

ENSEMBLE-BASED SUPERVISED LEARNING FOR PREDICTING DIABETES ONSET

NONSO ALEXANDA NNAMOKO

BEng (hons), MSc, AHEA

**A thesis submitted in partial fulfilment of the requirements of
Liverpool John Moores University for the Degree of Doctor of
Philosophy**

July 2017

ACKNOWLEDGEMENT

This thesis represents not only my work at the keyboard, but also a milestone in over three years of research work and career development at Liverpool John Moores University (LJMU); particularly within the Centre for Health and Social Care Informatics (CHaSCI). My experience at CHaSCI has been nothing short of amazing. From my first day of PhD research, I have felt at home. I have been given unique opportunities to improve my knowledge which includes engaging in teaching and learning activities, contributing in other research projects within CHaSCI, writing and presenting research papers to internal and external audience and many more. Throughout my years within CHaSCI, I have learned that there would be no research work without funding. In fact this research work is a result of funding from both LJMU and the Royal Liverpool and Broadgreen University Hospital (NHS) Trust (RLBUHT). This thesis is also the result of work and many experiences I have encountered from a lot of remarkable individuals who I wish to thank.

First and foremost I wish to thank God for granting me the strength, ability and determination to complete this PhD research. I would like to express my deepest gratitude to my family for the support they provided me through my entire life and in particular, I must acknowledge my children Leona and Logan, for putting a smile on my face each day; and my wife and best friend, Chioma, without whose love, encouragement and editing assistance, I would not have finished this thesis.

Very special thanks go out to my former director of studies, Dr Farath Arshad who retired unfortunately before my PhD ended. Without Farath's support, I would not have had the opportunity to undertake this PhD research work. Farath believed in me from the start and that belief made a difference in my life. It wasn't until I met her, that I truly developed an interest in Healthcare Informatics and her advice led to the work undertaken in my PhD work. She provided me with direction, technical support and became more of a mentor and friend, than a supervisor. She has been supportive since the first day we met when I helped her with a web development project. I remember she used to

say something like "there would be plenty of time to relax later!" to encourage me to work harder. Farath has also supported me emotionally through the rough road to finish this thesis. During the most difficult times when writing this thesis, she gave me the moral support and the freedom I needed to carry on. I doubt that I will ever be able to convey my appreciation fully, but I owe her my eternal gratitude.

I would also like to express my gratitude to my new director of studies, Dr Abir Hussain, for her patient guidance, enthusiastic encouragement and useful critiques of this research work. Dr Abir has helped immensely to keep my progress on schedule.

Special thanks go to the other members of my supervisory team, Dr David England and Professor Jiten Vora for the assistance they provided at all levels of the research project. I benefited immensely from David's vast knowledge and skill especially in technical writing which has helped in shaping my writing skills. Jiten provided the much needed clinical guidance but also his professional network led to the materials (data) used in this research.

I must also acknowledge Joe Bugler of Abbott Diabetes UK for facilitating the provision of data through his office to help this research work. Appreciation also goes out to Lucy Wilson for providing data at the initial stage of the research and to Danny Murphy, for his programming assistance to develop a prototype system for conference display. Many thanks to the technical support group at the School of Computing and Mathematical Science (CMS), LJMU for all of their computer and technical assistance throughout my PhD; and to the office staff Tricia Waterson, Carol Oliver and Lucy Tweedle for all the instances in which their assistance helped me along the way. Thanks also goes out to those who provided me with advice at times of critical need; Professor Abdenor El Rhalibi, Dr. William Hurst, Dr. Shamaila Iram and Dr Ibrahim Idowu. I would also like to thank my fellow PhD students at CMS, LJMU for our technical debates, exchanges of knowledge, skills, and venting of frustration during my PhD program, which helped enrich the experience.

In conclusion, I recognize that this research would not have been possible without the financial assistance of the Faculty of Technology and Environment, LJMU and the IT Innovations department at RLBUHT. I also express my gratitude to the following organisations for providing travel grants for conferences: PGR, LJMU and AISB.

ABSTRACT

The research presented in this thesis aims to address the issue of undiagnosed diabetes cases. The current state of knowledge is that one in seventy people in the United Kingdom are living with undiagnosed diabetes, and only one in a hundred people could identify the main signs of diabetes. Some of the tools available for predicting diabetes are either too simplistic and/or rely on superficial data for inference. On the positive side, the National Health Service (NHS) are improving data recording in this domain by offering health check to adults aged 40 - 70. Data from such programme could be utilised to mitigate the issue of superficial data; but also help to develop a predictive tool that facilitates a change from the current reactive care, onto one that is proactive.

This thesis presents a tool based on a machine learning ensemble for predicting diabetes onset. Ensembles often perform better than a single classifier, and accuracy and diversity have been highlighted as the two vital requirements for constructing good ensemble classifiers. Experiments in this thesis explore the relationship between diversity from heterogeneous ensemble classifiers and the accuracy of predictions through feature subset selection in order to predict diabetes onset. Data from a national health check programme (similar to NHS health check) was used. The aim is to predict diabetes onset better than other similar studies within the literature.

For the experiments, predictions from five base classifiers (Sequential Minimal Optimisation (SMO), Radial Basis Function (RBF), Naïve Bayes (NB), Repeated Incremental Pruning to Produce Error Reduction (RIPPER) and C4.5 decision tree), performing the same task, are exploited in all possible combinations to construct 26 ensemble models. The training data feature space was searched to select the best feature subset for each classifier. Selected subsets are used to train the classifiers and their predictions are combined using k-Nearest Neighbours algorithm as meta-classifier.

Results are analysed using four performance metrics (accuracy, sensitivity, specificity and AUC) to determine (i) if ensembles always perform better than single classifier; and (ii) the impact of diversity (from heterogeneous

classifiers) and accuracy (through feature subset selection) on ensemble performance. At base classification level, RBF produced better results than the other four classifiers with 78% accuracy, 82% sensitivity, 73% specificity and 85% AUC. A comparative study shows that RBF model is more accurate than 9 ensembles, more sensitive than 13 ensembles, more specific than 9 ensembles; and produced better AUC than 25 ensembles. This means that ensembles do not always perform better than its constituent classifiers. Of those ensembles that performed better than RBF, the combination of C4.5, RIPPER and NB produced the highest results with 83% accuracy, 87% sensitivity, 79% specificity, and 86% AUC. When compared to the RBF model, the result shows 5.37% accuracy improvement which is significant ($p = 0.0332$).

The experiments show how data from medical health examination can be utilised to address the issue of undiagnosed cases of diabetes. Models constructed with such data would facilitate the much desired shift from preventive to proactive care for individuals at high risk of diabetes. From the machine learning view point, it was established that ensembles constructed based on diverse and accurate base learners, have the potential to produce significant improvement in accuracy, compared to its individual constituent classifiers. In addition, the ensemble presented in this thesis is at least 1% and at most 23% more accurate than similar research studies found within the literature. This validates the superiority of the method implemented.

PUBLICATIONS

JOURNAL PAPERS & BOOK CHAPTERS:

J. Wilson, F. Arshad, N. Nnamoko, A. Whiteman, J. Ring and R. Bibhas (2013) "Patient Reported Outcome Measures PROMs 2.0: an On-Line System Empowering Patient Choice"; Journal of the American Medical Informatics Association, Vol 21, 725-729. DOI:10.1136/amiajnl-2012-001183

F.Arshad, N. Nnamoko, J. Wilson, R. Bibhas and M. Taylor (2014) "Improving Healthcare System Usability without real users: a semi-parallel design approach"; International Journal of Healthcare Information Systems and Informatics, Vol. 10, Iss. 1, 67 – 81. DOI: 10.4018/IJHISI.2015010104

N. Nnamoko, F. Arshad, L. Hammond, S. McPartland and P. Patterson (2016) "Telehealth in Primary Health Care: Analysis of Liverpool NHS Experience"; Applied Computing in Medicine and Health, Elsevier Edited Book, 269 - 286

N. Nnamoko, F. Arshad, D. England, J. Vora and J. Norman (2015) "Fuzzy Inference Model for Diabetes Management: a tool for regimen alterations"; Journal of Computer Sciences and Applications, Vol. 3, Iss. 3A, 40 – 45. doi: 10.12691/jcsa-3-3A-5

F.Arshad, L. Brook, B. Pizer, A. Mercer, B. Carter and N. Nnamoko (2017) "Innovations from Games Technology for Enhancing Communication among Children receiving End-of-Life Care"; British Medical Journal (working paper).

N. Nnamoko, F. Arshad, D. England, J. Vora and J. Norman (2017) "Ensemble Learning for Diabetes Onset Prediction"; IET Systems Biology – Special issue on Computational Models & Methods in Systems Biology & Medicine (working paper)

CONFERENCE PAPERS:

N. Nnamoko, F. Arshad, D. England, and J. Vora. (2013) "Fuzzy Expert System for Type 2 Diabetes Mellitus (T2DM) Management using Dual Inference Mechanism," Proc. AAAI Spring Symposium on Data-driven wellness: From Self tracking to Behaviour modification, 2013

N. Nnamoko, F. Arshad, D. England, J. Vora and J. Norman. (2014) "Evaluation of Filter and Wrapper Methods for Feature Selection in Supervised Machine Learning", 15th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, June 2014

N. Nnamoko, F. Arshad, D. England, J. Vora and J. Norman. (2014) "Meta-classification Model for Diabetes onset forecast: a proof of concept"; Proceedings of the IEEE International Conference on Bioinformatics and

Biomedicine, November 2014 [Note: selected for publication in extended form in a Special Issue of the journal 'IET Systems Biology'].

ABSTRACTS/POSTERS:

N. Nnamoko, F. Arshad, D. England, J. Vora and J. Norman (2013) “Intelligent Self-care System for Diabetes Support & Management”; Journal of Diabetes Science and Technology, March 2013.

Nonso Nnamoko, Farath Arshad, David England, Professor Jiten Vora (2015) “Evaluation of a Fuzzy Inference Model for continuous regimen alterations in Type 2 Diabetes”, Diabetes UK Professional Conference 2015

MAGAZINE ARTICLE:

Nonso Nnamoko (2014) “Social Media: an informal data source for healthcare intervention”; AISB Quarterly Magazine, 138: 20 – 22.

ABBREVIATIONS

ANN: Artificial Neural Network

AUC: Area Under the Receiver Operating Characteristic Curve

BG: Blood Glucose

BMI: Body Mass Index

CBR: Case Based Reasoning

CHaSCI: Centre for Health and Social Care Informatics

FN: False Negative

FP: False Positive

FPR: False Positive Rate

HBA1c or A1c: Glycated Haemoglobin

IT: Information Technology

LJMU: Liverpool John Moores University

MBR: Model Based Reasoning

ML: Machine Learning

NB: Naïve Bayes

NHS: National Health Service

OGTT: oral glucose tolerance test

RBF: Radial Basis Function

RBR: Rule Based Reasoning

REP: Reduced Error Pruning

RIPPER: Repeated Incremental Pruning to Produce Error Reduction

RLBUHT: Royal Liverpool and Broadgreen University Hospital (NHS) Trust

ROC: Receiver Operating Characteristic Curve

SMO: Sequential Minimal Optimisation

SMOTE: Synthetic Minority Over-Sampling Technique

SVM: Support Vector Machine

TN: True Negative

TP: True Positive

TPR: True Positive Rate

UK: United Kingdom

TABLE OF CONTENTS

Acknowledgement	1
Abstract.....	4
Publications	6
Abbreviations.....	8
CHAPTER 1: Introduction	17
1.1 Introduction	17
1.2 Research Aims	20
1.3 Research Objectives	20
1.4 Outline of the Chapters	20
CHAPTER 2: Literature Survey	22
2.1 Introduction	22
2.2 Diabetes and Screening Process	22
2.3 Computer Technology and Healthcare	25
2.3.1 Data driven Approaches To Diabetes Care	26
2.3.2 Machine Learning Ensembles	27
2.3.3 Review of Ensemble Methods and Related Research	31
2.4 Summary	38
CHAPTER 3: Technical Design Components	40
3.1 Introduction	40
3.2 Ensemble Member Classifiers	40
3.2.1 Support Vector Machines (SVM)	41
3.2.2 Artificial Neural Network (ANN)	42
3.2.3 Decision (classification) Trees	43
3.2.4 Naïve Bayes	45
3.2.5 Association Rule Learning	47

3.3	Experimental Data	47
3.4	Classifier Training Method	50
3.5	Performance Evaluation	52
3.6	Summary	55
CHAPTER 4: Methodology.....		56
4.1	Introduction	56
4.2	Design and Implementation	56
4.2.1	Feature Selection Approach	59
4.2.2	Stacked Generalisation	66
4.3	Summary	67
CHAPTER 5: Results & Analysis		69
5.1	Introduction	69
5.2	Base Level Performance with Full Training set	69
5.3	Feature selected subsets and performance	73
5.3.1	Naïve Bayes performance comparison	77
5.3.2	RBF performance comparison	78
5.3.3	SMO performance comparison	79
5.3.4	C4.5 performance comparison	80
5.3.5	RIPPER performance comparison	81
5.4	Ensemble Level Performances	83
5.4.1	Ensemble Vs Base Learner Performance	84
5.4.2	Impact of the Ensemble Method Implemented	86
5.5	Summary	92
CHAPTER 6: Conclusions & Future Work.....		94
6.1	Introduction	94
6.2	Restatement of Research Purpose	94
6.3	Limitations	95

6.4	Future Research	95
6.4.1	Variations of SMOTE Algorithm	96
6.4.2	Extended Research with different Weighted Vote	97
6.4.3	Base learner Optimisation and Further Experiments with External Dataset	99
6.4.4	Extended Research in Feature Search and Selection	100
6.5	Thesis Summary	100
Bibliography		103
Appendix A.1.....		118
Appendix A.2.....		120
Appendix A.3.....		122
Appendix A.4.....		126

LIST OF FIGURES

Figure 2.1: A guide for diabetes confirmatory test using HbA1c, FPG and/or OGTT (Source: [60])	25
Figure 2.2: Statistical reason why good ensemble is possible (Source [28]) ...	29
Figure 2.3: Computational reason why good ensemble is possible (Source [28])	30
Figure 2.4: Representational reason why good ensemble is possible (Source [28])	30
Figure 3.1: Simple Decision tree structure showing the root, internal and leaf nodes.	44
Figure 3.2: RIPPER algorithm (adapted from [142])	47
Figure 3.3: Data pre-processing operations applied on the original dataset.....	49
Figure 3.4: Visual representation of 10-fold cross validation method (Source: [154])	51
Figure 3.5: Simple confusion matrix or contingency table.....	53
Figure 3.6: Common performance metrics derived from a confusion matrix (Source: [157], [159]).	54
Figure 4.1: Experimental process of the base training feature selected subsets and ensemble training with K-NN algorithm.	57
Figure 4.2: Detailed diagram of feature selection (with Best-First search) and 10-fold cross validation	59
Figure 4.3: Best-First Algorithm with greedy step-wise and backtracking facility.....	62
Figure 4.4: A generic template for forward search (Source: [169])	63
Figure 4.5: Illustration of forward and backward selection drawbacks with 3 features.....	64
Figure 4.6: A generic template for bi-directional search (Source: [169])	65
Figure 4.7: Stacked generalisation using five base learners	66
Figure 5.1: Scatter plot showing class separation and distribution between BMI and other features of the experimental dataset.	70
Figure 5.2: Performance comparison between RBF and RIPPER models trained on full dataset.....	72
Figure 5.3: Naïve Bayes performance with full training set vs selected feature subset using Accuracy, Sensitivity, Specificity, AUC and Mc Nemar's test. ..	77

Figure 5.4: Graphic representation of Naïve Bayes performance trained on full dataset vs feature subset.....	78
Figure 5.5: RBF performance with full training set vs selected feature subset using Accuracy, Sensitivity, Specificity, AUC and Mc Nemar's test	78
Figure 5.6: Graphic representation of RBF performance trained on full dataset vs feature subset.....	79
Figure 5.7: SMO performance with full training set vs selected feature subset using Accuracy, Sensitivity, Specificity, AUC and Mc Nemar's test	79
Figure 5.8: Graphic representation of SMO performance trained on full dataset vs feature subset.....	80
Figure 5.9: C4.5 performance with full training set vs selected feature subset using Accuracy, Sensitivity, Specificity, AUC and Mc Nemar's test.....	81
Figure 5.10: Graphic representation of C4.5 performance on full dataset vs feature subset	81
Figure 5.11: RIPPER performance with full training set vs selected feature subset using Accuracy, Sensitivity, Specificity, AUC and Mc Nemar's test ...	82
Figure 5.12: Graphic representation of RIPPER performance on full dataset vs feature subset	83
Figure 5.13: Direct comparison of the 26 ensembles and RBF performance based on accuracy, sensitivity, specificity and AUC.....	85
Figure 5.14: EN-mod1 vs RBF performance using Accuracy, Sensitivity, Specificity, AUC and Mc Nemar's test	87
Figure 5.15: Graphic representation of EN-mod1 vs RBF model performance	87
Figure 5.16: EN-mod2 vs RBF performance using Accuracy, Sensitivity, Specificity, AUC and Mc Nemar's test	88
Figure 5.17: Graphic representation of EN-mod2 vs RBF model performance on AUC.....	89
Figure 5.18: EN-mod1 vs EN-mod3 performance using Accuracy, Sensitivity, Specificity, AUC and Mc Nemar's test	91
Figure 5.19: Graphic representation of EN-mod1 vs EN-mod3 model performance on AUC.....	91
Figure A.2.0.1: SMOTE algorithm (source: [151]).....	120
Figure A.3.0.1: Graphic representation of Naïve Bayes performance on balanced vs unbalanced dataset	122
Figure A.3.0.2: Graphic representation of RBF performance on balanced vs unbalanced dataset	123

Figure A.3.0.3: Graphic representation of SMO performance on balanced vs unbalanced dataset	124
Figure A.3.0.4: Graphic representation of c4.5 performance on balanced vs unbalanced dataset	124
Figure A.3.0.5: Graphic representation of RIPPER performance on balanced vs unbalanced dataset	125
Figure A.4.0.1: Data cluster of ‘age’ and other features of the training dataset	126
Figure A.4.0.2: Data cluster of ‘family pedigree’ and other features of the training dataset.....	126
Figure A.4.0.3: Data cluster of ‘bmi’ and other features of the training dataset	126
Figure A.4.0.4: Data cluster of ‘insulin’ and other features of the training dataset	127
Figure A.4.0.5: Data cluster of ‘skin fold’ and other features of the training dataset	127
Figure A.4.0.6: Data cluster of ‘blood pressure’ and other features of the training dataset.....	127
Figure A.4.0.7: Data cluster of ‘blood glucose’ and other features of the training dataset.....	128
Figure A.4.0.8: Data cluster of ‘pregnant’ and other features of the training dataset	128
Figure A.4.0.9: Scatter plot of the experimental dataset showing class distribution and density.....	128

LIST OF TABLES

Table 2.1: Guidelines for Body Mass Index classification and associated diabetes risk (Source [59]).....	24
Table 3.1: Five broad machine learning approaches and associated algorithms considered in this chapter.	41
Table 3.2: Characteristics of the Pima diabetes dataset from UCI database	48
Table 3.3: Characteristics of the revised dataset obtained from the Pima diabetes data.....	50
Table 3.4: Comparing k-fold cross-validation to other methods.....	52
Table 5.1: Results of base learner training with full experimental data	69
Table 5.2: Contingency table produced at base level experiment with full training dataset.....	71
Table 5.3: A guide for classifying the Accuracy of a model using AUC (Source: [158]).....	73
Table 5.4: Selected features for each classifier and performance based on the subsets.....	74
Table 5.5: Possible results of two classifier algorithms (Source: [189]).....	76
Table 5.6: Performance at ensemble level involving base classifier training (with data manipulation) in all possible combinations.	84
Table 5.7: Performance at ensemble level involving base classifier training (without data manipulation) in all possible combinations.....	90
Table 5.8: Research studies conducted with Pima Diabetes dataset	93
Table 6.1: Simple classification result from three classifiers, showing weighted predictions on each class	98
Table A.3.0.1: Tabular representation of Naïve Bayes performance on balanced vs unbalanced dataset	122
Table A.3.0.2: Tabular representation of RBF performance on balanced vs unbalanced dataset.....	123
Table A.3.0.3: Tabular representation of SMO performance on balanced vs unbalanced dataset.....	123
Table A.3.0.4: Tabular representation of C4.5 performance on balanced vs unbalanced dataset.....	124
Table A.3.0.5: Tabular representation of RIPPER performance on balanced vs unbalanced dataset.....	125

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

The research reported in this thesis is intended to explore methods through which health examination data generated in diabetes can be utilised to predict diabetes onset. Diabetes is a major cause of concern and its management is inherently a labour-intensive, complex and time-consuming task; requiring self-data tracking, medication and behaviour change from patients and a strong complementary component from clinicians who go through individual examination data to tailor therapy to patient needs [1]–[3]. Diabetes is caused by the malfunctioning of the pancreas, which secretes the hormone insulin, resulting in elevated glucose concentration in the blood. In some cases, the body cells fail to respond to the normal action of insulin.

Recent estimates suggest that around 3,333,069 adults are now living with diabetes in the United Kingdom (UK) [4]. This is an increase of more than 1.2 million adults compared with ten years ago when there were 2,086,041 diagnosed cases; and the number is estimated to rise to 5 million by 2025 [5]. This figure does not take into account the 549,000 adults estimated to have undiagnosed diabetes [5]. According to Diabetes UK [6], almost one in 70 people in the UK are living with undiagnosed diabetes. Several studies have revealed the potential to intervene and halt progression, if traces of diabetes are detected early [7]–[9]. Therefore, early identification of those at risk of developing the condition is vital so that prevention strategies can be initiated through lifestyle modifications and drug intervention [10]–[12].

On that note, several tools exist that use risk scores or questionnaire to identify people at risk of developing diabetes [13], [14]. One such tool is the ‘Know Your Risk’ [15] which is intended to help people identify their risk of developing Type 2 diabetes within the next ten years. The tool uses seven

variables (gender, age, ethnicity, family diabetes history, waist circumference, body mass index (BMI) and blood pressure history) for prediction. However, Abbasi et al. [16] warns that such simplistic tool may be unreliable because prediction is based on superficial data that can be accessed non-invasively. Such features cannot be considered sufficient to predict diabetes onset due to lack of vital information related to diabetes such as blood glucose concentration. Conventional biomarkers such as fasting plasma glucose (FPG) [17], glycated haemoglobin (HbA1c) and/or oral glucose tolerance test (OGTT) [18] are key features in diabetes screening. Inclusion of such features would lead to robust predictive models that approach full understanding of the condition.

Further research into available predictive models for diabetes onset did reveal some tools that use these biomarkers [16] and there is evidence that they predict cases slightly better than their simplistic counterparts. However, it emerged that the majority of those models were developed based on self-reported data. This data collection method is commonly used in healthcare but has been shown to be affected by measurement error as a result of recall bias [19]. For instance, subjects may not be able to accurately recall past events [20], [21]. Another concern about such data focusses on response bias, a general term used to describe a wide range of cognitive biases that influence the accuracy of participants' responses [22]. For instance, individuals tend to report what they believe the researcher expects to see and/or what reflects positively on their own abilities, knowledge, beliefs, or opinions [23]. According to Nederhof [24], responses of this sort are most prevalent in research studies that involve participant self-report. Response biases can have a big impact on the validity of data [22], [24]. Thus, the reliability of self-reported data is tenuous.

