, 121 result of inaccurate identification, 4 robust prognostic accuracy of cancer, 10 robust prognostic model for recurrent HNSCC prognosis, 31, 64
 ROC curve, 61, 66, 88, 90–91, 93–94, 96–97, 99–100, 102
 rows, 62, 76, 153, 157, 159, 162

S

salivary glands, 11, 15

SCCs. See Squamous Cell Carcinoma Singh, 22, 24, 29, 43–44, 135 SiteNPC, 84–85

smoking, 3, 17, 23–24, 30, 35, 116 space, meta-feature, 70–71

specificity, 60–61, 81, 88, 90–91, 94, 97, 100–111, 119, 163–66

specificity value, 94, 106 specify, 69, 138–48, 163–67

split, 58, 67–68, 70, 72, 74, 78, 151–52

Squamous Cell Carcinoma (SCCs), 1–2, 10, 12–13, 16, 23, 63, 126–27, 130, 132–35, 179

stability, 6, 59, 114, 123

Stable feature selection for biomarker discovery, 131

stacked ensemble, 5, 25, 31, 71, 91, 97–101, 122, 156–60

Stacked Ensemble Classification Model for HESCA, 65

stacked ensemble classification models, 5, 42, 66, 91, 120–24

stacked ensemble classifier model, 45–47

stacked ensemble learning technique, 4, 25

stacked ensemble learning technique of base classifiers, 4

stacked ensemble model and HESCA model, 110

Stacked Ensemble Model on Test Set, 102, 104

Stacked Ensemble Model on Training Set, 103

stacked ensemble models for recurrent HNSCC prognosis, 5

stacked ensemble of meta-classifier models, 97

stacked ensemble technique for recurrent HNSCC prognosis, 31

stacked ensemble techniques, 5, 25, 27, 31, 65, 88, 90–91, 94, 97–100, 102, 120

Stacked Ensemble Techniques in Cancer Study, 25

Stacked Model Metrics, 106–7

stack ensemble model, 26, 126

stacking, 4–5, 26, 44, 69, 91, 151

stacking ensemble, 25–26, 63, 66, 75, 77, 89–91, 96, 102, 104, 109, 111

stacking ensemble model, 66

staging, pathological tumor, 118–19

state-of-the-art, 5, 26, 79, 99, 102–5, 110, 124, 156–57, 179

state-of-the-art classification model, 121

state-of-the-art model, 105, 111

Step, 44–45, 65, 67–69, 72–73, 152

stopping, 76, 153, 158, 160–62

study for model evaluation purpose, 60

subtypes, 19, 21–22, 27–28

surgery, 3, 20–21

T

targeted therapies, 20–22

TCGA (The Cancer Genome Atlas), 24, 179

TCIA (The Cancer Imaging Archive), 24, 179

techniques, 1, 3–8, 10, 22–24, 26, 28, 31–32, 36–37, 42–44, 63–66, 85–86, 120, 124–25

modelling, 42

stacking, 44, 85

techniques and feature selection techniques, 8

techniques and feature selection techniques in recurrent HNSCC prognosis research, 8

techniques in modelling cancer prognosis, 1

techniques in recurrent HNSCC, 24

Techniques in Recurrent HNSCC Prognosis, 22

test, 22, 35, 37, 64, 124, 152–53, 156–62, 169–70

test accuracy, 86, 124

testing sets, 67–69, 72–74, 77–79, 86, 93, 99–102, 104–6, 108, 111, 114

TN. See true negative tobacco, 17, 131
 tolerance, 76, 153, 158, 160–62 TP (true positive), 60–61, 179
 train, 22, 47, 58–59, 123–24, 152–55, 158–62, 169–70
 training and evaluation metrics of baseline stacked ensemble models, 109
 training and test accuracy, 124 training and testing sets, 74, 77–78, 86, 100–101, 105
 training and test performance parameters, 123–24
 training examples, 36, 83, 114
 training features, 38, 56, 58, 83, 121–22 training feature TreatCCRT, 83
 training instances, 37, 41, 52, 64, 82, 121, 123–24
 training loss, 59, 114, 123
 training set and testing set, 114 training set and test set, 81, 86, 121
 training set for meta-classifier, 67–68, 72 training set for second-level classifiers, 67–68, 73
 treatment, 1, 3–4, 13, 19–22, 71, 79, 93, 96, 118, 121, 126
 treatment type, 22, 64

true negative (TN), 60–61, 179 true positive. See TP
 tumor, 2–3, 8, 10, 19–22, 24, 35, 71, 117, 119, 179
 tumor suppressor genes, 18–19, 117

U

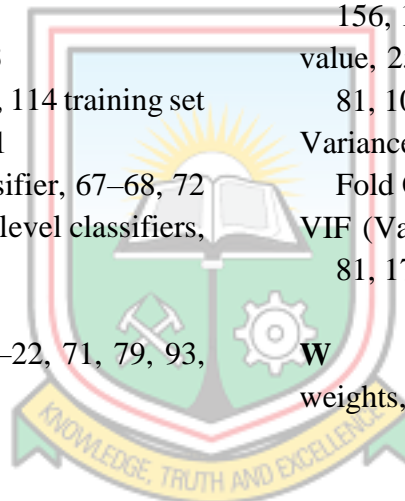
unstable instability of FS techniques in training, 57
 Urological, 14–15

V

validation, 7, 29–30, 59, 76, 114, 153–54, 156, 158–61
 value, 25, 48–49, 56, 58–59, 61, 70, 80–81, 106, 113
 Variance Inflation Factor. See VIF V-Fold Cross-Validation, 58, 69, 74
 VIF (Variance Inflation Factor), 59, 80–81, 179

W

weights, 44, 47, 53, 55, 64



APPENDIX N SIMILARITY INDEX

Damianus Kofi Owusu_Thesis

by Damianus Kofi Owusu

Submission date: 27-Oct-2023 02:07PM (UTC+0000)
Submission ID: 2209059053
File name: Damian_Full_Final_PhD_Thesis_Similarity_index.pdf (2.89M)
Word count: 35376
Character count: 182377



Damianus Kofi Owusu_Thesis

ORIGINALITY REPORT

15%	10%	10%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ijasr.forexjournal.co.in Internet Source	1%
2	en.wikipedia.org Internet Source	1%
3	Submitted to University of Hong Kong Student Paper	<1%
4	www.diva-portal.se Internet Source	<1%
5	doc.lagout.org Internet Source	<1%
6	stackoverflow.com Internet Source	<1%