Medical health data obtained from health assessment programmes is a suitable alternative. For instance, the National Health Service (NHS) has rolled out a health screening programme aimed at identifying adults at high risk of developing diabetes [25]. Basically, adults aged 40 – 70 without pre-existing conditions are offered a health check to look for traces of five health conditions

including diabetes. Treatment usually commence for those who tested positive and the rest are invited for re-test in the next 5 years. There is potential through advances in computer science, to utilise data from such healthcare programme such that they can be used to predict diabetes onset.

Machine learning is the subfield of computer science used to construct computer models (known as algorithms or classifiers) that learns from data in order to make predictions on new examples [26]. For example, a single classifier can be trained with data from NHS health check so that it can make prediction whether a person is likely to develop diabetes before a re-test is due. Furthermore, advances in machine learning have given rise to multiple classifier learning (also known as ensembles) [27], which is widely known to perform better than a single classifier [28]–[32]. An ensemble is constructed by training a pool of single classifiers on a given training dataset and subsequently combining their outputs with a function for final prediction [33]–[35]. Various methods have been proposed for selecting the best pool of classifiers [36]–[38], designing the combiner function [29], [31], [39], [40], pruning strategies to reduce the number of classifiers within the ensembles [41]–[50], or even performance optimisation through feature selection [51]. However, the general prerequisite for constructing good ensembles is to ensure that the individual base classifiers are both accurate and diverse in their predictions. [27].

The research reported in this thesis is intended to build on this knowledge. In particular, it examines the effects of feature selection and heterogeneous base classifiers on ensemble performance. Five different classifiers are employed for the experiment, namely: Sequential Minimal Optimisation (SMO), Radial Basis Function (RBF) network, C4.5 decision tree, Naïve Bayes and Repeated Incremental Pruning to Produce Error Reduction (RIPPER). The classifiers belong to five broad families of machine learning algorithms with different operational concepts. The training data is obtained from a health assessment programme (similar to the NHS health check). Each classifier is trained on a subset of the full dataset that leads to optimum accuracy; and it is expected that their operational differences would introduce diversity, ultimately leading to the construction of good ensembles. The experimental design follows the

Bayesian theory [52] in which all possible probabilities in the search space are examined. Thus, all possible combinations of the five classifiers are explored and performance compared.

1.2 RESEARCH AIMS

This thesis presents experiments conducted with historic health examination data, to train machine learning ensembles capable of predicting diabetes onset. The aim is to construct a model that is more accurate than similar research found within the literature.

1.3 RESEARCH OBJECTIVES

Diversity and accuracy of base learners have been identified as vital factors for constructing good ensembles. Therefore, the research objectives are:

1. To exploit diversity from heterogeneous classifiers with differing operational principles. Five machine learning classifiers would be employed as base learners for the ensemble.
2. To optimise the accuracy of prediction through feature subset selection. A search algorithm would be used to search the feature space of the training data in order to select a subset for each of the base classifiers that lead to optimum accuracy.

It is expected that the operational differences from the base classifiers would introduce diversity. In addition, the feature subset selected for each classifier is expected to optimise their individual accuracy. Predictions from the classifiers would be used in all possible combinations to train ensemble models.

1.4 OUTLINE OF THE CHAPTERS

The remainder of this thesis is organised as follows:

Chapter 2 LITERATURE REVIEW: This chapter provides an overview of diabetes and its management strategies, with highlights to the relevant

features/variables required for screening. Furthermore, the chapter provides a concise review of generic ensemble methods and related research in the domain.

Chapter 3 TECHNICAL DESIGN COMPONENTS: This chapter presents a detailed description of the technical components used to design the ensemble method implemented in this thesis. The idea is to provide the reader with detailed information to aid full understanding of the methodology.

Chapter 4 METHODOLOGY: This chapter sets out the experimental design to achieve the research aims and objectives. Detailed procedure is presented on how diversity and accuracy can be exploited to construct a good ensemble model.

Chapter 5 RESULTS & ANALYSIS: This chapter sets out the findings from the experiments conducted in Chapter 4. Graphical representations are used to present the results with in-depth analysis to highlight their meaning and relevance to the research aims and objectives.

Chapter 6 CONCLUSIONS AND LIMITATIONS: This chapter summarises the entire research and reviews the findings. It also discusses the constraints on the implementation and outlines future work that can be undertaken to improve the research.

CHAPTER 2: LITERATURE SURVEY

2.1 INTRODUCTION

This chapter provides a brief overview of diabetes, its screening process, management challenges and the importance of early detection. As the project is aimed at predicting diabetes onset using multiple classifier models in machine learning, the majority of this chapter describes early and recent research into ensembles. A review of generic ensemble methods is presented with highlights to previous studies comparing the methods. Some formulations are presented to uncover the reason that ensembles often perform better than single classifiers.

2.2 DIABETES AND SCREENING PROCESS

Diabetes is a common life-long health condition where the amount of glucose in the blood is too high because the body cannot use it properly. This occurs as a result of low or no insulin production by the pancreas, to help glucose enter the body cells. In some cases, the insulin produced does not work properly (known as insulin resistance). There are 2 main types of diabetes – Type 1 and Type 2. However, it is important to note that the research presented in this research is focused on Type 2 diabetes among adults (≥ 18 years) only. Other types of diabetes include pre-diabetes (i.e., increased risk of developing type 2) and gestational diabetes (developed during pregnancy).

Type 1 is the least common, developed when the body cannot produce any insulin – a hormone that helps the glucose to enter the cells where it is used as fuel by the body. It is still unclear as to the exact cause of type 1 diabetes, but family history appears to be a factor. Onset of Type 1 diabetes is unrelated to lifestyle and currently cannot be prevented, although maintaining a healthy lifestyle is very important towards its management. This type of diabetes usually appears before the age of 40 and accounts for around 10 percent of all people with diabetes [53].

Type 2 however develops when the body can still produce some insulin, but not enough. This type of diabetes is more common and accounts for around 90 percent of people with diabetes. Age is considered a risk in type 2, with most cases developing in middle or older age; although it may appear early among some high-risk ethnic groups. For instance, in South Asian people, it often appears after the age of 25 [53]. Evidence also shows that more children are being diagnosed with the condition, some as young as seven [53], [54]. Type 2 has a strong link with lifestyle (i.e., overweight/obesity, physical inactivity and unhealthy diet).

Unlike type 1, onset of type 2 diabetes can be prevented or delayed so early diagnosis is important so that treatment can be started as soon as possible. Even more important is the need to identify individuals at high risk of type 2 diabetes, because evidence suggests that lifestyle adjustments can help delay or prevent diabetes [1]–[3], [11], [55]. A 10 year research study, conducted by the Diabetes Prevention Program (DPP), showed that people at high risk of developing diabetes were able to quickly reduce their risk by losing five to seven percent of their body weight through dietary changes and increased physical activity [10], [56]. The study sample maintained a low-fat, low-calorie diet and engaged in regular physical activity, five times a week for at least 30 minutes. As a result, the onset of type 2 diabetes was delayed by an average of 4 years. The study also indicates that these strategies worked well regardless of gender, race and ethnicity.

With the conventional screening process, type 2 diabetes is often undetected until complications appear, and reports shows that undiagnosed cases amount to approximately one-third of the total people with diabetes [57]. These cases are mostly discovered during hyperglycaemic emergency when the individuals have already developed diabetes [58]. In some cases, screening is triggered by abnormal readings during health check examination such as the NHS health check [25]. For instance, type 2 diabetes is heavily linked to physical inactivity and/or being overweight/obese, so abnormal body mass index (BMI) or waist circumference during such examination may trigger further screening. The benchmark for assessing BMI and waist circumference is shown in Table 2.1.

Waist circumference is often measured in centimetre (cm) with measuring tape and BMI is calculated using human weight and height as shown in the expression (1).

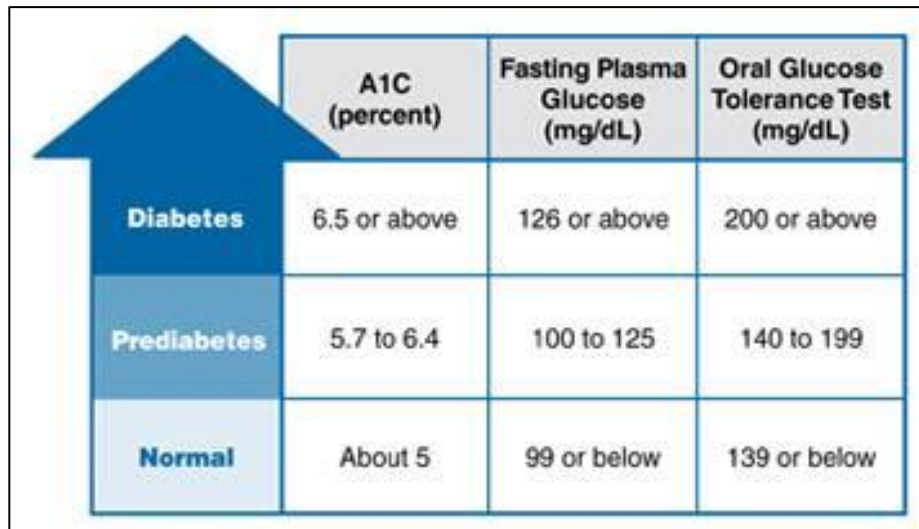
$$BMI = \frac{Wt \text{ in } kg}{(Ht \text{ in } m)^2} = \frac{Wt \text{ in } lb \times 703}{(Ht \text{ in } inches)^2} \quad (1)$$

Table 2.1: Guidelines for Body Mass Index classification and associated diabetes risk (Source [59])

	Body mass index (BMI)	Obesity class	Diabetes risk (relative to normal weight and waist circumference)	
			Men < 102 cm Women < 88 cm	Men > 102 cm Women > 88 cm
Underweight	< 18.5			
Normal	18.5 – 24.9			
Overweight	25.0 – 29.9		Increased	High
Obesity	30.0 – 34.9	I	High	Very high
	35.0 – 39.9	II	Very high	Very high
Extreme obesity	> 40.0	III	Extremely high	Extremely high

Further examination commences when an individual meets one or more of the above risk factors. Possible tests to assess for diabetes include urinalysis (urine test) and blood glucose concentration test, although the latter is the most widely used. Common blood-based diagnosis includes fasting plasma glucose (FPG) ≥ 126 mg/dL, 2-hr plasma glucose ≥ 200 mg/dL obtained during an oral glucose tolerance test (OGTT) or glycated haemoglobin test, commonly known as HbA1c $> 6.5\%$ [14]. The assessment criteria are shown much clearly in Figure 2.1.

The last couple of decades have seen enormous research in diabetes and an improved understanding of condition. The risk factors and bio markers are well researched and standardised recommendations exist for screening, diagnoses and management. However, this does not address the fact that a growing number of cases are still undetected. There is need for healthcare providers to transition from the current reactive screening process unto a model that is proactive so that individuals at high risk would be detected before onset. Fortunately, breakthroughs in research, information gathering, treatments and communications have provided new tools and fresh ways to practice and deliver healthcare.



	A1C (percent)	Fasting Plasma Glucose (mg/dL)	Oral Glucose Tolerance Test (mg/dL)
Diabetes	6.5 or above	126 or above	200 or above
Prediabetes	5.7 to 6.4	100 to 125	140 to 199
Normal	About 5	99 or below	139 or below

Figure 2.1: A guide for diabetes confirmatory test using HbA1c, FPG and/or OGTT (Source: [60])

2.3 COMPUTER TECHNOLOGY AND HEALTHCARE

The use of computer technology in healthcare (commonly known as healthcare informatics) has a long and interesting history, thanks to Charles Babbage's ideas on the first analytical computer system in the nineteenth century. It is very difficult to trace the origin of a major innovation, especially when it involves two or more disciplines (i.e., IT and healthcare). However, evidence suggests that healthcare informatics can be traced back to the twentieth century [61], [62], specifically in the early 1950s with the rise of computers [63]. This has since seen a series of revolutions from mere acquisition, storage and retrieval of data to more advanced models centred on patient needs and their contribution towards out-of-hospital care. Among the first published accounts is Einthoven's in 1906, where electrocardiograph data were transmitted over telephone wires [64]. Other reports include the 1957 medical image transmission [65], 1961 two way telephone therapy [66], nursing interactions in 1978 [67], clinician interaction in 1965 [68], education and training in 1970 and 1973 [69], [70], tele-visits to community health workers in 1972 [71], self-care in 1974 [72] and other applications.

A report by the World Health Organisation (WHO) [73] suggests that modern applications of IT in healthcare started in the 1960s, driven largely by the military and space technology sectors, as well as a few end-user demands for

readymade commercial equipment [62], [74]. For instance, the National Aeronautics and Space Administration (NASA) developed a manned space flight program to monitor and capture astronauts' health status such as heart rate, blood pressure, respiration rate and temperature while aboard. Despite these advancements, healthcare informatics saw a huge decline by mid 1980s with only one of the early North American programs recorded as still running [75]. This was quickly rectified in the early 1990s with an increase in federal funding of rural healthcare informatics projects, especially in the United States of America [76]. Since then, growth in healthcare informatics has continued to encompass information systems designed primarily for physicians and other healthcare managers and professionals. With the advent of internet services, there is now increasing interest in advanced approaches that analyse and make inference using stored data/information.

2.3.1 DATA DRIVEN APPROACHES TO DIABETES CARE

Two broad methods with data-driven capabilities in healthcare are heuristics based and model-based approaches [77]. The heuristics based approach works better with implicit knowledge [78]. Implicit knowledge is not directly expressed but inherent in the nature of the subject domain. It is mostly based on individual expertise and can be represented by non-standardised heuristics that even experts may not be aware of.

Simple case-based reasoning (CBR) is a good example of heuristics based approach for managing knowledge of the implicit nature [20],[39]. CBR utilises the specific knowledge of previous occurrences (commonly known as cases). Its operating principle is based on retrieving and matching historic cases that are similar to current ones, then applying the most successful previous case as the solution. To implement CBR, historic cases are structured into problems, solutions and outcomes based on the expert's problem detection strategies, and then used to solve new cases. Each structure can be reused and the current case can be retained in a case repository for future use. The case repository enables one to keep track of the subject evolution, and can be easily upgraded through the addition of new cases and possibly the deletion of obsolete ones.

That said, Lehmann and Deutsch [79] highlighted some limitations of heuristics based approaches, claiming that their role in patient care will be limited by the lack of, or incomplete information to develop a case. In agreement, Bichindaritz and Marling [80] argued that case-based reasoning systems require cooperation between the various information systems which may often be impractical or expensive. On that note, Lehmann and Deutsch [79] suggested that model-based approaches often based on explicit knowledge may be a useful alternative.

Explicit knowledge is well established, standardised, often available in books/research articles and can be represented by some formalism for developing knowledge-based systems [77]. Among the methods for representing and managing knowledge of the explicit type are the rule based reasoning (RBR) approach such as fuzzy logic; and the statistical/data mining approach such as machine learning. Model-based approaches have been applied successfully in diabetes management. For instance, Dazzi et al. [81] and San et al. [82] presented models aimed at diabetes management using neuro-fuzzy inference. Another example is the automated insulin dosage advisor (AIDA) – a mathematical model to simulate the effects of changes in insulin and diet, on blood glucose (BG) concentration [83],[84]. The authors declared the model insufficiently accurate for patient use in BG – insulin regulation, but believe there is value in its capability as an educational tool for carers and researchers. In fact, Robertson et al. [85] trained an artificial neural network (ANN) model for BG prediction with simulated data from AIDA.

Based on the evidence present, it is fair to say that available knowledge about diabetes is of explicit nature and therefore lend itself to the (model-based) machine learning approach implemented in this thesis.

2.3.2 MACHINE LEARNING ENSEMBLES

Ensembles are machine learning systems that combine a set of classifiers and use a vote of their predictions to classify new data points [28], [33], [86]. In a standard classification problem, the fundamental training concept is to approximate the functional relationship $f(x)$ between an input $X =$

$\{x_1, x_2, \dots, x_n\}$ and an output Y , based on a memory of data points $\{x_i, y_i\}$ where $i = 1, \dots, N$. Usually The x_i is a vector of real numbers and the y_i is a real numbers (scalar), drawn from a discrete set of classes. For instance, let p -dimensional vector $x \in R^p$ denote a pattern to be classified, and scalar $y \in \{\pm 1\}$ denote its class label. Given a set of N training samples $\{(x_i, y_i), i = 1, 2, \dots, N\}$ a classifier outputs a model h that represents a hypothesis about the true function $f(x)$; so that when new x values are presented, it predicts the corresponding y values. An ensemble is therefore a set of classifier models h_1, \dots, h_n whose individual decisions are combined by weighted or unweighted vote to classify new examples.

The general benchmark for measuring ensembles performance is the accuracy of individual classifiers that make them up. According to Hansen and Salmon [27], a necessary and sufficient condition for an ensemble of classifiers to be more accurate than its constituent members lies within the individual accuracy and the diversity of their outputs. A classifier is said to be accurate if its error rate is better than random guessing on new x values. On the other hand, two classifiers are said to be diverse if they make different errors on new data points [28]. For instance, consider the classification of a new value x using an ensemble of three classifiers $\{h_1, h_2, h_3\}$. If the three classifiers are not diverse, then when $h_1(x)$ is correct, $h_2(x)$ and $h_3(x)$ will also be correct. However, diverse classifiers will produce errors such that when $h_1(x)$ is wrong $h_2(x)$ and $h_3(x)$ may be correct, so that a majority vote will classify x correctly.

This informal depiction is fascinating but does not address the question of whether it is possible to construct good ensembles. Dietterich [28] addresses this question theoretically, using three fundamental reasons.

Statistical issue – Consider a single classifier searching a space H of hypotheses to identify the best hypothesis. A statistical problem arises when the size of the hypotheses space H is disproportionately bigger than the amount of training data available. Without sufficient data, the classifier is likely to identify many different hypotheses in H with same accuracy on the training data. By constructing an ensemble out of all of these hypotheses, the model can

average their votes and reduce the risk of choosing the wrong one as shown in Figure 2.2. The outer curve denotes the hypothesis space H while the inner curve denotes the set of hypotheses that produced good accuracy on the training data. The point labelled f is the true hypothesis and it is fair to say that averaging the accurate hypotheses would lead to a good approximation to f .

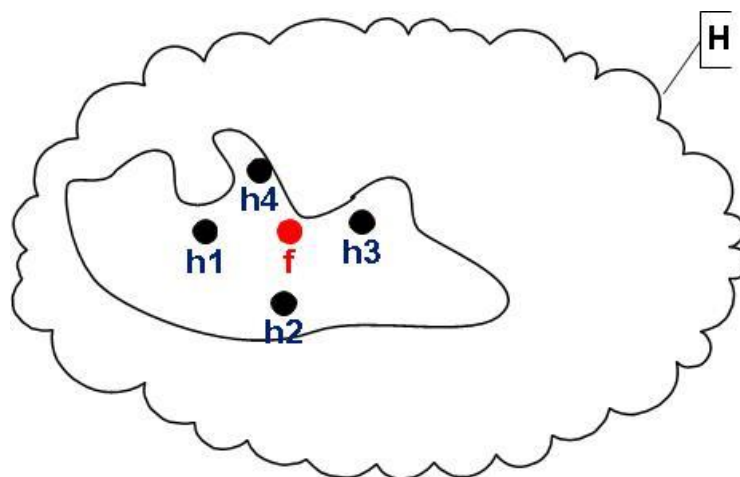


Figure 2.2: Statistical reason why good ensemble is possible (Source [28])

Computational issue: Assuming there is enough training data so that the statistical problem is absent, it may still be very difficult computationally for a classifier to find the best hypothesis within the search space. Many classifiers work by conducting some form of local search that may get stuck in local optima. For instance decision trees such as C4.5 grows the tree by using a greedy search rule that typically makes the local optimum choice at each stage with the hope of finding a global optimum [87]. An exhaustive search would be computationally difficult through this means. An ensemble constructed by running the local search from many different starting points (e.g., variations of the same classifier or even different classifiers) may provide a better approximation to the true unknown function than a single classifier as shown in Figure 2.3.

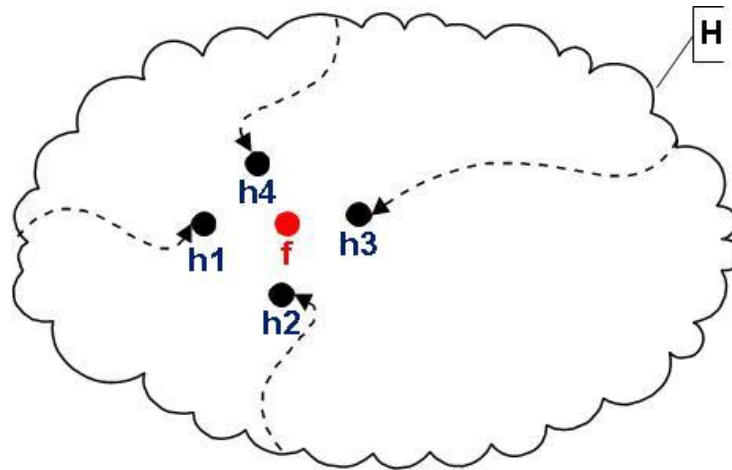


Figure 2.3: Computational reason why good ensemble is possible (Source [28])

Representational issue: It is possible that the true function f cannot be represented by any of the hypotheses in H as shown in Figure 2.4. However, this can be achieved through ensemble voting. Thus, by using a weighted or unweighted votes of hypotheses drawn from within H , it may be possible to arrive at the true function f .

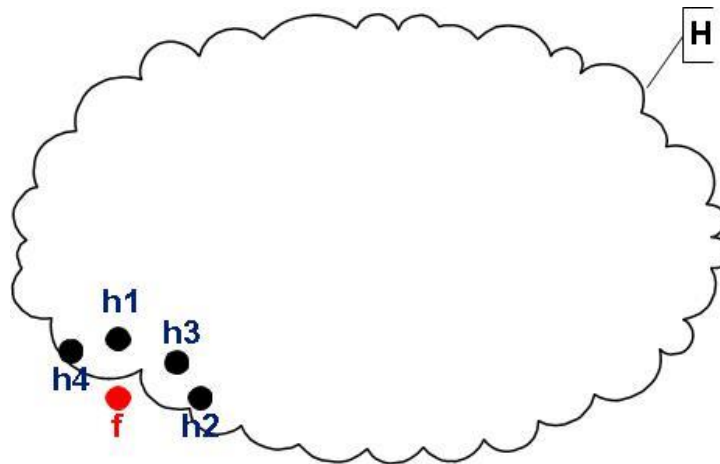


Figure 2.4: Representational reason why good ensemble is possible (Source [28])

That said, it is important to note that H does not always represent the space of hypotheses. For instance neural networks and decision trees classifiers perceive H as a space of all possible classifier models rather than hypothesis. As such, many research studies have reported asymptotic representation for them [88]–[90]. This means that, they explore the space of all possible classifier models when given enough training data. With a modest training dataset however, they only explore a finite set of hypotheses (not classifier models) and stop searching when they find a hypothesis that fits the training data. The

illustration in Figure 2.4 considers the space H as the effective space of hypotheses searched by the classifier.

2.3.3 REVIEW OF ENSEMBLE METHODS AND RELATED RESEARCH

One of the most active areas of research in ensembles has been to study methods for constructing good pool of classifiers. The original ensemble method is Bayesian model averaging (BMA) which samples each model within the ensemble individually and predictions are averaged and weighted by how plausible they are [91], [92]. Several modifications to BMA has given rise to a number of ensembles, most notably Buntine's work to refine Bayesian Networks [50], Bagging [36], Boosting algorithm [93], [94] and efforts by Hansen & Salamon to validate the Boosting algorithm [27]. This section presents an in-depth discussion about BMA and other general purpose ensemble methods applicable to other classifiers.

2.3.3.1 BAYESIAN MODEL AVERAGING

When presented with a training sample S , a standard ensemble outputs a set of classifier models h_1, \dots, h_n that represents hypotheses about the true unknown function f . In a Bayesian setting, each hypothesis h defines a conditional probability distribution: $h(x) = P(f(x) = y|x, h)$ where x is the new sample to be predicted and y is the class value. The problem of predicting the value of $f(x)$ can then be viewed as computing $(f(x) = y|S, x)$. This can be rewritten as the weighted sum of all hypotheses in H shown in equation (2).

$$P(f(x) = y|x, h) = \sum_{h \in H} h(x) P(h|S) \quad (2)$$

This ensemble method can be said to consist of all the hypotheses in H , each weighted by its posterior probability $P(h|S)$. According to Bayes rule the posterior probability is proportional to the product of prior probability of h and the likelihood of the training data. This can be expressed as (3).

$$P(h|S) \propto P(S|h) P(h) \quad (3)$$

Bayesian Model Averaging (BMA) primarily addresses the statistical characterisation of ensembles discussed earlier. When the training sample is small, many hypotheses h will have significantly large posterior probabilities and the voting process can average these to diminish any remaining uncertainty about f . When the training sample is large, it is typical for only one hypothesis to produce substantial posterior probability. Thus, the ensemble effectively shrinks to contain only a single hypothesis. In complex situations where H cannot be enumerated, it may be possible to approximate the voting process by drawing a random sample of hypotheses distributed according to the posterior $P(h|S)$.

The most idealised aspect of the Bayesian rule is the prior belief $P(h)$. If $P(h)$ completely captures all the knowledge about f before the training sample S is known, by definition one cannot achieve better. In practice however, it is often difficult to construct a space H and assign a prior $P(h)$ that captures all prior knowledge adequately. It is often the case that H and indeed $P(h)$ are chosen for computational convenience and they are known to be inadequate. In such cases, the BMA is not optimal and other ensemble methods may produce better results. In particular, the Bayesian approach does not address the computational and representational problems in any significant way.

2.3.3.2 INPUT TRAINING DATA MANIPULATION

In this method, the training data is manipulated to generate multiple hypotheses. Basically, the classifier is run several times, each with a different subset of the input training samples. This method works particularly well for unstable classifiers whose output models undergo major changes in response to any change(s) in the input data. For instance, decision trees, neural networks and rule based classifiers are known to be unstable [95]–[97]. On the other hand, linear regressions, nearest neighbour and linear threshold algorithms are generally very stable [28].

A common and perhaps the most straightforward way of manipulating the input dataset is known as Bagging (derived from bootstrap aggregation [36]). On each classification run, Bagging presents the classifier with a training set

that consists of a sample of m training examples drawn randomly with replacement from the original training set of m items. Such a training set is called a bootstrap replicate of the original training set and the technique is called bootstrap aggregation. On average, each bootstrap replicate contains approximately 63.2% of the original training set with several training samples re-used multiple times.

Another manipulation method is to construct the training sets by leaving out disjoint subsets of the overall data. For example the training set can be randomly divided into 10 disjoint subsets. Then 10 overlapping training sets can be constructed by dropping out a different one of the 10 disjoint subsets. This procedure is commonly employed to construct training datasets for 10 fold cross validation, so ensembles constructed in this way are sometimes called cross validated committees [98].

A more advanced method for manipulating the training set is illustrated by the AdaBoost algorithm [94]. Like Bagging, AdaBoost manipulates the training examples to generate multiple hypotheses. AdaBoost maintains a set of weights over the training samples. In each iteration l , the classifier is invoked to minimise the weighted error on the training set, and it returns a hypothesis h_l . The weighted error of h_l is computed and applied to update the weights on the training examples. The effect of the change in weights is to place more weight on training examples that were misclassified by h_l and less weight on examples that were correctly classified. In subsequent iterations therefore, AdaBoost constructs progressively more difficult learning problems.

The ensemble classifier $h_f(x) = \sum_l w_l h_l(x)$ is constructed by a weighted vote of the individual classifiers. Each classifier is weighted by w_l according to its accuracy on the weighted training set that it was trained on. AdaBoost is commonly applied as a stage wise algorithm for minimising a particular error function [99].

2.3.3.3 OUTPUT TARGET MANIPULATION

Another general technique for constructing a good ensemble of classifiers is to manipulate the number of classes that are fed to the classifier. Dietterich and Bakiri [100] describe a technique for multi-class data called error-correcting output coding. Consider a multiclass classification problem where the number of classes K is more than two (at least > 2). New learning problems can be constructed by randomly partitioning the K classes into two subsets A_l and B_l . The input data can then be re-labelled so that any of the original classes in set A_l are given the derived label 0 and the original classes in set B_l are given the derived label 1. This re-labelled data is then used to construct a classifier h_l . By repeating this process L times (i.e., generating different subsets A_l and B_l) one would obtain an ensemble of L classifiers h_1, \dots, h_L . Now given a new data point x , each classifier h_l will produce a class value (0 or 1). If $h_l(x) = 0$, then each class in A_l receives a vote. If $h_l(x) = 1$ then each class in B_l receives a vote. After each of the L classifiers has voted, the class with the highest number of votes is selected as the prediction of the ensemble.

This technique was found to improve the performance of both the C4.5 decision tree algorithm and the backpropagation neural network algorithm on a variety of complex classification problems [100]. In fact, Schapire [101] combined AdaBoost with error-correcting output coding to produce an ensemble classification method called AdaBoost.OC. The performance of the method was found to be significantly better than the error-correcting output coding and Bagging methods but essentially the same as another quite complex algorithm called AdaBoost.M2. The good thing about AdaBoost.OC is its implementation simplicity as it can be applied to any classifier for solving binary class problems.

2.3.3.4 INJECTING RANDOMNESS

In the backpropagation algorithm for training neural networks, the initial weights of the network are set randomly. If the algorithm is applied to the same training examples but with different initial weights, the resulting classifier can be quite different [102]. This is perhaps the most common way of generating ensembles of neural networks. However, injecting randomness into the training

set (rather than the classifier) may be more effective. This was proven in a comparative study conducted with one synthetic data set and two medical diagnosis data sets. Multiple random initial weights on neural network was compared to Bagging and 10-fold cross-validated ensembles [98]. The result shows that cross-validated ensembles worked best, Bagging second and multiple random initial weights third.

It is also easy to inject randomness into other classifiers such as the C4.5 decision tree [103][49]. The key decision of C4.5 is to choose a feature to test at each internal node in the decision tree. At each internal node C4.5 applies a criterion known as the information gain ratio to rank and order the various possible feature tests. It then chooses the top ranked feature-value test. For discrete-valued features with V values, the decision tree splits the data into V subsets depending on the value of the chosen feature. For real-valued features, the decision tree splits the data into two subsets, depending on whether the value of the chosen feature is above or below a chosen threshold.

Raviv and Intrator [104] injected noise into the features of bootstrapped training data to train an ensemble of neural networks. They drew training samples with replacement from the original training data during training. Basically, the x values of each training sample are perturbed by adding Gaussian noise to the input features and this method led to some improvement.

2.3.3.5 INPUT FEATURE MANIPULATION

Another general technique for generating multiple classifiers is to manipulate the set of input features available for classification. The process (commonly known as feature selection) is a very important part of data pre-processing in machine learning [86], [105], [106] and statistical pattern recognition [107]–[110]. Researchers are often faced with data having hundreds or thousands of features, some of which are irrelevant to the problem at hand. Running a classification task with all the features can result in a deteriorating performance, as the classifier can get stuck trying to figure out which features are useful and which are not. Therefore, feature selection is often employed as a preliminary step, to select a subset of the input data that contain useful

features. In addition, feature selection tends to reduce the dimensionality of the feature space, avoiding the well-known curse of dimensionality [108].

A major disadvantage of feature selection is that some features that may seem less important, and are thus discarded, may bear valuable information. It seems a bit of a waste to throw away such information that could possibly in some way contribute to improving classifier performance. This is where ensembles come into play by simply partitioning the input features among the individual classifiers in the ensemble. Hence, no information is discarded. Rather, all the available information in the training set are utilised whilst making sure that no single classifier is overloaded with unnecessary features.

Initial implementations of feature selected ensembles used random or grouped features for training classifiers. For instance, Liao and Moody [111] proposed a technique called Input Feature Grouping. The idea was to group the input features into clusters based on their mutual information, such that features in each group are greatly correlated to each other, and are as little correlated with features in other groups as possible. Each member classifier of the ensemble is then trained on a given feature cluster. Liao and Moody used a hierarchical clustering algorithm [112] to cluster the input features.

Tumer and Oza [113][114] presented a similar approach but the grouping was based on the class values. Basically, for a classification problem with y class labels, it constructs y classifier models. Each model is given a subset of the input features, such that these features are the most correlated with that class. The individual classifier model outputs are averaged to produce the ensemble results.

In an image analysis problem, Cherkauer [115] trained an ensemble of 32 neural networks of four different sizes, based on 8 different subsets out of 119 available input features. The input feature subsets were selected (by hand) to group together features that were based on different image processing operations. The resulting ensemble classifier was able to match the performance of human experts. Similarly, Stamatatos and Widmer [116] used

multiple SVMs successfully, each trained using grouped feature subsets for music performer recognition.

On the contrary, Tumer and Ghosh [117] applied a similar technique to a sonar dataset with 25 input features. They grouped features with similar characteristics and discarded those that did not fit into any group. The results show that deleting a few of the input features hurt the performance of the individual classifiers so much that the voted ensemble did not perform very well. Obviously, this strategy only works when the discarded input features are highly redundant.

Subsequently, researchers started to implement the grouping strategy with random selection so that none of the input features is discarded. For instance, Ho [118][119] implemented a technique called Random Subspace Method using C4.5 decision trees [120] as the base classifier. Subsets of the features were randomly selected to train various C4.5 models. At each run, half of the total number of features was selected and a decision forest was grown up to 100 decision trees. This technique produced better performance than bagging, boosting, and single tree prediction models.

Other researchers have implemented similar concepts with systematic manipulation to the input data. Among them, Bay [121] who applied random feature selection to nearest neighbour classifiers with two sampling functions: sampling with replacement and sampling without replacement. In sampling with replacement, a given feature can be replicated within the same classifier model. In sampling without replacement, however, a given feature cannot be assigned more than once to the same model.

It is very clear that the methods discussed so far are very similar in that they assign features to each individual classifier model randomly or through some form of grouping. However, further strategies have been developed that use more sophisticated selection processes. Among them, Alkoot and Kittler [122] who proposed three methodical approaches for building ensembles: the parallel system, the serial system, and the optimised conventional system. In the parallel system, the member classifiers are allowed in turns, to take one of

many features such that the overall ensemble performance is optimised on a validation set. In the serial system, the first classifier is allowed to take all the features that achieve the maximum ensemble accuracy on the validation set. If some features remain, a second expert is used, and so on. The optimised conventional system builds each expert independently, and features are added and/or deleted from the ensemble as long as the ensemble increases performance.

Günter and Bunke [123] proposed an ensemble creation technique based on two well-known feature selection algorithms: floating sequential forward and backward search algorithms [124]. In this approach, each classifier is given a well performing set of features using any of the two feature selection algorithms. Opitz [125] implemented a similar concept using a genetic algorithm to search and select the most diverse sets of feature subsets for the ensemble. Other researcher who used a genetic algorithm include Guerra-Salcedo and Whitley [126] who applied the CHC genetic search algorithm [127] to two table-based classifiers, namely KMA [10] and Euclidean Decision Tables (EDT) [128]. Oliveira et al. [129] also used a genetic search algorithm with a hierarchical two-phase approach to ensemble creation. In the first phase, a set of good prediction models are generated using Multi-Objective Genetic Algorithm (MOGA) search [130]. The second phase searches through the space created by the different combinations of these good prediction models, again using MOGA, to find the best possible combination.

2.4 SUMMARY

This chapter describes diabetes along with the various types, diagnosis and effects they have on patients. Traditional management strategies were explained with highlights to their weaknesses as well as the challenges involved in diabetes management. The middle section of the chapter presents an evidence based review about two broad methods with data-driven capabilities that can be adapted to develop healthcare tools. This looks at the type of data required to develop diabetes models and also their accessibility.

Conclusions were drawn based on the evidence, that machine learning approach is more appropriate for the type of problem addressed in this thesis.

Further discussions highlighted some fundamental reasons why single classifier models fail, and the potentials available through ensembles to eliminate the shortcomings. Precisely, single classifiers fail due to statistical, computational and representational problems discussed in section 2.3.2. Ensembles have the potential to overcome these problems if the constituent classifiers are diverse. Indeed, majority of the ensemble methods reviewed in this chapter had manipulated either input training data or the class label to train variations of a single classifier. The method proposed in this thesis is intended to probe further in this direction, by training heterogeneous classifiers rather than variations of a single classifier. To ensure optimum accuracy is achieved with each classifier, training would be conducted with a subset of the full dataset that leads to optimum performance. Feature selection would be used to select the subsets for each classifier. The descriptive part of the experiment is provided in Chapter 3 to aid full understanding of the method presented in Chapter 4.

3.1 INTRODUCTION

This chapter presents a detailed description of the technical components used to design the ensemble method implemented in this thesis. The idea is to provide the reader with detailed information to aid full understanding of the methodology in Chapter 4. Concise descriptions of the ensemble member classifiers are presented in section 3.2 to highlight their operational properties. Brief description of the experimental data is provided in section 3.3 along with pre-processing activities to transform the data into useable format. Classifier training and performance evaluation methods form the concluding part of this chapter.

3.2 ENSEMBLE MEMBER CLASSIFIERS

To construct the ensembles proposed in this thesis, five heterogeneous classifiers were employed as base learners – Sequential Minimal Optimisation (SMO), Radial Basis Function (RBF) network, C4.5 decision tree, Naïve Bayes and RIPPER. The classifiers are purposefully selected, to represent the five broad families of machine learning algorithms as shown in Table 3.1. The idea is to overcome the limited diversity issue that may exist with just using variations of a single classifier. For instance, classifiers such as neural networks and C4.5 decision trees are often used to construct a variety of ensembles due to their sensitivity to change(s) in the dataset. However, diversity in such situation is limited to data manipulation [33]. In other words, the classifier maintains its operational characteristics, and so errors and biases are restricted to its predictive power. This is likely to affect the ensemble accuracy, especially if the classifier has some weaknesses that restrict its ability to classify the data.

Consider a colour blind person who has to decide on the car to buy, on the basis of different properties such as size, colour and cost. If he decides on the

basis of cost alone, his conclusion is likely to differ from further decisions based on size or colour. However, his decision involving car colour is only as good as his ability to recognise colours properly. Therefore, an aggregate of his decisions is likely to be skewed and restricted to his ability. One possible solution is to ask a friend who does not have his weakness (i.e., colour blindness) to contribute on the car colour; so that he can use this in his final judgement. This process has the potential to improve the decision making capability of the person. The ensemble method proposed in this thesis addresses similar problems. It is intended to manipulate the experimental data features, to train heterogeneous base classifiers. A brief description is provided (in the following section) of the five member classifiers used for the ensemble. This is mainly to highlight their individual operational characteristics

Table 3.1: Five broad machine learning approaches and associated algorithms considered in this chapter.

Learning algorithm	Family
Sequential Minimal Optimization (SMO)	Support Vector Machine (SVM)
Radial Basis Function (RBF) network	Artificial Neural Network (ANN)
C4.5	Decision Tree
Naïve Bayes	Bayesian
RIPPER	Rule based

.

3.2.1 SUPPORT VECTOR MACHINES (SVM)

Sequential Minimal Optimisation (SMO) belongs to the Support Vector Machine (SVM) family. SVM operation mechanism is based on the principle of structural risk minimisation, aimed at minimising the bound on the generalisation error (i.e., error made by the learning algorithm on data unseen during training) rather than minimising the mean square error (MSE) over the data set [131]. Basically, an SVM model uses an associated learning algorithm to represent each example data as points in space, mapped so that the examples of the class categories are divided by a clear gap as wide as possible [132]. New examples are then generated and mapped into that same space; then predicted to belong to a class category based on the side of the gap they appear. For instance, using the following equation (4),

$$D = \left\{ (x_i, y_i) \mid x_i \in R^p, y_i \in \{-1, 1\} \right\}_{i=1}^n \quad (4)$$

where D is the training data with a set of n points, the class label $y_i = \pm 1$ indicating the class to which the point x_i belongs and x_i is a p -dimensional vector; the SVM learning algorithm builds a model by finding the maximum-margin hyper plane (gap) that divides the points $y_i = 1$ from $y_i = -1$; making it a non-probabilistic binary linear classifier.

In addition to performing linear classification, SVMs can efficiently handle a non-linear classification problem by using kernel tricks to map implicitly. Basically, mapping their inputs into high-dimensional feature spaces (through an underlying nonlinear mapping), before applying linear classification in these mapped spaces. SVMs tend to perform well when applied to new data not included during training due to its fundamental classification principle i.e., generates and maps new examples into the relevant class. Several research studies have also found SVM to outperform competing methods in some real-world applications [133]–[135]. The SVM model examined in this thesis is based on John Platt's sequential minimal optimisation (SMO) algorithm [132].

3.2.2 ARTIFICIAL NEURAL NETWORK (ANN)

ANNs are powerful computational models capable of computing values from inputs. ANNs are inspired by an animal's central nervous systems (particularly the brain) and generally presented as systems of inter-connected neurons [136]. The ANN classifier utilised in this thesis is the Radial Basis Function (RBF), trained by a logistic regression algorithm applied to K-means clusters as basis function.

Generally, RBF networks have three layers namely, input layer, hidden layer with a non-linear RBF activation function and a linear output layer. Its training is typically a two-step process. In the first training step, the centre vector c_i is chosen from the RBF functions within the hidden layer. This can be performed in several ways such as random sampling from a set of examples. For the experiment reported in this thesis, an unsupervised method commonly known as K-means clustering was used [136]. In the second step, logistic regression is

applied and symmetric multivariate Gaussians fitted to the hidden layer's outputs. Assume that the input is a vector of real number $x \in R^n$. The output is then a scalar function of the input vector $\varphi : R^n \rightarrow R$, and is given by (5)

$$\varphi(x) = \sum_{i=1}^N w_i \phi(\|x - c_i\|) \quad (5)$$

where N is the number of neurons in the hidden layer, c_i is the centre vector for neuron i , and w_i is the weight of neuron i in the linear output neuron. Functions that depend only on the distance from a centre vector are radially symmetric about that vector, hence the name radial basis function. For the experiment reported in this thesis, all inputs are connected to each hidden neuron using the Euclidean distance and the radial basis function is a Gaussian represented as (6).

$$\phi(\|x - c_i\|) = \exp[-\beta\|x - c_i\|^2] \quad (6)$$

The input neurons correspond to the number of features in the dataset, with one output neuron. The number of hidden layer neurons was tuned with cross validation during training for optimal accuracy. Therefore, parameters w_i , c_i and β are determined in a manner that optimizes the fit between φ and the training data. Only one output neuron was used. Like other machine learning techniques that learn from data, ANN has been used to perform a wide variety of tasks that are difficult using ordinary rule-based methods [127], [128].

3.2.3 DECISION (CLASSIFICATION) TREES

In machine learning, Classification or Decision trees is a classifier characterised by repetitive partition of the instance space [96], [137]. Generally, classification trees consist of several (non-leaf) nodes connected to the leaf nodes. The line connecting two nodes is called an edge (or a branch) which specifies a feature condition that splits data into subsequent nodes. A node that has no incoming branch is called non-leaf because it signifies the root. It starts at the topmost position and may have zero or more outgoing edges. An internal or test node has just one incoming branch and two or more

outgoing branches. All other nodes are called leaves (also known as terminal or decision nodes).

For example, given a data instance modelled as a vector of many features, to be classified into one of two classes; the decision tree grows incrementally downward by splitting the data instance into smaller and smaller subsets. A subset is known as a node and the first few nodes at the top of the tree are essentially the features that contribute to the most information gain from the class. The mapping continues with each internal node splitting into more sub-spaces according to a certain discrete function of the input feature value. The branch connecting each split node would specify the feature condition (test) that splits them into subsequent nodes. Each test considers a single feature, such that the sub-space is partitioned according to the feature's value. Each leaf is then assigned to one group representing the most appropriate class value, or a probability distribution over the classes. The structure of this decision tree is depicted in Figure 3.1.

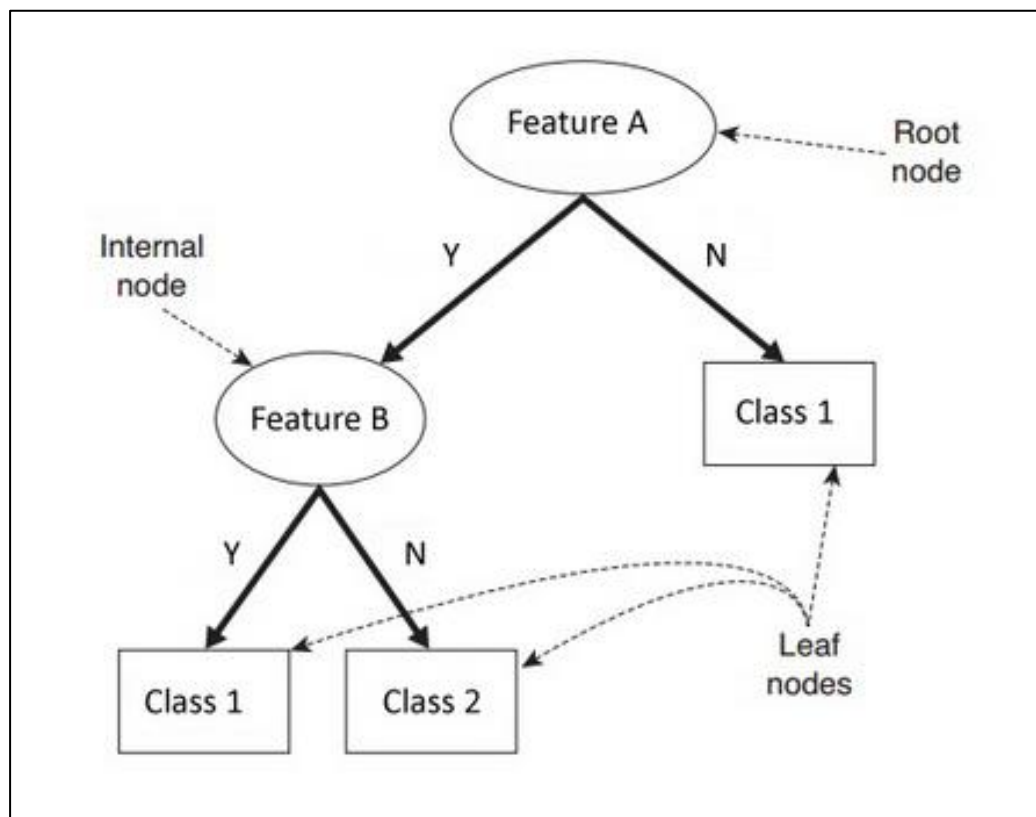


Figure 3.1: Simple Decision tree structure showing the root, internal and leaf nodes.

In this thesis, we examined the C4.5 algorithm developed by Quinlan [120]. C4.5 uses the concept of information entropy to build decision trees from a set of training data. For instance, let the training dataset $S = s_1, s_2, \dots, s_n$ of classified samples and each sample s_i is a p -dimensional vector containing $x_{1i}, x_{2i}, \dots, x_{pi}$. The x_j values represent the features of the sample data, as well as the class in which s_i belong. This operation can be represented mathematically as (7), where entropy $H(s)$ represents the amount of uncertainty in the dataset S , (i.e., S is the current dataset for which entropy is being calculated), X is a set of classes in S and $p(x)$ is the proportion of the number of elements in class x to the number of elements in set S .

$$H(s) = - \sum_{x \in X} p(x) \log_2 p(x) \quad (7)$$

At each (non-leaf) node, C4.5 selects the feature of the data that most effectively splits its set of samples into subsets enriched in one of the leaf node classes. The splitting criterion is the difference in entropy (called normalised information gain) [138]. The attribute with the highest normalised information gain is selected to make the decision. The C4.5 algorithm then recurs on the smaller subsets. The recursion terminates when all the subsets at a node have the same value of the class variable, or when splitting no longer adds information gain to the predictions [120].

Reduced Error Pruning (REP) [139] was applied to the decision tree as this has been proven to reduce tree complexity and possible over-fitting [140]. Predictive accuracy was used as pruning operator at each stage to identify rules that yield the greatest reduction of error on the pruning set. Typically pruning operation would eliminate any node(s) or single condition/rule that does not provide additional information [108].

3.2.4 NAÏVE BAYES

Naive Bayes is a simple classification technique based on Bayes' theorem with the naïve assumption that features within a data instance are independent of each other [141]. Basically, if we have a data instance x represented as a vector

of x_1, \dots, x_n features (independent variables), to be classified into class y_j , the conditional probability according to Bayes theorem can be expressed as (8), where $p(y_j|x_1, \dots, x_n)$ is the probability of instance x being in class y_j ; $p(x_1, \dots, x_n|y_j)$ is the probability of generating instance x , given class y_j ; $p(y_j)$ is the probability of occurrence of class y_j ; and $p(x_1, \dots, x_n)$ is the probability of instance x occurring.

$$p(y_j|x_1, \dots, x_n) = \frac{p(x_1, \dots, x_n|y_j)p(y_j)}{p(x_1, \dots, x_n)} \quad (8)$$

Consider a school with 100 students, where 60% are boys and 40% are girls. The girls wear trousers or skirts in equal numbers and the boys all wear trousers. If an observer sees a random student from a distance who is wearing trousers, what is the probability that this student is a girl? Using equation (8),

1. The probability of the student being a girl, $p(G)$ is 0.4, since the school has 40% girls.
2. The probability of the student not being a girl is (i.e., a boy), $p(B)$ is 0.6, since the school has 60% boys.
3. The probability of the student wearing trousers given that the student is a girl, $p(T|G)$ is 0.5 since they are likely to wear skirt or trouser.
4. The probability of the student wearing trousers given that the student is not a girl, $p(T|B)$ is 1 since all boys wear trouser.
5. The probability of a randomly selected student wearing trousers regardless of any other information $p(T) = p(T|G) p(G) + p(T|B) p(B)$

By substituting these values, equation (8) can be re-written as (9).

$$p(G|T) = \frac{p(T|G) p(G)}{p(T)} = \frac{0.5 \times 0.4}{0.8} = 0.25 \quad (9)$$

Therefore, Bayes' interpretation is that out of the hundred students from the school (60 boys and 40 girls), the observed student is one of 80 who wear trouser (60 boys and 20 girls). Since $20/80 = 1/4$ of these are girls, the probability that the student in trousers is a girl is $1/4$.

3.2.5 ASSOCIATION RULE LEARNING

Association rule classifiers and Decision trees use similar classification principles. A propositional rule learning algorithm will be examined using, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), proposed by Cohen [142]. It uses the concept of association rules with reduced error pruning (REP), a very common and effective technique found in decision tree algorithms. The RIPPER algorithm is illustrated in Figure 3.2.

```
Input      : Training dataset  $D$  with  $n$  instances and  $a$  attributes
Output     : Ruleset

Begin
  sort classes in the order of least prevalent class to the most
  prevalent class.
  create a new rule set
  while iterating from the least to most prevalent class
    split  $D$  into  $D_{pos}$  and  $D_{neg}$ 
    while  $D_{pos}$  is not empty
      split  $D_{pos}$  and  $D_{neg}$  into growing subsets  $G_{pos}$  and  $G_{neg}$ ; and
      pruning subsets  $P_{pos}$  and  $P_{neg}$ 
      create and prune a new rule
      if the error rate of the new rule is very large then
        end while
      else
        add new rule to rule set
        compute the total description length  $l$ 
        if  $l > d$  then
          end while
        end while
      end while
    end while
  end while
end
```

Figure 3.2: RIPPER algorithm (adapted from [142])

3.3 EXPERIMENTAL DATA

The literature review indicated that simplistic risk assessment models were deemed unsuitable for predicting diabetes onset, due to lack of domain knowledge caused by limited (and often superficial) data [16]. The experimental data described in this section is intended to overcome the knowledge deficiency issue. The data obtained from UCI Machine Learning

Repository [143], originates from a national study (called index examination) conducted on the Pima Indian population in the 1960s [144]. Although the data involved a different population, and possibly not representative of the UK population; the overall experiment sets the context as to how similar data from the NHS could be utilised to identify at an early stage, those at increased risk of diabetes, thus reducing the number of undiagnosed cases.

The Pima Indian data was obtained through a standardised health check conducted every two years, in which community residents over 5 years of age are tested for diabetes. However, only a fraction of the original data consisting of female subjects aged 21 or above was made available in the UCI database. The data consists of 768 samples, each defined as a row vector with eight features and a class value (i.e., negative or positive). The class value was determined by selecting one examination per subject that revealed a negative test result for diabetes and met one of the following two criteria:

1. Diabetes was diagnosed within five years of the examination
2. Diagnosis test performed five years later was negative

Of the samples, 500 tested negative and the rest ($n = 268$) tested positive over the 5 year period. Feature characteristics of the sample data are shown in Table 3.2 and full description of the data is provided in Appendix A. The source did not disclose experimental evidence that led to the selected features or indeed the total number of features available in the original database. In medical science, this decision is often based on expert knowledge drawn from empirical evidence.

Table 3.2: Characteristics of the Pima diabetes dataset from UCI database

Features	Min	Max	Mean \pm SD
No of times pregnant	0	17	3.8 ± 3.4
Fasting plasma glucose	0	199	120.9 ± 32
Diastolic blood pressure	0	122	69.1 ± 19.4
Triceps skin fold	0	99	20.5 ± 16
2-hr Serum Insulin	0	846	79.8 ± 115.2
Body mass index	0	67.1	32 ± 7.9
Pedigree function	0.1	2.4	0.5 ± 0.3
Age	21	81	33.2 ± 11.8

3.3.1.1 DATA PRE-PROCESSING

Some abnormalities were evident in the data presented in Table 3.2. For instance, a person is considered dead if their blood pressure is zero. Such abnormality in the dataset could be due to missing values or human error which is common in real life examples. It is also clear that the class categories were not equally represented in the experimental data (i.e., 500 negative : 268 positive instances). Again, this is a common situation in real life example, such as the UK where the number of diabetes cases is significantly lower than non-diabetics. To address these issues, two pre-processing operations are applied as shown in Figure 3.3.

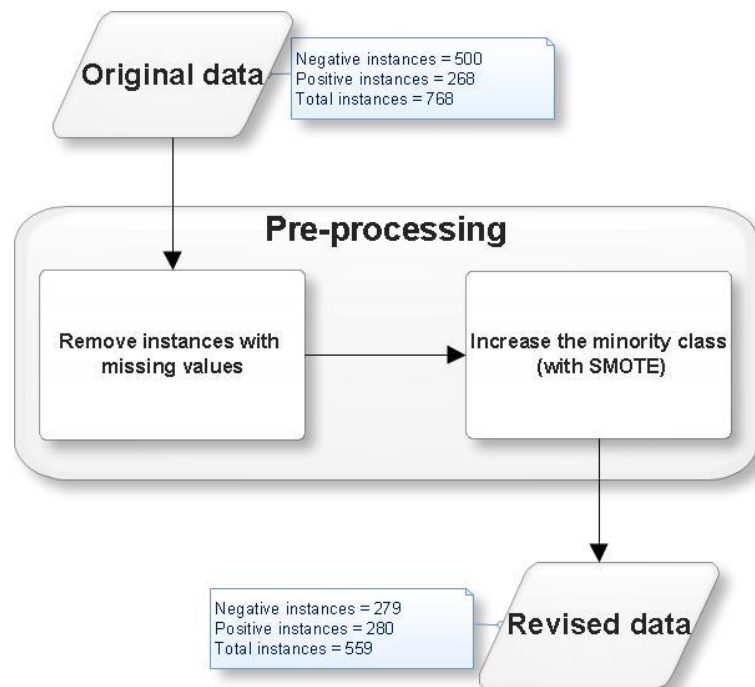


Figure 3.3: Data pre-processing operations applied on the original dataset

For the missing values, all samples with value ‘0’ are eliminated in any of the eight features except ‘No of times pregnant’. We assumed that subjects with ‘0’ value for this feature have never been pregnant. As a result, the total data sample was reduced to 419 of which 279 tested negative and 140 tested positive. To ensure unbiased estimates of prediction during experiment, it is important to address the issue of class imbalance in the dataset. A number of approaches have been proposed that could solve this issue. Among them, Pazzani et al. [145] and Domingos [146] who proposed a method that assigns

distinct costs to training examples. Other researchers [147]–[150] addressed the issue by re-sampling the original dataset, either by oversampling the minority class and/or under-sampling the majority class with replacement. Despite their efforts, these approaches have been noted not to improve minority class recognition.

In this thesis, we adopted an approach by Chawla et al. [151] commonly known as SMOTE (acronym for Synthetic Minority Over-sampling Technique). The technique blends under-sampling of the majority class with a special form of over-sampling the minority class. In the SMOTE algorithm, synthetic examples are generated by operating in feature space of the sample dataset. This is achieved by taking each minority class sample and introducing synthetic examples along the line segments joining any or all of the k minority class nearest neighbours. Detailed description of SMOTE algorithm is provided in Appendix A.2. It is important to note that the original version of SMOTE was implemented in this experiment, that uses only five nearest neighbours. Neighbours from the k (five) nearest neighbours are randomly selected based on the amount of over-sampling required. For example, if the amount of over-sampling needed is 300%, only three neighbours are selected and one sample is generated in the direction of each. We adopted this approach to increase the minority class (i.e., positive instances) within the revised dataset by 100%, thus only one neighbour was chosen for each data sample. As a result, a better balance of 279 negative and 280 positive instances is obtained. The feature characteristics of the revised dataset are shown in Table 3.3.

Table 3.3: Characteristics of the revised dataset obtained from the Pima diabetes data

Features	Min	Max	Mean	Std. Dev
No of times pregnant	0	17	3.6	3.3
Fasting plasma glucose	56	198	127.7	31.7
Diastolic blood pressure	24	110	71.8	12.2
Triceps skin fold	7	63	30	10
2-hr Serum Insulin	14	846	164.9	118.1
Body mass index	18.2	67.1	33.4	6.7
Pedigree function	0.1	2.4	0.5	0.3
Age	21	81	33.2	11.8

3.4 CLASSIFIER TRAINING METHOD

A number of methods exist for training classifiers and the ultimate goal is to measure performance. The general concept is to train the classifier using a set of data and test the resultant model on a separate dataset not used during training. However, there is often limited data samples available (as is the case in this research) so maximising data usage becomes very important. To maximise the original data, $k - fold$ cross-validation [152] was applied, where $k = 10$. In general k remains an unfixed parameter but 10-fold cross-validation is the most commonly used [153].

In $k - fold$ cross-validation, the original sample is randomly partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation set for testing the model, and the remaining $k - 1$ subsamples are used as training data. The cross-validation process is then repeated k times, with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged to produce a single estimation as shown in Figure 3.4.



Figure 3.4: Visual representation of 10-fold cross validation method (Source: [154])

The advantage of this method over others is that all observations are used for both training and validation, and each observation is used for validation exactly once. The method is proven to be statistically better than other similar methods in evaluating classifier performance [155], especially when the data is small [156]. A brief comparison with other methods is shown in Table 3.4.

Table 3.4: Comparing k-fold cross-validation to other methods

Methods	End product	Comments
Re-substitution – all available data is used for both training and testing	Optimal classifier in the sense that all available data was used for modelling	Using the same data for training and testing generates biased estimate of error rate. Not recommended unless the training set is very large.
Data Partition (Holdout) – available data is divided into two groups for training and testing. Often the training set is twice the size of the test set.	Sub-optimal classifier in the sense that it uses only part of available data for training.	Testing suffers from a small sample but result estimate of error rate is unbiased.
Cross-validation – total data sample N is divided into n equal sizes to train n different classifiers, each using $n - 1$ group and holding out each of the groups for testing.	Classifier is very close to optimal in the sense that all samples get used for both training and testing.	The result is unbiased because for each of the n classifiers, the hold out group is tested and the n test results are averaged.
Jack-knife (Leave-out-one) – similar to cross-validation but $n = N$.	Classifier is very close to optimal in the sense that all samples get used for both training and testing.	The result is unbiased but the approach is computationally intensive (slow).
Bootstrap – training set is randomly selected from N samples using replacement (i.e., sample can be selected more than once). The process is repeated many times.	Classifier is never optimal in the sense that training and testing datasets are randomly selected.	The result estimate is unbiased but the method is very computationally intensive. Recommended when only few samples are available.

3.5 PERFORMANCE EVALUATION

The performance of machine learning classifier is typically evaluated; using values from contingency table, commonly known as confusion matrix (see Figure 3.5). The figure displays multivariate frequency distribution of the class variables. The rows represent the Predicted class while the columns represent the Actual/True class. True positives (TP) and true negatives (TN) denote the correct classifications of positive examples and the correct classifications of negative examples respectively. Similarly, false positives (FP) represent negative examples incorrectly classified into positive class while false negatives (FN) represent the positive examples incorrectly classified into negative class.

		True class		Row total
		p	n	
Predicted class	y	True Positive (TP)	False Positive (FP)	Y
	n	False Negative (FN)	True Negative (TN)	N
Column total		P	N	

Figure 3.5: Simple confusion matrix or contingency table

Based on the contingency table, several measurements can be obtained to evaluate classifier performance as shown in Figure 3.6. However, sensitivity, specificity and accuracy are the most widely used [157], particularly when describing medical data classification [158]. Thus, for the task of predicting diabetes discussed in this research, these metrics would be used to determine how well (or not) the classifiers performed.

Accuracy measures the total number of correct predictions (i.e., both positive and negative). It is measured by adding TP and TN from the contingency table and dividing the value by the total number of predictions made. Unlike accuracy where the overall correct prediction is measured as an entity, sensitivity (also known as True Positive Rate) only measures the proportion of positives that are correctly identified as such while specificity measures the proportion of negatives that are correctly identified as such. In other words, sensitivity evaluates how good the classifier is at detecting those who are at risk of developing diabetes in five years' time, while specificity estimates how likely individuals without diabetes risk can be correctly ruled out.

Sensitivity or True Positive Rate (TPR):	$TPR = \frac{TP}{P}$
Specificity or True Negative Rate (TNR)	$TNR = \frac{TN}{N}$
Precision or Positive Predictive Value (PPV)	$PPV = \frac{TP}{Y}$
Negative Predictive Value (NPV)	$NPV = \frac{TN}{\bar{N}}$
Fall-out or False Positive Rate (FPR)	$FPR = \frac{FP}{N}$
False Discovery Rate (FDR)	$FDR = \frac{FP}{Y} = 1 - PPV$
Miss Rate or False Negative Rate (FNR)	$FNR = \frac{FN}{P}$
Accuracy (ACC)	$ACC = \frac{(TP + TN)}{(P + N)}$
F Score (F1) or Harmonic mean of Precision & Sensitivity	$F1 = \frac{2TP}{(2TP + FP + FN)}$
Mathews Correlation Coefficient (MCC)	$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$
<u>Informedness</u>	$Informedness = Sensitivity + Specificity - 1$
<u>Markedness</u>	$Markedness = Precision + NPV - 1$
<p><u>Nb</u>: ROC is a graph in which TPR is plotted on the Y axis and FPR on the X axis. Therefore <u>aROC</u> represents the area under the ROC.</p>	

Figure 3.6: Common performance metrics derived from a confusion matrix (Source: [157], [159]).

For any classification experiment, there is often a trade-off between sensitivity and specificity. This trade-off can be represented graphically as a Receiver Operating Characteristic (ROC) curve [160]. On a ROC curve the Y axis represents the sensitivity or True Positive Rate (TPR) of a classifier and the X axis represents the fall-out or False Positive Rate (FPR). Mathematical representation of both TPR and FPR can be seen in Figure 3.6. The ideal point on a ROC curve would be (0,100), which means that all positive examples are classified correctly and no negative examples are misclassified as positive. In cases where the ROC curves of two or more classifiers intersect, area under the

ROC (AUC) can be used to establish a dominance relationship between the classifiers [161]. Therefore, AUC performance is also considered in this research (in addition to accuracy, sensitivity and specificity).

3.6 SUMMARY

Five well known classifiers are described in this chapter to highlight their operational differences in making predictions on unseen data. It is believed that their individual biases would introduce the much needed diversity to improve performance at ensemble level. With regards to the experimental data, three issues were noted. Some samples have missing values and these were removed. There is also an issue with the class imbalance which was resolved using SMOTE algorithm. Although a better balance was obtained, issues regarding data size and class severability still remain. This was taken into account in the method implemented in Chapter 4. By applying feature selection, it is believed that the adverse effect caused by features with little or no information gain would be reduced. Also the training method (10-fold cross validation) is known to maximise the training set when there is data shortage.

4.1 INTRODUCTION

This chapter presents an ensemble-based experimental design using five heterogeneous classifiers trained on feature selected subset of the original dataset. The task is to predict the onset of diabetes. It was noted within the literature that accuracy and diversity are the two vital requirements to achieve good ensembles [27]. Single classifiers such as neural network and C4.5 decision trees are known to produce diverse models due to their sensitivity to change(s) in the dataset. However, diversity (i.e., individual bias) in such situations are limited to data manipulation only [33].

A number of methods were discussed in section 2.3.3 for manipulating data and selecting the features that lead to optimum results. However, it is fair to say that majority of the methods used random assignment of features or some form of feature grouping. It is believed that improvement can be achieved by utilising enhanced statistical feature assignment techniques applied to heterogeneous base classifiers. Therefore, the method presented in this Chapter exploits diversity in form of heterogeneous base classifiers. To ensure optimum performance, each classifier is trained with specific feature subset of the training data that leads to optimum accuracy. The approach is described explicitly in section 4.1. A concise description of the feature selection approach is presented in section 4.2.1; and the meta-classification approach is presented in section 4.2.2.

4.2 DESIGN AND IMPLEMENTATION

Five classifiers (described in Chapter 3) are employed as base learners namely: Sequential Minimal Optimization (SMO), Radial Basis Function (RBF), C4.5 decision tree, Naïve Bayes (NB) and Repeated Incremental Pruning to Produce Error Reduction (RIPPER). Each classifier is trained with a subset of the full dataset selected with best-first search algorithm [162]. Outputs from the

classifiers are used as input to train a K-Nearest Neighbour (K-NN) [167] (meta-classifier), in order to make a final prediction. All possible combinations of the five classifiers are explored, using both the full training dataset and feature selected subsets. It is expected that individual biases of the classifiers would introduce diversity, and the induced feature subsets would improve accuracy; ultimately leading to construction of good ensembles. To maximise the modest data size available for this experiment, 10-fold cross validation was used during training. The experimental process is shown in Figure 4.1.

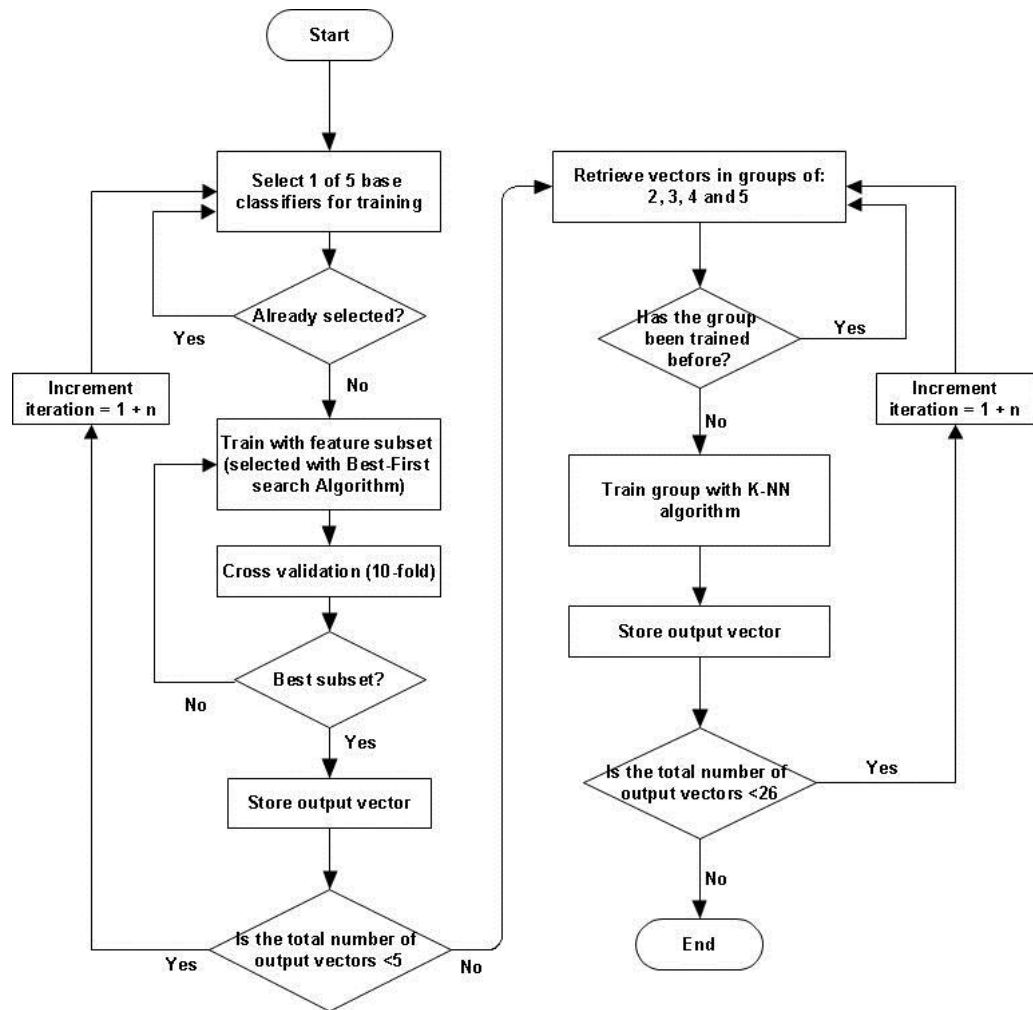


Figure 4.1: Experimental process of the base training feature selected subsets and ensemble training with K-NN algorithm.

The entire process of the ensemble method can be divided into the following three phases:

- i. Feature selection to partition the original dataset into various subsets for each classifier. Selection is done with Best-first search algorithm;

- ii. Classifier training with 10-fold cross-validation, to measure classification accuracy. Each classifier is used to validate the feature subset selected with Best-first search and the subset that leads to optimum accuracy is retained;
- iii. Training at ensemble level with K-NN algorithm. Results obtained from each classifier are used as input to train K-NN algorithm in all possible combinations; order to make a final prediction.

In a nutshell, all the five base classifiers are used to estimate the merits of the features selected with Best-first search algorithm. This is done by conducting a search through the feature space with best-first search algorithm [166] and validating the eligible feature combinations with the classifier accuracy. All the available data ($n = 559$) was used during this process. The idea is to identify the best feature subset for each classifier. To maximise the use of data, the subsets are validated by applying 10-fold cross-validation during classifier training. Detailed description of the feature selection and cross validation process is shown in Figure 4.2.

To construct the ensemble, stacked generalisation strategy (commonly known as stacking) was employed. This involves training the predictions of two or more classifiers on a given dataset, with an independent or meta-classifier. Each of the output vectors from the pool of five base learners were applied in all possible combinations (i.e., pairs, then in threes, fours and all five) to train several ensemble models. K-NN algorithm was used as the meta-classifier. By exploiting outputs from the five base classifiers in all possible combinations, a total of 26 ensemble models were trained. The results are compared to identify the ensemble with the highest performance, using predictive accuracy. Sensitivity, specificity and Receiver Operative Curve (ROC) metrics were also measured and analysed to highlight their significance in the experiment.

In the next sections, in-depth discussion is provided for the feature selection and stacking approach implemented. This is intended to highlight their importance towards the ensemble method implemented. Where necessary, references are made to other generic methods to justify our approach.

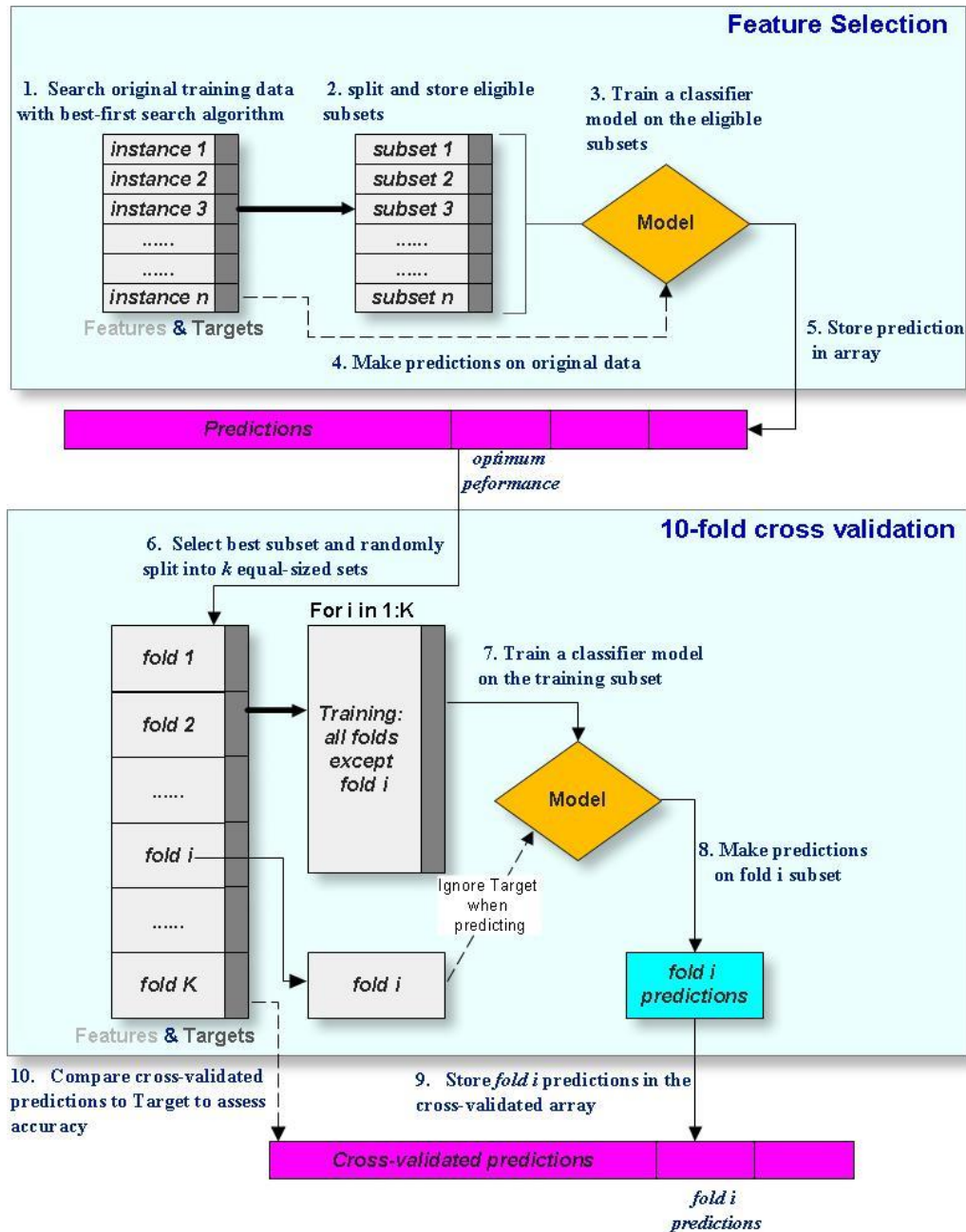


Figure 4.2: Detailed diagram of feature selection (with Best-First search) and 10-fold cross validation

4.2.1 FEATURE SELECTION APPROACH

The feature selection method adopted in this research wraps each classifier up in a feature search algorithm [166] to select the best subset for each classifier. Unlike most approaches where features are evaluated individually and independent of the classifier, the approach adopted in this research uses the classifier together with the search algorithm to induce the best feature subset for the classifier.

Consider an illustrative example where the sample data contains ten features. A search algorithm applied independent of classifier would use an evaluation function that relies solely on properties of the data to rank each feature in terms of effectiveness. If the top five features {1, 2, 3, 4, 5} are selected and the rest discarded, there is no guarantee that this would lead to better performance. One or more of the discarded features may provide some useful information when used together with the top five. Also, features 6, 7 and 8 might not be much worse than feature 5, and so could be useful to consider.

In this thesis, features are not assumed to be independent and so advantages may be gained from looking at their combined effect. Also, by using the accuracy of the classifier to evaluate the selected feature subsets, the approach presented in this thesis will pick out features which work well together for each classifier. For instance, each of the five classifiers may take different but overlapping set of features {1,2,5,7,8}, {1,3,4,6,8}, {2,4,5,6,7}, {2,3,5,6,7} and {1,2,3,4,7}. The output of the five classifier models will then be combined to train a k-NN algorithm.

The best-first search algorithm [162] is used to search the feature space. The algorithm performs a search by greedy step-wise process augmented with a backtracking facility. Basically, the algorithm explores the space of features by expanding the most promising node n chosen according to a heuristic evaluation function $f(n)$ which may depend on the promise of node n , difficulty of solving its sub-problems, quality of solution represented by node n and/or the amount of information gained [166][167]. The heuristic evaluation used in this research is focused on correlation and diversity of the selected feature subset, to gauge its merit. It takes into account the usefulness of individual features for predicting the class label as well as the level of correlation among them. This idea is motivated by the hypothesis that good subsets contain features that are highly correlated with the class but uncorrelated with each other [168].

In fact, the same principle was applied in the classical test theory where an external variable of interest is determined by a composite test (i.e., the sum or average of individual tests). Consider the procedure for awarding a bachelor's

degree in computing. Accurate prediction of a person's success is measured from a composite of modules measuring a variety of traits (e.g., ability to code, ability to write critically etc), rather than from any one individual module which measures a restricted scope of trait. In this thesis, the features are individual modules that measure the traits related to the class label (variable of interest). The heuristic can be formalised as (10)

$$Merit_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k - 1)\overline{r_{ff}}}} \quad (10)$$

where $Merit_s$ is the heuristic merit of a feature subset S containing k features, r_{cf} is the average feature – class correlation, and r_{ff} is the average feature – feature correlation. The numerator can be thought of giving an indication of how predictive a feature subset is, while the denominator indicates how much redundancy there is among them. The result is a scalar value that varies between 0 (good) and 1 (bad), so lower values indicate better merit.

In simple terms, the best-first algorithm attempts to search with the heuristic to predict how close the end of a path is to zero, and those paths which are judged to be closer are extended first. Consider a scenario where a search is initiated by expanding the first successor of the parent node. If the successor's heuristic is better than its parent, the successor is set at the front of the queue (with the parent reinserted directly behind it), and the loop restarts. However, if the parent is better, the successor is inserted into the queue (in a location determined by its heuristic value). Figure 4.3 depicts a complete feature search loop with backtracking to evaluate the remaining successors (if any) of the parent.

Best-First Algorithm

1. Put the start node s on a list called **OPEN** of unexpanded nodes.
2. If **OPEN** is empty exit with failure; no solutions exists.
3. Remove the first **OPEN** node n at which f is minimum (break ties arbitrarily), and place it on a list called **CLOSED** to be used for expanded nodes.
4. If n is a goal node, exit successfully with the solution obtained by tracing the path along the pointers from the goal back to s .
5. Otherwise expand node n , generating all its successors with pointers back to n .
6. For every successor n' on n :
 - a. Calculate $f(n')$.
 - b. if n' was neither on **OPEN** nor on **CLOSED**, add it to **OPEN**.
Attach a pointer from n' back to n .
Assign the newly computed $f(n')$ to node n' .
 - c. if n' already resided on **OPEN** or **CLOSED**, compare the newly computed $f(n')$ with the value previously assigned to n' .
If the old value is lower, discard the newly generated node.
If the new value is lower, substitute it for the old (n' now points back to n instead of to its previous predecessor).
If the matching node n' resides on **CLOSED**, move it back to **OPEN**.
7. Go to step 2.

Figure 4.3: Best-First Algorithm with greedy step-wise and backtracking facility

4.2.1.1 DIRECTION OF FEATURE SELECTION

The feature search experiment conducted in this research uses a bi-directional selection approach. This was mainly due to the greedy nature of best-first search algorithm and associated drawbacks of using single direction approach. Best-first search algorithm is known to be too greedy and prefers states that look good very early in the search.

In a forward selection approach, the algorithm starts with a preferred feature and incrementally adds in all the other features. For each step, the feature that satisfies some heuristic function is added to the current feature set, (i. e., one step of the best-first selection is performed) and the new subset evaluated with the associated classifier. The new feature is only kept if there is a notable increase in accuracy. The algorithm also verifies the possibility of improving the criterion if some feature is excluded. In this case, the worst feature is eliminated from the set by back tracking along the line of the goal node. The selection proceeds dynamically increasing and decreasing the number of features until the desired subset d is reached. Figure 4.4 gives a general

overview of forward search algorithms, expressed using the state-space representation.

```

FORWARD SEARCH
1   $Q.Insert(x_I)$  and mark  $x_I$  as visited
2  while  $Q$  not empty do
3       $x \leftarrow Q.GetFirst()$ 
4      if  $x \in X_G$ 
5          return SUCCESS
6      forall  $u \in U(x)$ 
7           $x' \leftarrow f(x, u)$ 
8          if  $x'$  not visited
9              Mark  $x'$  as visited
10              $Q.Insert(x')$ 
11          else
12              Resolve duplicate  $x'$ 
13 return FAILURE

```

Figure 4.4: A generic template for forward search (Source: [169])

At any point during the search, there will be three types of states, namely:

1. **Unvisited:** States that have not been visited yet. Initially, this is every state except x_I .
2. **Dead:** States that have been visited, and for which every possible next state has also been visited. A *next state* of x is a state x' for which there exists a $u \in U(x)$ such that $x' = f(x, u)$. In a way, these states are dead because there is nothing more that they can contribute to the search (i.e., there are no new leads that could help in finding a feasible plan).
3. **Alive:** States that have been encountered, but possibly have unvisited next states. Such states are considered alive because initially, the only alive state is x_I .

The set of alive states is stored in a priority queue Q , for which a priority function must be specified. The only significant difference between various search algorithms is the particular function used to sort Q . The illustration in Figure 4.4 assumes a First-In First-Out queue. Initially, Q contains the initial state x_I . A **while** loop is then executed, which terminates only when Q is

empty. This will only occur when the entire feature space has been explored without finding any goal states, which results in a FAILURE. In each **while** iteration, the highest ranked element x of Q is removed. If x lies in X_G , then it reports SUCCESS and terminates; otherwise, the algorithm tries applying every possible action $u \in U(x)$. For each next state $x' = f(x, u)$ it must determine whether x' is being encountered for the first time. If it is unvisited, then it is inserted into Q ; otherwise, there is no need to consider it because it must be either dead or already in Q [169].

As evident from the forward selection template Figure 4.4, the selected subset is not assessed in the context of others not included yet. This argument is illustrated with the example in Figure 4.5, in which the circles represent three features and the values within them represent the accuracy on a classifier. Feature 2 produces better accuracy by itself than either of the two other ones taken alone and will therefore be selected first by forward selection. At the next step, when it is complemented by either of the two other features, the resulting accuracy will not be as good as the one obtained jointly by the two features that were discarded at the first step.

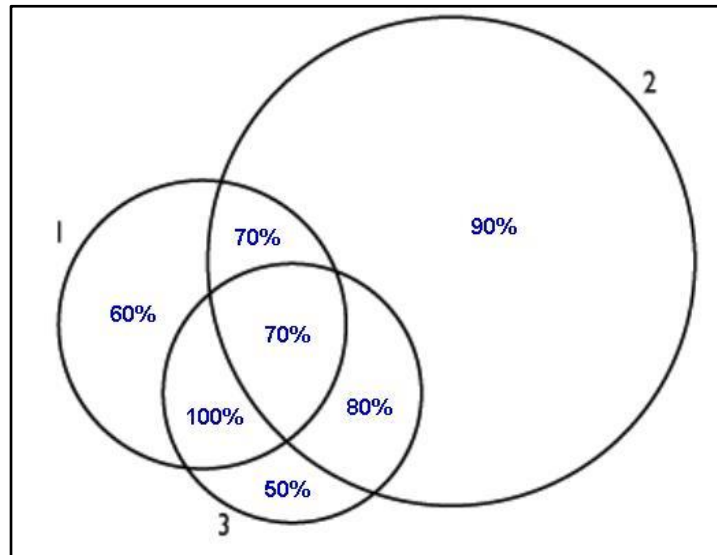


Figure 4.5: Illustration of forward and backward selection drawbacks with 3 features

On the contrary, a backward search starts with the full feature set and performs the search until the desired dimension d is reached. Therefore, it may outsmart forward selection by eliminating at the first step the feature that by itself provides the best accuracy to retain the two features that together perform best.

On the other hand, if for some reason only a single feature is required, backward elimination will have gotten rid of the feature that works best on its own.

In view of these drawbacks, a bidirectional search was utilised in the experiment reported in this thesis. Figure 4.6 shows the combination of both forward and backward search.

BIDIRECTIONAL SEARCH

```

1   $Q_I.Insert(x_I)$  and mark  $x_I$  as visited
2   $Q_G.Insert(x_G)$  and mark  $x_G$  as visited
3  while  $Q_I$  not empty and  $Q_G$  not empty do
4      if  $Q_I$  not empty
5           $x \leftarrow Q_I.GetFirst()$ 
6          if  $x = x_G$  or  $x \in Q_G$ 
7              return SUCCESS
8          forall  $u \in U(x)$ 
9               $x' \leftarrow f(x, u)$ 
10             if  $x'$  not visited
11                 Mark  $x'$  as visited
12                  $Q_I.Insert(x')$ 
13             else
14                 Resolve duplicate  $x'$ 
15         if  $Q_G$  not empty
16              $x' \leftarrow Q_G.GetFirst()$ 
17             if  $x' = x_I$  or  $x' \in Q_I$ 
18                 return SUCCESS
19             forall  $u^{-1} \in U^{-1}(x')$ 
20                  $x \leftarrow f^{-1}(x', u^{-1})$ 
21                 if  $x$  not visited
22                     Mark  $x$  as visited
23                      $Q_G.Insert(x)$ 
24                 else
25                     Resolve duplicate  $x$ 
26 return FAILURE

```

Figure 4.6: A generic template for bi-directional search (Source: [169])

One tree is grown from the initial state, and the other is grown from the goal state. The search terminates with success when the two trees meet and failure occurs if either priority queue has been exhausted. Predictive accuracy is used as performance validator at each search loop.

4.2.2 STACKED GENERALISATION

Stacked generalization (known as stacking) is a way of combining multiple models, that introduces the concept of a meta learner [170]. Unlike bagging and boosting, stacking is normally used to combine models of different types such as the one described in this thesis. The procedure is as follows:

1. Manipulate original data into 10 disjoint sets.
2. Train each base classifier on 10 – 1 sets.
3. Test the base classifiers on the hold out set; and repeat steps 1 – 3 until all the 10 sets have been used once for testing
4. Average the predictions on all the sets
5. Using the predictions from (4) as the inputs, and the correct responses as the outputs, train a higher level learner.

Traditionally, ensembles are often combined through voting (majority wins) or averaging the results. However, steps 1 to 4 in stacking are the same as cross-validation which makes this method more rigorous. Instead of using a winner-takes-all approach, we combined the base classifiers using k-NN as shown in Figure 4.7.

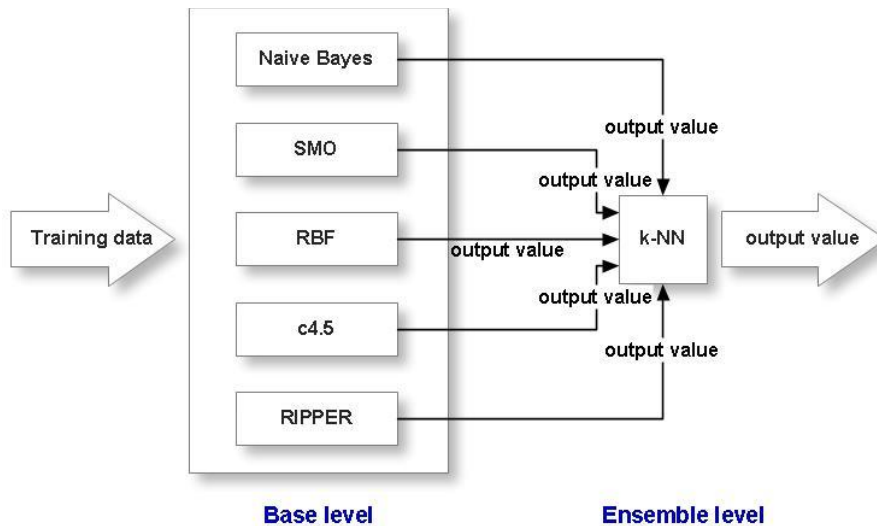


Figure 4.7: Stacked generalisation using five base learners

The nearest neighbour is a nonparametric method, where a new observation is classified based on the learning set that is closest to the new observation, with respect to the covariates used [163]–[165]. The determination of this similarity

is based on distance measures. For instance, let $L = \{(x_i, y_i), i = 1, \dots, n_L\}$ be a training set of observed data, where $y_i \in \{1, \dots, c\}$ denotes class membership and the vector $x'_i = (x_{i1}, \dots, x_{ip})$ represents the feature values. The nearest neighbour determination is based on an arbitrary distance function $d(.,.)$. So for a new observation (x, y) , the nearest neighbour $(x_{(1)}, y_{(1)})$ within the training set is determined by $d(x, x_{(1)}) = \min_i(d(x, x_i))$ and the class of the nearest neighbour $\hat{y} = y_{(1)}$, is selected as prediction for y .

For the experiment reported in this thesis, euclidian distance is used as the distance function. The Euclidean distance between two points x_i and x_j is the length of the line segment connecting them $(\overline{x_i x_j})$ [171]. Therefore, the distance function can be represented as (11), where x_j represents the j^{th} nearest neighbour of x .

$$d(x_i, x_j) = \left(\sum_{s=1}^p (x_{is} - x_{js})^2 \right)^{\frac{1}{2}} \quad (11)$$

We recognise a possible drawback of this distance measure, particularly when the class distribution is skewed. For instance, examples of a more frequent class may dominate the prediction of the new example, due to their large numbers [172]. Thus, a basic majority voting by distance may be biased by the class common among the k nearest neighbours. A common scheme to overcome this problem is to assign weight to the contributions of the neighbours, so that the nearer neighbours contribute more to the average than the more distant ones. In this thesis, the class of each of the k nearest points is multiplied by a weight proportional to the inverse of the distance from that point to the test point. In other words, each neighbour is assigned a weight $1/d$, where d is the distance to the neighbour.

4.3 SUMMARY

The ensemble method proposed in this chapter seeks to utilise the individual biases of different learning algorithms to select the best training subsets. Unlike

most approaches where features are selected individually and independent of the classifier, the approach uses a search algorithm to select the most diverse but useful features from a dataset; and subsequently validates their plausibility with the classifier for which they were selected.

Another issue addressed in this chapter is how ensemble models can be constructed to account for complexities in the training data class distribution as a result of oversampling. When faced with complex learning problems that involve highly unbalanced data sets, researchers often modify the class distribution of the training set. However, these modifications are rarely done in a systematic manner and additional measures are not considered to address the effects of any change in the distribution. In describing the proposed ensemble method, this chapter discussed in depth the oversampling method used with a clear understanding of how changes made to class distribution affects learning. In particular, explanation was provided to why the originally majority class may dominate the feature space, thereby causing undue bias in predicting new examples. This was addressed in three key strategies by optimising the training data through k-fold cross validation; personalising feature subset selection at base level through validation with the classifier; and using a weighting system at ensemble level training so that the learning set that is closest to the new observation contribute more to the average than the more distant ones.

5.1 INTRODUCTION

This chapter presents the results from the proposed ensemble method described in Chapter 4. The results are analysed with a modular approach so that individual components of the method are discussed appropriately. The performance of the classifiers at base training level with the full dataset is presented in section 5.2, followed by performance at base training level with feature selected subsets in section 5.3. Section 5.4 covers the ensemble level training with full dataset and feature selected subset. This includes a comparative study between the most accurate ensembles from both groups; to measure the impact of feature selection towards improving ensemble accuracy. The results are also compared with similar studies that used the same dataset within the literature.

5.2 BASE LEVEL PERFORMANCE WITH FULL TRAINING SET

This section presents the classifier performance at base level on the full training dataset. The results shown in Table 5.1, are intended to be a benchmark against which the ensembles would be measured, to determine if improvement was made. Detailed analysis is provided for each of the four performance metrics, to highlight their relevance to the experiment.

Table 5.1: Results of base learner training with full experimental data

Classifiers	Accuracy	Sensitivity	Specificity	AUC
Naïve Bayes	0.75	0.72	0.77	0.83
RBF	0.78	0.82	0.73	0.85
SMO	0.76	0.74	0.78	0.85
C4.5	0.77	0.81	0.74	0.79
RIPPER	0.78	0.80	0.77	0.79

It appears from Table 5.1, that the RIPPER and RBF models are the most accurate (accuracy = 78%). However, it may be argued that an accuracy value of 78% is low. There is increasing evidence that redundant features, class imbalance and skewed class distribution affects classifier accuracy [173]–

[176]. Although SMOTE is quite effective in increasing the minority class, it does not eliminate possible performance degradation in complex data situations where the classes are overlapping. SMOTE generates synthetic data based on the distance to the closest minority instance. Therefore, the generated samples may be spread across both minority and majority instances in class coupling situations, hence reducing the performance of classification. In fact, this is the case in our experimental data shown in Figure 5.1; in which the two classes overlap so much that SMOTE cannot get a good sense of the distribution. The figure depicts a 2D scatter plot in which the BMI feature is plotted against the other features within the dataset (see Appendix A.4 for scatter plots of the other features). The red data points represent negative instances while the blue data points represent positive instances.

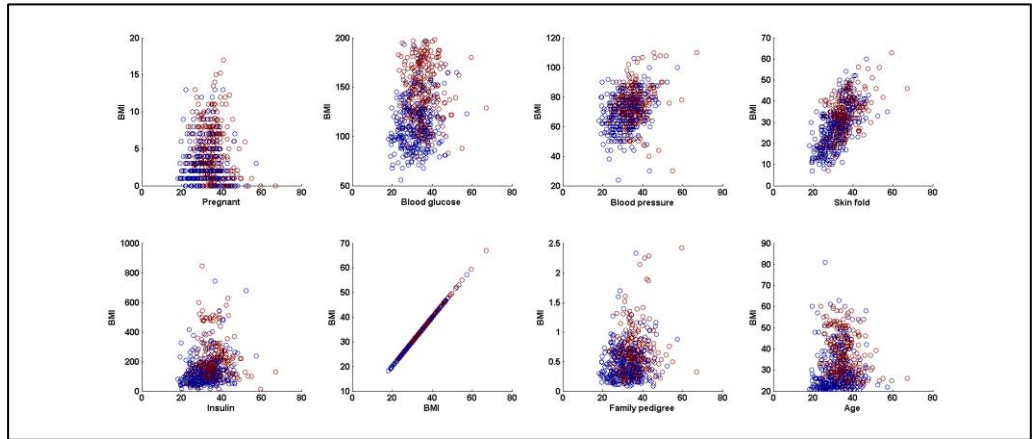


Figure 5.1: Scatter plot showing class separation and distribution between BMI and other features of the experimental dataset.

Despite the likelihood of performance degradation due to overlapping class, the accuracy obtained with the balanced dataset (Table 5.1) is considerably better than that obtained with the imbalanced data before SMOTE was applied (see Appendix A.3). That said, there has been some interesting research to modify the location and direction of synthetic data generation implemented by SMOTE algorithm. Among them, Batista et al. [177] who combined SMOTE and Tomek Links [178] to delete synthetic samples located in the area of the minority data. Ramentol et al. [179] applied the rough set theory to improve synthetic data generated by SMOTE. Han et al. [180] divided the dataset into three locations based on the amount of majority data in the nearest neighbours of minority data. Bunkhumpornpat et al. [181] focused on finding the safe area

to perform over sampling based on the ratio between the number of minority data and the nearest neighbours.

To determine the superior model between RBF and RIPPER in terms of accuracy, there is a need to look at the parameters from which the value was calculated. The numerical value of accuracy represents the proportion of both true positive and true negative in the selected population, thus assumes even class balance with equal error cost. This is not always the case in real world examples and certainly not in the research reported in this thesis where the abnormal class is disproportionately lower; and the cost of misclassifying an abnormal example as normal is much higher. Consider the binary classification of the UK population as either positive or negative in terms of diabetes. Recent estimates suggest that 4.6% of the population are affected [182], leaving 95.4% normal cases. A diabetes prediction model that classified all the majority class correctly and all the minority class wrong would give a very high accuracy of 95.4%. This result is misleading because such model (although with high accuracy) failed to identify those at risk of developing diabetes. In fact, this is the case in our experiment as shown in the contingency Table 5.2. Compared to RBF, the RIPPER model predicted more instances correctly ($TP + TN = 437$). However, predictions of the minority class are proportionately lower with the RIPPER model ($TP = 223$). The nature of the model discussed in this chapter requires a fairly high rate of correct detection in the minority class (positive) and allows for a small error rate in the majority class. This means that there is higher consequence of misclassifying a person at high risk of developing diabetes as normal.

Table 5.2: Contingency table produced at base level experiment with full training dataset

Classifier	True Positive (TP)	False Positive (FP)	True Negative (TN)	False Negative (FN)
Naïve Bayes	201	79	216	63
RBF	229	51	205	74
SMO	206	74	219	60
C4.5	226	54	207	72
RIPPER	223	57	214	65

Given that the RBF model produced relatively higher true positives ($TP = 229$) with lower false positives ($FP = 51$), it is fair to say that RBF performed

slightly better than RIPPER in terms of accuracy. This comparison can be seen more clearly in Figure 5.2.

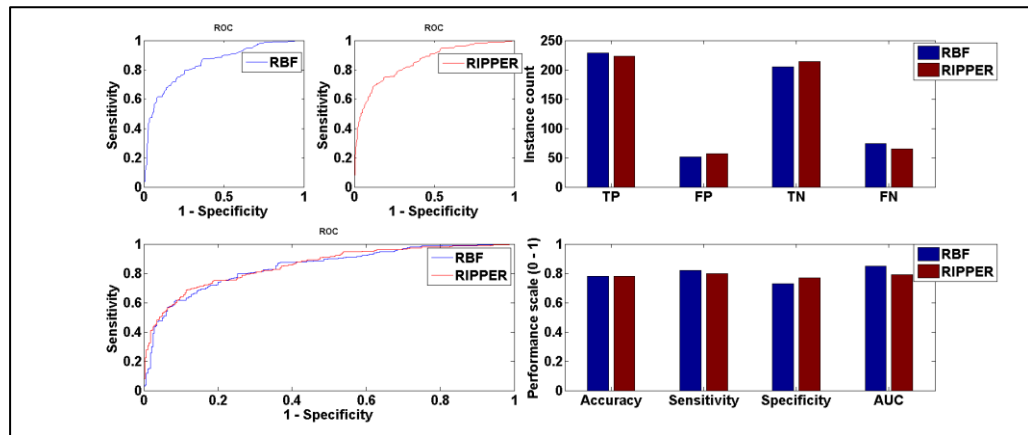


Figure 5.2: Performance comparison between RBF and RIPPER models trained on full dataset

The result in Figure 5.2 is even more interesting when we use performance metrics that disassociates the errors (or hits) that occurred within each class. From the results in Table 5.2, it is possible to derive two performance metrics that directly measure the classification performance on the positive (sensitivity) and negative (specificity) classes independently. Unlike predictive accuracy, both performance measures are prevalence-independent, as their values are inherent to the test data and not the actual prevalence in the population of interest [183]. The sensitivity value measures the percentage of positive cases correctly classified as belonging to the positive class while specificity measures the percentage of negative cases correctly classified as belonging to the negative class.

Both metrics are mostly useful in medical science where the target class (positive) is often smaller with heavy consequence if misclassified; so the trade-off between the two are considered carefully to get a good balance. Consider the results presented in Figure 5.2, where the RBF model produced a higher sensitivity, but lower specificity than the RIPPER model. From a medical view point, the RBF model could be seen as predicting based on the rare positive cases such as athletes (low specificity), in order to reduce the risk of missing those unusual cases where an active person might be at high risk of developing diabetes (high sensitivity). Although the specificity value is slightly

higher with the RIPPER model, the cost of incorrectly identifying those without risk (specificity) is lower. Therefore, decisions as to which model to choose would largely be in favour of RBF.

The trade-off between sensitivity and specificity of a classifier can be represented graphically using a receiver operating characteristic curve (ROC). The position of each point on the ROC indicates the trade-off between sensitivity and specificity and the area under the ROC (AUC) measures its discrimination to give an indication of how accurate the prediction is. For instance, consider the predictions of the RBF model in which individuals were already classified into two classes (negative or positive). If one data instance is drawn at random from each of the classes to validate the model, the patient at increased risk of developing diabetes should be classified into the positive class. The AUC is the percentage of randomly drawn pairs for which this result is true (i.e., RBF correctly classifies the two patients in the random pair). A rough guide for classifying AUC is the traditional academic point system shown in Table 5.3.

Table 5.3: A guide for classifying the Accuracy of a model using AUC (Source: [158])

AUC Range	Classification
$0.9 < \text{AUC} < 1.0$	Excellent (A)
$0.8 < \text{AUC} < 0.9$	Good (B)
$0.7 < \text{AUC} < 0.8$	Fair (C)
$0.6 < \text{AUC} < 0.7$	Poor (D)
$0.5 < \text{AUC} < 0.6$	Fail (F)

In view of this knowledge, it can be said that the RBF model ($\text{AUC} = 0.85$) was more capable of classifying instances into the correct class than the RIPPER model ($\text{AUC} = 0.79$). Furthermore, giving consideration to performance on the other metrics, the RBF model produced the best results and thus, used as benchmark for measuring the ensemble performance.

5.3 FEATURE SELECTED SUBSETS AND PERFORMANCE

This section presents the selected features for each base classifier based on best-first bi-directional search. Each base classifier was trained on the selected feature subset and performance recorded as shown in Table 5.4. The aim is to

compare the results with individual performances obtained during training with the full dataset (see Table 5.4). Detailed analysis is provided for each of the five classifiers, to highlight any improvement(s) and their relevance at ensemble level.

Table 5.4: Selected features for each classifier and performance based on the subsets

Features	Naïve Bayes	RBF	SMO	C4.5	RIPPER
No of times pregnant	∅	√	∅	∅	∅
Fasting plasma glucose	√	√	√	√	√
Diastolic blood pressure	√	√	√	√	√
Triceps skin fold	√	√	√	√	√
2-hr Serum Insulin	∅	∅	√	√	√
Body mass index	∅	∅	√	√	√
Pedigree function	√	√	√	√	√
Age	∅	√	∅	√	√
Number of features selected	4	6	6	7	7
Number of subsets evaluated	72	104	88	96	96
Merit of selected subset	0.231	0.181	0.222	0.086	0.138
Accuracy	0.77	0.79	0.76	0.78	0.78
Sensitivity	0.79	0.77	0.77	0.76	0.76
Specificity	0.75	0.80	0.75	0.80	0.81
AUC	0.84	0.85	0.84	0.80	0.80

As noted in Chapter 4, the central premise of this phase of the experiment is to remove features that are either redundant or irrelevant, without incurring much loss of information. However, it is important to note that redundant and irrelevant features are two distinct notions that must be interpreted in context. A relevant feature may be redundant in the presence of another relevant feature with which it is strongly correlated. In fact, this is evident in Table 5.4 where variations of the features are selected by each classifier; and each feature is selected at least once. This shows that features are selected based on correlation induced by individual classifier biases.

Interestingly, blood glucose and blood pressure are among the few features selected by all the five classifiers. This reinforces literature evidence that such bio markers are very important to develop robust predictive models that approach full understanding of diabetes. Two additional features within the experimental dataset (i.e., tricep skin fold and diabetes pedigree function) were also selected by all the classifiers. Their selection supports research evidence about the correlation between both features and diabetes onset. According to

Chandra et al. [184], skin fold thickness is mandatory to identify progression to diabetes. Freeman et al. [185] and Zuchinali et al. [186] also highlights the importance of tricep skin fold in predicting diabetes onset. Likewise, diabetes pedigree has been applied successfully to identify individuals at high risk of developing diabetes [187]. Diabetes pedigree function in the dataset holds information about diabetes history in relatives and the genetic relationship of those relatives to the patient. It provides a general idea of the hereditary risk the patient might have with the onset of diabetes. The results in Table 5.4, particularly the new observation highlights the benefits of feature selection to the ensembles implemented in this thesis.

The best-first search algorithm goes through the forward and backward passes, features are added or removed and subsets are evaluated based on accuracy and the heuristic described in section 4.2.1. Subset evaluation continues until a stale search condition is reached from node expansions. Therefore, the number of subsets evaluated varies with each classifier. For instance, Naïve Bayes produced stale search after 72 subsets with maximum merit of 0.231. Using the performance guide in Table 5.3, the merit of all subsets can be said to fall within the ‘good’ and ‘excellent’ range. Note that the merit value varies between 0 (good) and 1 (bad), thus Table 5.3 was interpreted backwards. It is also important to note that the merit values are classifier dependent and therefore renders cross comparison irrelevant.

Unlike the merit of subset values, cross comparison between the classifiers could be made with the other performance metrics shown in Table 5.4 (i.e., Accuracy, Sensitivity, Specificity and AUC). However, there is little value in this analysis since all the classifiers would be used at ensemble level, regardless of their individual performance. On the other hand, there is value in comparing the results for each classifier with the full training set and the feature subset, to determine if improvements were made.

To measure the differences in predictive accuracy, Mc Nemar’s test was conducted with each classifier’s predictions before and after the feature selection process. Mc Nemar’s test [188]–[190] is a non-parametric test on a 2x2 classification table to measure the difference between paired proportions.

This means that two discrete dichotomous variables with the classification data must be identified to produce 4 possible outcomes arranged in a 2×2 contingency table as shown in Table 5.5.

Table 5.5: Possible results of two classifier algorithms (Source: [189])

	Classifier B failed	Classifier B succeeded
Classifier A failed	N_{ff}	N_{fs}
Classifier A succeeded	N_{sf}	N_{ss}

N_{ff} denotes the number of times both classifiers failed to classify instances correctly and N_{ss} denotes success for both classifiers. These two values do not give much information about the classifiers' performances as they do not indicate how their performances differ. However, the other two parameters (N_{sf} and N_{fs}), shows cases where one of the classifier failed and the other succeeded indicating the performance discrepancies.

For the test analysed in this section, predicted class values are recorded and compared with true class values before and after feature selection is applied to the dataset. Classifier A represents all instances where there is a hit (i.e., true positive and true negatives) between the predicted and true class for each classifier trained on full dataset. Classifier B represents all instances where there is a hit (i.e., true positive and true negatives) between the predicted and true class for each classifier trained on feature selected subset. The difference between the proportions were calculated and expressed as a percentage with 95% confidence interval according to Sheskin [191]. The P-values are also calculated based on the cumulative binomial distribution to measure the significance of any difference in performance. When the P-value is less than the conventional 0.05, the conclusion is that there is a significant difference between the two proportions.

It is not possible to compare sensitivity, specificity and AUC values with Mc Nemar's test. This is mainly because their values are not dichotomous and therefore could not be expressed in a form suitable for Mc Nemar's test. Nonetheless, these metrics are discussed in statistical terms and comparisons made within the context of the experiment being analysed (similar to the analysis in section 5.2).

5.3.1 NAÏVE BAYES PERFORMANCE COMPARISON

This section presents the results for the Naïve Bayes classifier model. As shown in Figure 5.3, accuracy of classification B model (with selected feature subset) is marginally better than Classification A model (with full training set).

Classification A	Result_fullset	
Classification B	Result_subset	

Classification A	Classification B		
	0	1	
0	31	111	142 (25.4%)
1	98	319	417 (74.6%)
	129 (23.1%)	430 (76.9%)	559

McNemar test	
Difference	2.33%
95% CI	-2.74 to 7.39
Exact probability (binomial distribution)	
Significance	P = 0.4066

	Classification A	Classification B
True positive (count)	201	207
False negative (count)	79	73
True negative (count)	216	223
False Positive (count)	63	56
Classified correctly (count)	417	430
Classified incorrectly (count)	142	129
Accuracy (%)	75	77
Sensitivity (%)	72	74
Specificity (%)	77	80
AUC (%)	83	84

Figure 5.3: Naïve Bayes performance with full training set vs selected feature subset using Accuracy, Sensitivity, Specificity, AUC and Mc Nemar's test.

The total number of instances classified correctly is higher in classification B with reduced errors (i.e., False negatives and False positives). However, the Mc Nemar's test result shows that the accuracy difference between the two models is marginal. 74.6% of the instances are correctly classified before feature selection (*Classification A = 1*) and 76.9% are correctly classified after feature selection (*Classification B = 1*). The difference before and after feature selection is 2.33% with 95% confidence interval from -2.74% to 7.39%, which is not significant ($P=0.4066$, $n=559$). Slight improvements are also recorded for the sensitivity (2%), Specificity (3%) and AUC (1%). Visual representations of the results are shown in Figure 5.4.

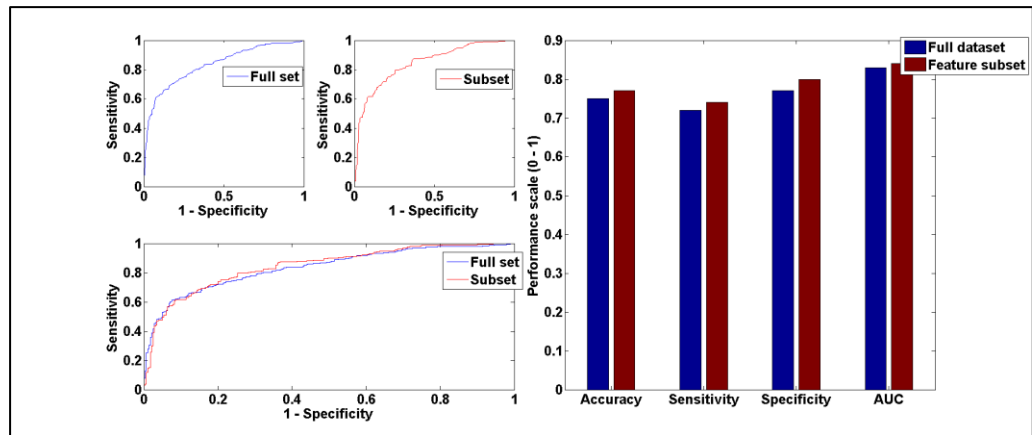


Figure 5.4: Graphic representation of Naïve Bayes performance trained on full dataset vs feature subset

5.3.2 RBF PERFORMANCE COMPARISON

Mixed results are obtained with RBF as shown in Figure 5.5. Predictive accuracy is marginally higher in classification B (1%) and the total number of instances classified correctly is higher in classification B ($n = 439$) than classification A ($n = 434$). However, the hits on true positive instances were higher in classification A ($n = 229$), compared to classification B ($n = 226$). This situation is not good at this training level but may well contribute in identifying the negative instances at ensemble level.

According to Mc Nemar's test result, the accuracy difference between the two models is highly marginal (0.89%) with 95% confidence interval from -1.40% to 3.19%, which is not significant ($P=0.5424$, $n=559$). 77.6% of the instances were correctly classified before feature selection (Classification A = 1) and 78.5% were correctly classified after feature selection (Classification B = 1).

Classification A	Result_fullset
Classification B	Result_subset

Classification A	Classification B		
	0	1	
0	101	24	125 (22.4%)
1	19	415	434 (77.6%)
	120 (21.5%)	439 (78.5%)	559

McNemar test	
Difference	0.89%
95% CI	-1.40 to 3.19
Exact probability (binomial distribution)	
Significance	P = 0.5424

	Classification A	Classification B
True positive (count)	229	226
False negative (count)	51	54
True negative (count)	205	213
False Positive (count)	74	66
Classified correctly (count)	434	439
Classified incorrectly (count)	125	120
Accuracy (%)	78	79
Sensitivity (%)	82	81
Specificity (%)	73	76
AUC (%)	85	85

Figure 5.5: RBF performance with full training set vs selected feature subset using Accuracy, Sensitivity, Specificity, AUC and Mc Nemar's test.

Given the slightly higher hits on true positives with classification A, it is not surprising that the sensitivity performance was marginally higher (1%). Nonetheless, the specificity was higher in classification B (3%) and AUC performance was tied at 85%. Visual representations of the results are shown in Figure 5.6.

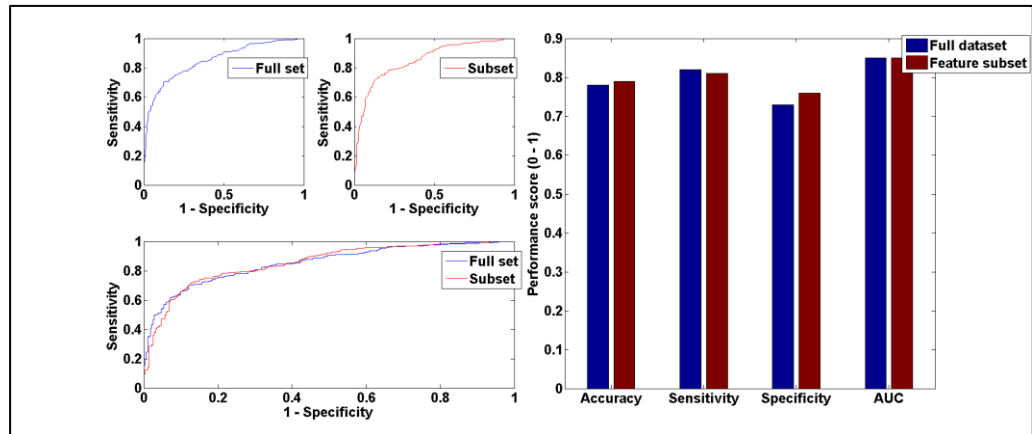


Figure 5.6: Graphic representation of RBF performance trained on full dataset vs feature subset

5.3.3 SMO PERFORMANCE COMPARISON

Performances appear very similar between the two classification experiments as shown in Figure 5.7. The same predictive accuracy value (76%) was recorded for both models. However, the total number of instances classified correctly is slightly higher in classification B ($n = 427$) than classification A ($n = 425$). Similarly, the hits on true positive instances is higher in classification B ($n = 209$), compared to classification B ($n = 206$).

Classification A	Result_fullset		
Classification B	Result_subset		

	Classification B		
Classification A	0	1	
0	112	22	134 (24.0%)
1	20	405	425 (76.0%)
	132 (23.6%)	427 (76.4%)	559

McNemar test

Difference	0.36%
95% CI	-1.91 to 2.63

Exact probability (binomial distribution)

Significance	P = 0.8776
--------------	------------

	Classification A	Classification B
True positive (count)	206	209
False negative (count)	74	71
True negative (count)	219	218
False Positive (count)	60	61
Classified correctly (count)	425	427
Classified incorrectly (count)	134	132
Accuracy (%)	76	76
Sensitivity (%)	74	75
Specificity (%)	78	78
AUC (%)	85	85

Figure 5.7: SMO performance with full training set vs selected feature subset using Accuracy, Sensitivity, Specificity, AUC and Mc Nemar's test

The Mc Nemar's test result shows that the accuracy difference between the two models is marginal. 76.0% of the instances were correctly classified before

feature selection (*Classification A = 1*) and 76.4% were correctly classified after feature selection (*Classification B = 1*). The difference before and after feature selection is 0.36% with 95% confidence interval from -1.91% to 2.63%, which is insignificant ($P=0.8776$, $n=559$).

Slight improvement was recorded for the sensitivity (1%). This may seem insignificant at this training level but may well have bigger impact at ensemble level. Visual representations of the results are shown in Figure 5.8, including specificity and AUC which were tied at 78% and 85% respectively.

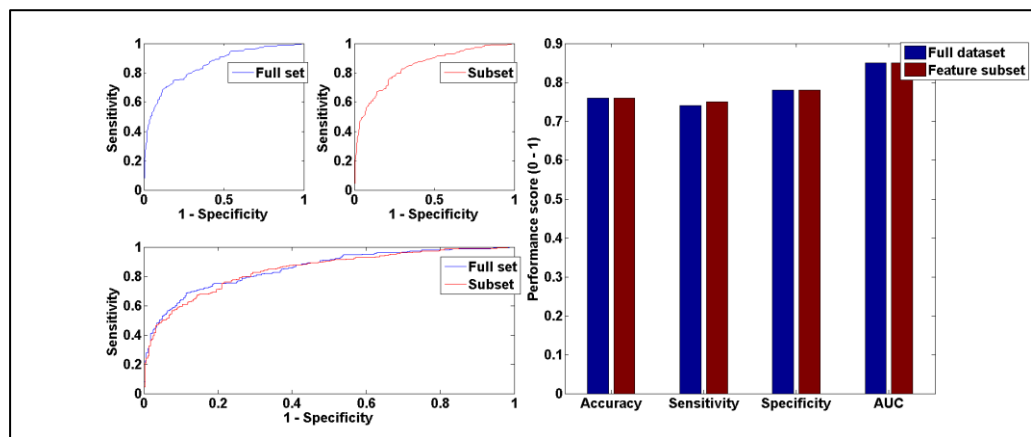


Figure 5.8: Graphic representation of SMO performance trained on full dataset vs feature subset

5.3.4 C4.5 PERFORMANCE COMPARISON

The results in Figure 5.9 indicate that classification B predictive accuracy (with selected feature subset) is marginally better than Classification A (with full training set). The total number of instances classified correctly is slightly higher in classification B with minimal and perhaps insignificant percentage error reduction on False negatives (0.02%); but same error count on False positives ($n = 72$). The Mc Nemar's test result shows that the accuracy difference between the two models is not significant ($P=0.7266$, $n=559$). Although a greater percentage of instances (77.8%) were correctly classified after feature selection (*Classification B = 1*) compared to 77.5% before feature selection (*Classification A = 1*); the difference is minimal (2.33%) with 95% confidence interval from -0.63% to 1.35%.

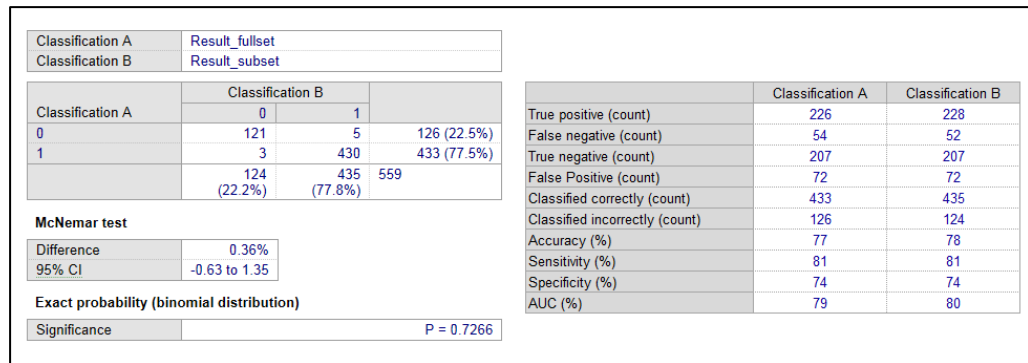


Figure 5.9: C4.5 performance with full training set vs selected feature subset using Accuracy, Sensitivity, Specificity, AUC and Mc Nemar's test

Similarly, the AUC performance is marginally higher in classification B (80%), compared to classification A (79%). Visual representations of the results are shown in Figure 5.10, including specificity and specificity performances which were tied at 81% and 74% respectively.

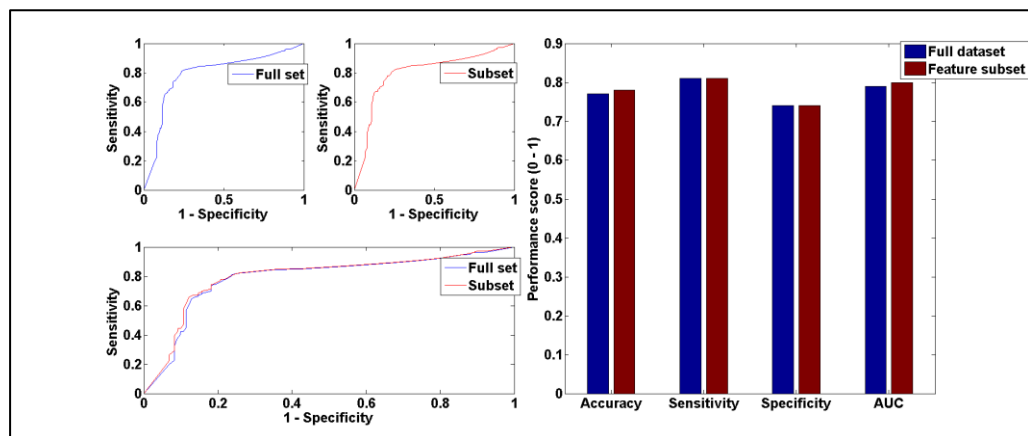


Figure 5.10: Graphic representation of C4.5 performance on full dataset vs feature subset

5.3.5 RIPPER PERFORMANCE COMPARISON

Predictive accuracy values appear to be the same (78%) between both classification experiments using RIPPER, as shown in Figure 5.11. However, the total number of instances classified correctly is marginally higher in classification B ($n = 438$) than classification A ($n = 437$). There is considerable difference in the hits on true positive instances with 223 for classification A (before feature selection) and 233 for classification B (after feature selection).

Classification A	Result_fullset		
Classification B	Result_subset		
	Classification B		
Classification A	0	1	
0	86	36	122 (21.8%)
1	35	402	437 (78.2%)
	121 (21.6%)	438 (78.4%)	559
McNemar test			
Difference	0.18%		
95% CI	-2.78 to 3.13		
Exact probability (binomial distribution)			
Significance	P = 1.0000		

	Classification A	Classification B
True positive (count)	223	233
False negative (count)	57	47
True negative (count)	214	205
False Positive (count)	65	74
Classified correctly (count)	437	438
Classified incorrectly (count)	122	121
Accuracy (%)	78	78
Sensitivity (%)	80	83
Specificity (%)	77	73
AUC (%)	79	80

Figure 5.11: RIPPER performance with full training set vs selected feature subset using Accuracy, Sensitivity, Specificity, AUC and Mc Nemar's test

That said, the Mc Nemar's test result shows that the accuracy difference between the two models is marginal (0.18%) with 95% confidence interval from -2.78% to 3.13%, which is insignificant ($P=1.0000$, $n=559$). Nonetheless, it is fair to say that any improvement at this level is acceptable because it has the potential to add value at ensemble level.

Slight improvement was achieved with classification B on the sensitivity (3%) and AUC (1%) performances. However, specificity value was higher in classification A by 4%. In general terms, this seems a bad result for classification B but the case is different from a medical view point, and perhaps preferable for the purpose of this experiment. The nature of the model discussed in this chapter requires a fairly high rate of correct detection in the minority class (positive) and allows for a small error rate in the majority class. Therefore, classification B could be seen as predicting based on the rare positive cases (low specificity), in order to reduce the risk of missing those unusual cases at high risk (high sensitivity). Visual representations of the results are shown in Figure 5.12.

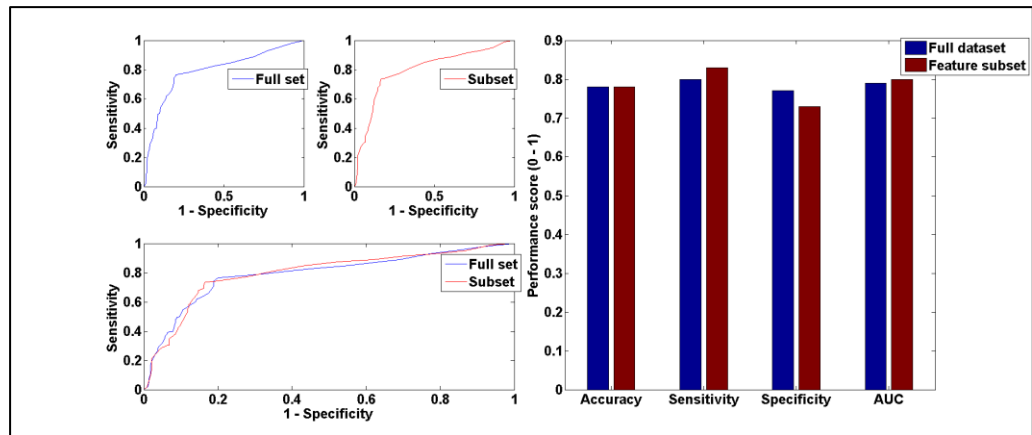


Figure 5.12: Graphic representation of RIPPER performance on full dataset vs feature subset

Recall that the purpose of feature selection in this thesis is mainly to improve accuracy at base level. However, it seems from the results that this was only achieved to a limited extent. For example, the predictive accuracy remained the same after feature selection was applied to two of the classifiers, namely SMO and RIPPER. In addition, where improvements were made (i.e, Naïve Bayes, RBF and C4.5), the accuracy differences are statistically insignificant. Nonetheless, improvements at this stage (no matter how small) must be viewed as positive because it has the potential to add value at ensemble level.

5.4 ENSEMBLE LEVEL PERFORMANCES

This section presents the performance results at ensemble level, of all the possible combinations of the five base classifiers (i.e., pair-wise, groups of threes, fours and all five). In total, 26 ensembles were trained and evaluated using predictive accuracy, sensitivity, specificity and AUC as metrics. The results (shown in Table 5.6) would be analysed to address the following questions:

4. Do ensembles **always** lead to better performance than the best individual constituent member at base level? (note: RBF model preferred at base level and all 4 performance metrics were compared separately).
5. Is the implemented ensemble method fit for purpose?

The first question is quite straight forward because it involves direct comparison of model performances. The second question however looks into any improvement(s) made from the base level. If any, what is the significance and how it relates to the data manipulation strategies implemented (i.e., feature selection and k -fold cross validation).

Table 5.6: Performance at ensemble level involving base classifier training (with data manipulation) in all possible combinations.

Classifier Models	Accuracy	Sensitivity	Specificity	AUC
RBF	0.78	0.82	0.73	0.85
SMO + RBF	0.78	0.82	0.73	0.80
SMO + C4.5	0.77	0.83	0.70	0.81
SMO + NB	0.77	0.77	0.77	0.77
SMO + RIPPER	0.78	0.83	0.72	0.80
RBF + C4.5	0.77	0.76	0.78	0.81
RBF + NB	0.78	0.81	0.75	0.80
RBF + RIPPER	0.76	0.79	0.73	0.81
C4.5 + NB	0.77	0.86	0.69	0.81
C4.5 + RIPPER	0.80	0.78	0.82	0.80
RIPPER + NB	0.78	0.87	0.69	0.81
SMO + RBF + C4.5	0.79	0.80	0.78	0.82
SMO + RBF + RIPPER	0.77	0.78	0.77	0.82
SMO + RBF + NB	0.77	0.81	0.73	0.80
SMO + C4.5 + RIPPER	0.79	0.80	0.78	0.82
SMO + C4.5 + NB	0.77	0.84	0.71	0.82
SMO + RIPPER + NB	0.78	0.85	0.70	0.81
RBF + C4.5 + RIPPER	0.79	0.82	0.77	0.82
RBF + C4.5 + NB	0.79	0.80	0.79	0.83
* C4.5 + RIPPER + NB	0.83	0.87	0.79	0.86
RBF + RIPPER + NB	0.78	0.79	0.77	0.82
SMO + RBF + C4.5 + RIPPER	0.79	0.83	0.76	0.82
SMO + RBF + C4.5 + NB	0.79	0.80	0.79	0.82
* RBF + C4.5 + RIPPER + NB	0.80	0.83	0.77	0.82
* SMO + NB + RIPPER + C4.5	0.80	0.83	0.77	0.82
RIPPER + SMO + RBF + NB	0.77	0.77	0.78	0.82
* NB + RBF + SMO + C4.5 + RIPPER	0.80	0.82	0.78	0.82

Note:

* denotes the top 4 ensembles

Selection criteria: Specificity $\geq 70\%$; and Accuracy, Sensitivity & AUC $\geq 80\%$

5.4.1 ENSEMBLE VS BASE LEARNER PERFORMANCE

To establish whether ensembles always lead to better performance than the best constituent member, comparison was made between RBF (preferred base model) and all the 26 ensemble models. The analyses would be conducted

separately for each of the four metrics. This is because performance metrics measure different trade-offs in the predictions made by a classifier and it is possible for classifiers to perform well on one metric, but be suboptimal on other metric(s).

Performance can vary between classifiers due to a number of reasons such as dataset composition, class distribution etc. For instance, classifiers that are based on training error minimisation (e.g., C4.5 and RIPPER) tend to do well in cases where there is clear separation between the classes within a dataset. That said, the focus of investigation in this section is whether ensembles always perform better than their constituent base classifiers (in terms of accuracy, sensitivity, specificity and AUC).

Figure 5.13 shows a cross comparison between the 26 ensemble models and RBF. Each data point on the graph represents a model and its relative performance. As shown in Figure 5.13(a), the RBF model is more accurate than some of the ensemble models. In fact, 9 out of 26 ensembles have lower accuracy value than RBF. This includes 5 ensemble models that included RBF.

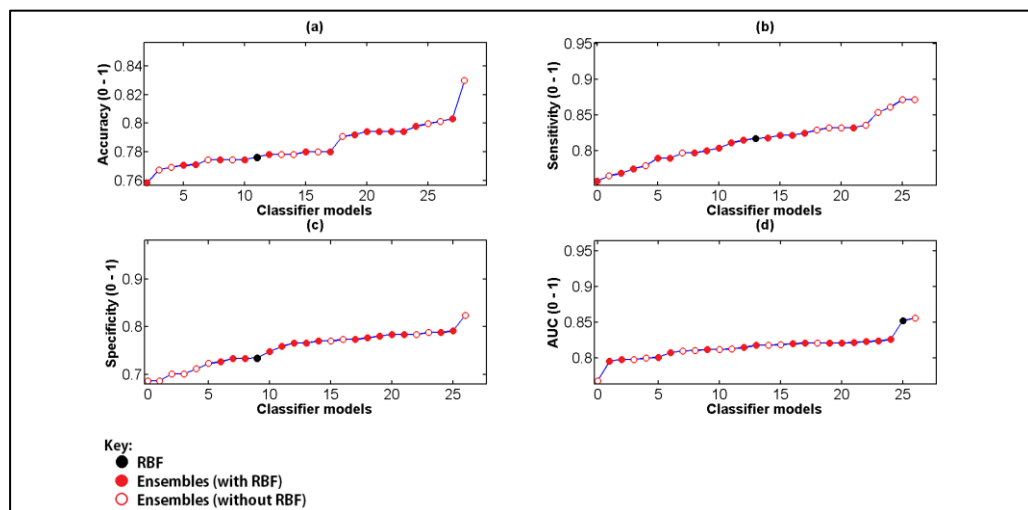


Figure 5.13: Direct comparison of the 26 ensembles and RBF performance based on accuracy, sensitivity, specificity and AUC.

Similar results were observed with the other metrics. As shown in Figure 5.13(b), RBF model produced better sensitivity than 13 ensembles; and 10 of them included RBF. In terms of specificity, RBF performed better than 9 ensembles as shown in Figure 5.13(c). This includes 3 ensemble models that included RBF. For the AUC, RBF is only second best as can be seen in Figure

5.13(d). The result suggests that ensembles do not always lead to better performance than its constituent members. In addition, some of the ensembles involving the RBF classifier produced lower accuracy than the RBF model on its own. This may be due to various reasons. For example, a model that contributes very little within the combination is likely to affect the final outcome in the same way a redundant feature within a dataset does during classification.

5.4.2 IMPACT OF THE ENSEMBLE METHOD IMPLEMENTED

This section evaluates the ensemble performances as a result of data manipulation at base level. It is important to note that analyses are specific to the ensemble method implemented in this thesis. One of the 26 ensemble models is selected with justification as the most appropriate for the classification task investigated (i.e., correct prediction of diabetes onset).

Basically, classifier models were selected if they achieved at least 80% in all the performance metrics, except specificity. It was decided to accommodate those with specificity value of at least 70%, because none of them achieved 80%. The selection threshold was chosen so that analysis can be focused on the area of interest (i.e., the best performing ensembles). Only 4 out of the 26 ensembles met the set criterion and were selected for further analysis. The models are denoted with ‘*’ in Table 5.6. Of the 4 models, the ensemble of C4.5+RIPPER+NB clearly performed better on all the metrics, thus selected as the preferred ensemble model. Henceforth, this model would be called ‘EN-mod1’ for simplicity.

It is clear from Table 5.6, that EN-mod1 performed better than RBF (the preferred base model). However, there is a need to examine the extent to which this is true. For this, Mc Nemar’s test (shown in Figure 5.14) was used to compare their predictions on the experimental data. Visual representations of the results are shown in Figure 5.15.

Classification A	Result_RBF		
Classification B	Result_EN_mod1		
	Classification B		
Classification A	0	1	
0	17	108	125 (22.4%)
1	78	356	434 (77.6%)
	95 (17.0%)	464 (83.0%)	559
McNemar test			
Difference	5.37%		
95% CI	0.61 to 10.13		
Exact probability (binomial distribution)			
Significance	P = 0.0332		

	Classification A	Classification B
True positive (count)	229	245
False negative (count)	51	36
True negative (count)	205	219
False Positive (count)	74	59
Classified correctly (count)	434	464
Classified incorrectly (count)	125	95
Accuracy (%)	78	83
Sensitivity (%)	82	87
Specificity (%)	73	79
AUC (%)	85	86

Figure 5.14: EN-mod1 vs RBF performance using Accuracy, Sensitivity, Specificity, AUC and Mc Nemar's test

83.0% of the instances were correctly classified by EN-mod1 (*Classification B = 1*) and 77.6% were correctly classified by RBF (*Classification A = 1*). The accuracy difference between both models is 5.37% with 95% confidence interval from 0.61% to 10.13%, which is significant ($P=0.0332$, $n=559$). The result highlights the predictive power of ensembles in complex data situations where the base classifiers struggle to improve performance individually. Although improvements were noted after feature selection at base level, they were so marginal and of no significance.

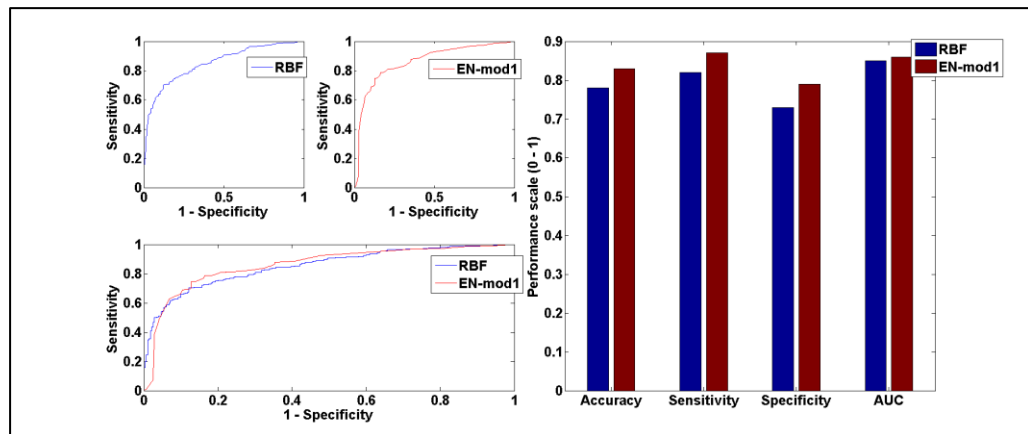


Figure 5.15: Graphic representation of EN-mod1 vs RBF model performance

Given the high classification accuracy on the positive class (true positive), it is not surprising that the sensitivity is relatively higher (5%) than the best recorded improvement after feature selection at base level (3% with RIPPER). Slight improvement was also achieved on the AUC performance (1%). However, specificity value was higher in classification A (with RBF) by 6%. This highlights the need to consider the characteristics of the problem when analysing any classification task. As discussed earlier, high sensitivity with low

specificity is preferable due to the nature of the task investigated in this thesis. In fact, the low specificity value reinforces the decision to lower the threshold when selecting the preferred ensemble classifier.

5.4.2.1 IMPACT OF DATA MANIPULATION

In order to establish if the feature selection applied at base level contributed to EN-mod1 performance and ensemble of c4.5, RIPPER and Naïve Bayes was re-trained. This time, the classifiers are trained on the full dataset and their predictions combined with k-NN algorithm. This would be called EN-mod2 and the results are compared with RBF to measure the level of improvement (if any). In addition, EN-mod2 would be compared to EN-mod1 as this would show the performance difference when trained with and without the feature subset.

Evidently from Figure 5.16, EN-mod2 did not improve the results obtained at base level. In fact, RBF performed considerably better on all the metrics. Predictive accuracy is better with RBF (78%) in comparison to 72% recorded for EN-mod2. This is not surprising because RBF had a hefty lead in terms of cases classified correctly. Of the correct classifications, the hits on true positive instances is considerably higher with RBF ($n = 229$), compared to classification B ($n = 189$).

Classification A	Result_RBF		
Classification B	Result_EN_mod2		

Classification A	Classification B			
	0	1		
	30	95		125 (22.4%)
	126	308		434 (77.6%)
	156 (27.9%)	403 (72.1%)		559

McNemar test

Difference	-5.55%
95% CI	-10.74 to -0.35

Exact probability (binomial distribution)

Significance	P = 0.0433
--------------	------------

	Classification A	Classification B
True positive (count)	229	189
False negative (count)	51	91
True negative (count)	205	214
False Positive (count)	74	65
Classified correctly (count)	434	403
Classified incorrectly (count)	125	156
Accuracy (%)	78	72
Sensitivity (%)	82	68
Specificity (%)	73	77
AUC (%)	85	78

Figure 5.16: EN-mod2 vs RBF performance using Accuracy, Sensitivity, Specificity, AUC and Mc Nemar's test

According to Mc Nemar's test result, 77.6% of the instances were correctly classified at base level (*Classification A = 1*) and 72.1% were correctly classified after ensemble level (*Classification B = 1*). The accuracy difference between the two models is in favour of classification A (RBF), signified by the

negative percentage value (-5.55%), with 95% confidence interval from -10.74% to -0.35%, which is significant ($P=0.0433$, $n=559$).

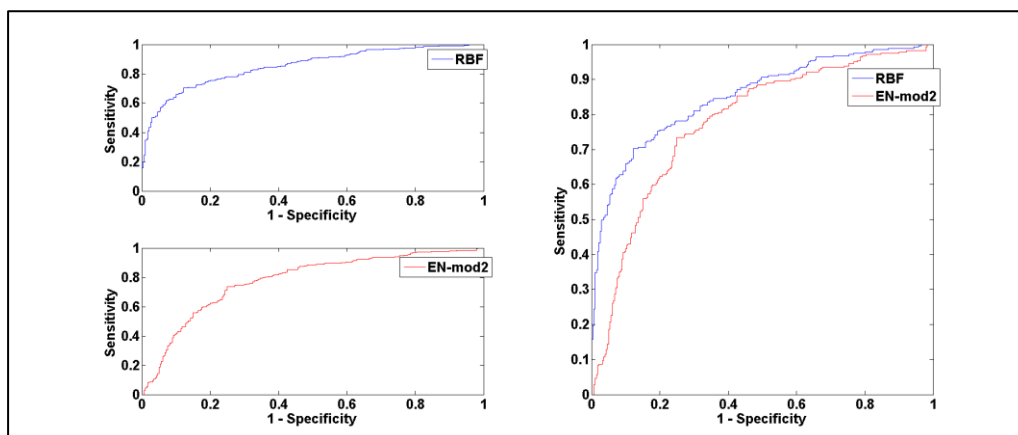


Figure 5.17: Graphic representation of EN-mod2 vs RBF model performance on AUC.

Considerable difference in performance was also recorded in favour of RBF on sensitivity (14%) and specificity (-4%). Negative difference is preferred for specificity due to the nature of the classification problem. Visual representation of the AUC results is shown in Figure 5.17, in which RBF performed better by 7%.

In this case, it can be noted that no improvement was made at ensemble level and the result highlights the negative impact of redundant features on classification tasks. However, the same cannot be implied (without proof) for all the possible ensembles, should the experiment be re-run on full training dataset to include the other 25 ensemble models. Classifiers react differently to changes in dataset so it is possible that EN-mod2 combination (c4.5, RIPPER and Naïve Bayes) is not among the high performing ensembles when trained on full data set. Therefore, the ensemble experiment was re-run to include all possible combinations of the base classifiers trained on full dataset. The results are shown in Table 5.7.

Since none of the ensemble models met the selection criteria used earlier (in Table 5.6), the criteria was amended to include only those models that produced at least 80% in AUC and at least 70% in the other three metrics. Only 6 of the models met this criteria (denoted with ‘*’ in Table 5.7), thus selected for further analysis. Of the 6, the ensembles of ‘RBF+c4.5+RIPPER+NB’ and

‘RBF+c4.5+NB’ produced the highest accuracy (76%). In view of the classification task investigated, the latter (i.e., RBF+C4.5+NB) was selected as the preferred ensemble model due to higher sensitivity (76%) with lower specificity (75%). Henceforth, this model would be called ‘EN-mod3’ for simplicity.

Table 5.7: Performance at ensemble level involving base classifier training (without data manipulation) in all possible combinations.

Classifier Models	Accuracy	Sensitivity	Specificity	AUC
RBF	0.78	0.82	0.73	0.85
SMO + RBF	0.70	0.70	0.70	0.74
SMO + C4.5	0.72	0.70	0.74	0.76
SMO + NB	0.65	0.63	0.68	0.73
SMO + RIPPER	0.69	0.70	0.68	0.71
RBF + C4.5	0.73	0.74	0.73	0.76
RBF + NB	0.73	0.73	0.73	0.78
RBF + RIPPER	0.69	0.69	0.70	0.72
C4.5 + NB	0.72	0.69	0.76	0.77
C4.5 + RIPPER	0.75	0.73	0.77	0.77
RIPPER + NB	0.68	0.67	0.69	0.71
SMO + RBF + C4.5	0.72	0.72	0.73	0.79
SMO + RBF + RIPPER	0.69	0.69	0.70	0.76
SMO + RBF + NB	0.70	0.70	0.70	0.76
SMO + C4.5 + RIPPER	0.72	0.71	0.72	0.77
SMO + C4.5 + NB	0.71	0.68	0.73	0.76
SMO + RIPPER + NB	0.70	0.69	0.72	0.75
RBF + C4.5 + RIPPER	0.73	0.71	0.75	0.76
* RBF + C4.5 + NB	0.76	0.76	0.75	0.82
C4.5 + RIPPER + NB	0.72	0.68	0.77	0.78
RBF + RIPPER + NB	0.71	0.72	0.70	0.77
* SMO + RBF + C4.5 + RIPPER	0.73	0.70	0.76	0.80
* SMO + RBF + C4.5 + NB	0.73	0.74	0.72	0.80
* RBF + C4.5 + RIPPER + NB	0.76	0.74	0.79	0.84
* SMO + NB + RIPPER + C4.5	0.74	0.71	0.76	0.80
RIPPER + SMO + RBF + NB	0.74	0.76	0.73	0.79
* NB + RBF + SMO + C4.5 + RIPPER	0.74	0.74	0.75	0.82

Note:

* denotes the top 6 ensembles

Selection criteria: AUC \geq 80%; and Accuracy, sensitivity & Specificity \geq 70%

It is clear from the result (in Table 5.7) that RBF performed better than EN-mod3, which makes it pointless to conduct detailed comparison between the two. On the other hand, there is value in comparing the performance of EN-mod3 with EN-mod1 because the performance difference would affirm the

significance of data manipulation in the ensemble method implemented in this thesis.

Figure 5.18, shows that EN-mod1 performed considerably better than EN-mod3 on all the metrics.

Classification A	Result_EN_mod3		
Classification B	Result_EN_mod1		

	Classification B		
Classification A	0	1	
0	24	112	136 (24.3%)
1	71	352	423 (75.7%)
	95 (17.0%)	464 (83.0%)	559

McNemar test

Difference	7.33%
95% CI	2.63 to 12.04

Exact probability (binomial distribution)

Significance	P = 0.0030
--------------	------------

	Classification A	Classification B
True positive (count)	214	245
False negative (count)	66	36
True negative (count)	209	219
False Positive (count)	70	59
Classified correctly (count)	423	464
Classified incorrectly (count)	136	95
Accuracy (%)	76	83
Sensitivity (%)	76	87
Specificity (%)	75	79
AUC (%)	82	86

Figure 5.18: EN-mod1 vs EN-mod3 performance using Accuracy, Sensitivity, Specificity, AUC and Mc Nemar's test

EN-mod1 has a hefty lead in terms of cases classified correctly. As a result, the accuracy value is considerably better with EN-mod1 (83%) in comparison to 76% recorded for EN-mod3. Of the correct classifications, the hits on true positive instances is considerably higher with EN-mod1 ($n = 245$), compared to classification B ($n = 214$). According to Mc Nemar's test result, the accuracy difference between the two models is 7.33% in favour EN-mod1, with 95% confidence interval from 2.63% to 12.04%, which is significant ($P=0.0030$, $n=559$).

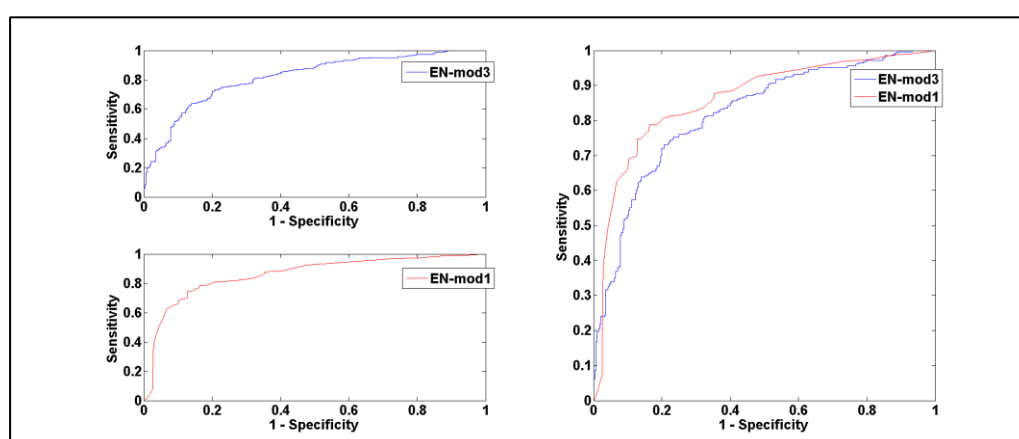


Figure 5.19: Graphic representation of EN-mod1 vs EN-mod3 model performance on AUC.

Considerable difference in performance was also recorded in favour of EN-mod1 on sensitivity (11%) and specificity (4%) and AUC (4%). Therefore, it

can be said that the ensemble method implemented in this thesis is fit for purpose. By manipulating the training data at base level, EN-mod1 model made a significant difference from what would have been the best ensemble, if feature selection was not applied. A visual representation of the AUC results is shown in Figure 5.17.

5.5 SUMMARY

The performance of 26 machine learning ensemble models trained with all possible combinations of 5 base classifiers was evaluated; and compared to the best constituent base classifier. The task is to determine if all the ensemble models outperformed the base classifiers; and where improvements were made, to measure its significance.

According to the results, ensemble models tend to yield better results than individual constituent classifiers. However this is not a certainty, as various factors may affect their ability to improve on performance, particularly at base level training. Issues such as redundant features, class imbalance and distribution within the training data were found to be major contributors to low performance. For instance, performance was relatively lower when the base classifiers were trained with unbalanced dataset compared to training with balanced dataset (see Appendix A.3).

The penalty that occurs if redundant features are not eliminated is evident in Table 5.7. The result shows that RBF performed better than any of the 26 ensemble models trained with full dataset. In fact, the penalty of redundant feature on accuracy is quite significant as shown in Figure 5.16. Nonetheless, the penalty was mitigated through feature selection applied at base level training. This is shown in Figure 5.14, where the RBF model is compared with the best ensemble model trained with feature selected subset (i.e., EN-Mod1). Significant improvement was noted in favour of EN-Mod1. This shows that feature selection played a key part to the improved accuracy.

Further observations from the experiment suggest that the highly desirable diversity when training ensembles can be achieved by using base classifiers un-

related to each other. Much of the previous work on ensemble classifier models have focused on a collection of a single base classifier trained in several variations. In this research, the base classifiers were selected from five broad families of machine learning algorithms. Therefore, each classifier induced a model based on its operational characteristics. Although none of them made improvement(s) of any significance at base level, the cumulative effect of their individual biases contributed to wider knowledge at ensemble level about the classification problem being addressed; ultimately leading to significant improvement.

It is important to note that the vast majority of the reported experiments in diabetes prediction only enhanced classification accuracy, up to 82% [192]. In fact, literature search of all the research conducted with the same dataset revealed a total of 70 eligible studies with accuracy results ranging from 59.5% to 82% (see Table 5.8). The research reported in this thesis produced 83%, so the implemented method can be said to perform relatively better.

Table 5.8: Research studies conducted with Pima Diabetes dataset

Paper	Method and Results	Result	Remarks
Tomasz Winiarski (http://www.fizyka.umk.pl/~twin/index.html)	Comparative study with 60 different classifiers.	Range = 59.5% - 77.7%	- Recorded accuracy value only. - Not sure if data was used in its original form.
K Polata, S Güneş & A Arslanb. (2008)	Ensemble of Generalized Discriminant Analysis (GDA) and Least Square Support Vector Machine (LS-SVM).	LS-SVM alone = 78.21% GDA+LS-SVM = 82%	- Used the original dataset which contains impossible values. - Recorded accuracy value only.
K Kayaer, & T Yıldırım. (2003).	- General Regression neural Network (GRNN). - Multilayer Neural network (MLNN) with Levenberg-Marquardt (LM) - Radial Basis Function (RBF) - Gradient Descent (GD) - BFGS quasi Newton	GRNN = 80.21% MLNN + LM = 77.08% RBF = 68.23% GD = 77.60% BFGS = 77.08%	- Used the original dataset which contains impossible values. - Recorded accuracy value only.
H Temurtasa, N Yumusak & F Temurtas. (2009).	Multilayer Neural network (MLNN) with Levenberg-Marquardt (LM)	MLNN + LM = 82.37%	- Used the original dataset which contains impossible values. - Recorded accuracy value only.
G A Carpenter & N Markuzon. (1998).	ARTMAP-Instance Counting (ARTMAP-IC)	ARTMAP-IC = 81%	- Used the original dataset which contains impossible values. - Recorded accuracy value only.
D Deng & N Kasabov. (2001).	Evolving Self Organising Maps (ESOM)	ESOM = 78.4%	- Used the original dataset which contains impossible values. - Recorded accuracy value only.
C V Subbulakshmi & S N Deepa. (2015).	Hybrid methodology that integrates Particle Swarm Optimization (PSO) algorithm with the Extreme Learning Machine (ELM) classifier.	93.09% Accuracy 91.47% Sensitivity 96.29% Specificity	Paper retracted for Malpractice

CHAPTER 6: CONCLUSIONS & FUTURE WORK

6.1 INTRODUCTION

This chapter restates the purpose of the research reported in this thesis. It presents a summary of the main points, results and knowledge contributions of the research undertaken from both health and computing perspectives. A concise assessment is provided for each point on how they support the purpose, and whether they align with or differ from other researchers' findings. Conclusions are drawn based on available evidence from the results with highlights to the limitations of the research work. A brief section on recommendation(s) for future research and practical applications forms the closing part of this chapter.

6.2 RESTATEMENT OF RESEARCH PURPOSE

The underlying goal of the research reported in this thesis is to examine how health examination data, can be utilised effectively to predict diabetes onset. A number of risk assessment tools exist that require some simple and easily accessible features to determine if a person is at risk of developing diabetes. However, such tool cannot be considered reliable due to lack of domain knowledge caused by limited (and often superficial) information. Features such as blood glucose concentration have been proven as reliable in diabetes screening [17], [18]; as such required to provide sufficient understanding of the condition, ultimately leading to better prediction.

In this research, medical data acquired through diabetes health check program was used. The task is to conduct experiments using machine learning approach, in order to learn from the data. In particular, the research explores the relationship between ensembles and their constituent base classifiers, to construct a high performance classifier model for diabetes prediction. Data optimisation strategies were applied during the experiment and their impact

evaluated. Results were analysed based on four performance measures to illustrate the level of achievements made.

6.3 LIMITATIONS

Majority of the research limitations revolve around data. The experimental data involving from the Pima Indian population is rather small and consists only of females aged 20 or over. Although the data was oversampled and measures put in place to counter any adverse effect on performance, there is some evidence that rebalancing the class distributions artificially does not have much effect on the performance of the induced classifier [193], This could be due to a number of reasons such as classifier not being sensitive to differences in class distributions. It seems that a clearer understanding is required of how class distributions affect each phase of the learning process at both base and ensemble levels. For instance, in C4.5 decision trees, how class distributions affect the node expansion, pruning and leaf labelling. A deeper understanding of the basics will lead to the design of better methods for dealing with the problems associated with skewed class distributions.

6.4 FUTURE RESEARCH

Although the experiment addressed the aims of the research with positive results, many directions still remain that could improve the performance. For instance, data pre-processing with other sampling methods may improve the dimensionality issue experienced with the experimental data. SMOTE was used in its basic version to oversample the minority class. Perhaps, other versions of SMOTE would improve the experimental data. Follow up experiment is necessary using other feature search algorithms, feature selection methods, and meta-classification methods. Another question that arose during the experiment is whether or not the base classifiers contribute equally to the training at ensemble level. It is intended to conduct further investigations in these directions. In the next sub sections, detailed plan of work is provided for each of the future research directions identified.

6.4.1 VARIATIONS OF SMOTE ALGORITHM

The version of SMOTE applied during the experiment uses the Heterogeneous Value Difference Metric (HVDM) [194] to compute the distance between examples; and considers a maximum of 5 nearest neighbours for each sample. Research into other data sampling methods would lead to a better understanding of the dimensionality issue experienced with the experimental data. It is intended to conduct further research using other versions of the SMOTE algorithm, to see what improvement(s) could be made.

In particular, the SMOTE + Tomek Links [195] has a built in facility to separate the synthetic samples generated by SMOTE. The method uses Tomek Links [196] to remove examples after applying SMOTE, that are considered to be noisy or lying in the decision border. By definition, a Tomek Links is a pair of examples x and y from different classes, that has no example z such that $d(x, z)$ is lower than $d(x, y)$, or $d(y, z)$ is lower than $d(x, y)$, where d is the distance metric.

SMOTE + ENN [197] is another version worth considering. According to Prati et al [198], Edited Nearest Neighbour (ENN) tends to remove more examples than the Tomek Links, so it is expected that a more thorough data cleaning would be achieved through this method. For instance, ENN uses three nearest neighbours to assess examples from both classes, thus any example that is misclassified by its three nearest neighbours is removed from the training set.

Borderline SMOTE [180] is another variation of SMOTE that considers the minority borderline examples, when generating synthetic data. This method uses the K-Nearest Neighbour (K-NN) algorithm [165] to identify the k nearest neighbours of each minority class example. If a minority class example X_i has more than $\frac{k}{2}$ nearest neighbors from other classes, then X_i is considered a borderline example that might be misclassified, and X_i is fed to SMOTE so that synthetic examples are generated around it. If however, X_i has exactly the same k nearest neighbours from other classes, then X_i is considered noisy and no synthetic examples are generated for it.

SL-SMOTE is another useful method in which an assessment is conducted for each minority class, in order to identify its safe level before generating synthetic examples. By definition, the safe level of one example is the number of positive instances among its K-NN. Synthetic examples are positioned closer to the examples with the largest safe level to reduce the chances of misclassification.

It is believed that these methods would be beneficial to the experimental data used in this thesis. By applying additional selection measures around the minority examples, distinctive classes may be generated, ultimately leading to improved performance of the classifiers. Further work is planned to compare performance using these methods. In particular, it would be interesting to see the results of SMOTE + Tomek Links and SMOTE + ENN which are noted within the literature to perform better than the other two versions of SMOTE discussed in this section [177][199].

6.4.2 EXTENDED RESEARCH WITH DIFFERENT WEIGHTED VOTE

One of the questions that arose during the experiment was whether or not the base classifiers contribute equally at ensemble level. In the experiment, K-NN was used as meta-classifier to combine the predictions of the base classifiers. Classification through this process is done based on the distance between new observation and the learning set closest to the new observation. The problem is that synthetic data generated through SMOTE are not properly separated, so it is possible that some of the nearest neighbours to the new observation are of the opposite class. Therefore, weighting was assigned to the contributions of the neighbours, so that the nearer neighbours contribute more to the average than the more distant ones.

While this approach produced good ensemble results, there are other directions not yet exploited. For instance, K-NN only considers the predicted class label and not the performance of the individual classifiers that make up the ensemble. Since contributions made by the ensemble members vary, there is a need to acknowledge each classifier's contribution so that those with greater information gain would have more votes towards the ensemble prediction.

One way of doing this is to assign weight to each classifier based on its probability distribution over the class. When a classifier outputs the most likely class that a new sample should belong to, it provides the degree to it believes the prediction is true. This degree of certainty, (commonly known as prediction probability) can be utilised to assign weights to the base classifiers' predictions so that those with higher probability on the class contribute more towards the ensemble prediction. For instance, given a binary classification task with class labels $i \in \{0,1\}$, N number of base classifiers, the prediction y by weighted predicted probability p is given by (12), where w_j is the weight assigned to the j^{th} classifier.

$$y = \arg \max_i \sum_{j=1}^N w_j p_{ij} \quad (12)$$

This can be implemented in two ways; a) with equal weight assigned to each classifier and b) different weight for each classifier based on some function. To illustrate the two methods using a simple example, the base classifiers could produce predicted probabilities like the one in Table 6.1.

Table 6.1: Simple classification result from three classifiers, showing weighted predictions on each class

Classifiers	Class 0	Class 1
$C_1(x)$	$w \times 0.8$	$w \times 0.2$
$C_2(x)$	$w \times 0.6$	$w \times 0.4$
$C_3(x)$	$w \times 0.4$	$w \times 0.6$

Using uniform weights $w = 1$, for each classifier, the prediction y by average probabilities can be computed as:

$$p(i_0|x) = \frac{0.8 + 0.6 + 0.4}{3} = 0.6$$

$$p(i_1|x) = \frac{0.2 + 0.4 + 0.6}{3} = 0.4$$

$$y = \arg \max_i [p(i_0|x), p(i_1|x)] = 0$$

However, assigning different weights $\{0.1, 0.1, 0.8\}$ would yield a prediction $y = 1$

$$p(i_0|x) = \frac{0.1 \times 0.8 + 0.1 \times 0.6 + 0.8 \times 0.4}{3} = 0.46$$

$$p(i_1|x) = \frac{0.1 \times 0.2 + 0.1 \times 0.4 + 0.8 \times 0.6}{3} = 0.54$$

$$y = \arg \max_i [p(i_0|x), p(i_1|x)] = 1$$

Since both strategies produced different outcomes for y , it seems logical to implement and compare both of them to the result achieved with K-NN algorithm. It is intended that further research would be conducted in this direction. Prediction probability could be used to calculate weighting function such that those with larger values are assigned more weight.

6.4.3 BASE LEARNER OPTIMISATION AND FURTHER EXPERIMENTS WITH EXTERNAL DATASET

The ensembles reported in this thesis utilises five base classifiers in their standard form, learning from a single dataset. Further research is intended to optimise the base learners by tuning their hyper-parameters. In the context of machine learning, hyper-parameters are parameters whose values are set prior to the classifier training process [200]. By contrast, the value of other parameters is derived via training and dependent on the data. It may be possible to improve performance at base level through this process, ultimately leading to improved ensembles.

All the experiments reported in this thesis are based on a single dataset. It is possible that the conclusions drawn from the experiments would hold for other datasets, but this is not a certainty. Abbasi et al. [16], argued that the performance of a prediction model is generally overestimated in the population in which it was developed. Therefore external validation of such model in an independent population is essential to broadly evaluate the performance and

thus the potential utility of such models in different populations and settings. It would be interesting to see if the achieved performance would be replicated, given a different dataset. Therefore, it is intended to replicate the research using external datasets; and perhaps more base classifiers.

6.4.4 EXTENDED RESEARCH IN FEATURE SEARCH AND SELECTION

For the experiment reported in this thesis, only one feature search and selection approach was implemented. The approach uses the best-first algorithm to search the feature space such that the selected subsets are tested and scored with the base classifiers, for optimum performance. This means that each new subset is used to train a model, which is tested on a hold-out set. By comparing the error rate of the models, scores are allocated to each subset.

In a separate research, this approach was compared with a different feature selection method (known as filter) [201]. Filter methods use statistical measure to consider each feature independently, and assign a scoring with regard to information gain to the class [202]. Comparison between the two methods have been covered by several researchers and there is a general consensus that filters do not perform well because features are considered independently [201][203]. As a result, further research in this is focused on the various search algorithms used for selecting subsets from the feature space. For instance, genetic search [204], exhaustive search [205] and greedy hill climbing [206][207] are some of the most frequently used search algorithms within the literature. All three would be implemented with the feature selection approach used in this thesis.

6.5 THESIS SUMMARY

Problems of data are one of the most emphasised factors affecting diabetes prediction tools within the literature, particularly superficial data and small/skewed data for training. The former was rectified in the research presented in this thesis by including vital bio markers most closely associated with diabetes development such as blood pressure and glucose concentration. In fact, the result obtained during feature selection in (see chapter 5.3), validates their inclusion and supports the wider claim about their relevance in

diabetes data classification. Blood glucose and blood pressure are among the few features selected by all the five classifiers.

Evidence from this research also aligns with previous research work about the adverse effects of the latter problem involving data size. In the comparison involving base classifier training with unbalanced and balanced data (see Appendix A.3), small data sample coupled with skewed class distribution was seen to affect classifier performance. An attempt was made in this thesis to address the issue through over sampling the minority class using SMOTE algorithm.

Evidence within the literature suggests that feature selection improves performance. This was corroborated by the results in this thesis, particularly when the base classifiers trained with feature selected subsets were compared to their counterparts trained on full dataset (see chapter 5.3). That said, this observation is declared with caution because only one feature selection method was investigated herein. Extensive research with other methods would provide stronger claims on this note.

The experiments show that heterogeneous pool of base classifiers is capable of producing accurate and diverse ensemble classifiers. The implemented method performed a search over all possible heterogeneous model compositions involving only five base classifier models. There was significant improvement in predictive accuracy when the best ensemble was compared to the best base learner. That said, some of the ensemble models produced lower performance than the best base classifier. Therefore, the results of the experiment differ from any claim(s) within the literature that ensembles always lead to better performance than its constituent base classifiers.

Further observations suggest that feature selection played a major role towards the results. This was proven in section 5.4.2.1 in which comparison was made between ensembles trained with and without feature selected data. The results revealed some poor performance from the latter but validates claims in the literature about the effects of redundant data on classifier performance.

As noted in the previous chapter (Section 5.5), the accuracy of the ensemble method implemented in this thesis is superior compared to other methods described in the literature. 70 research studies were found in the literature that used the same dataset. Their accuracy results are between 59.5% and 82%, which is lower than 83% obtained in this research.

BIBLIOGRAPHY

- [1] C. L. Gillies, K. R. Abrams, P. C. Lambert, N. J. Cooper, A. J. Sutton, R. T. Hsu, and K. Khunti, "Pharmacological and lifestyle interventions to prevent or delay type 2 diabetes in people with impaired glucose tolerance: systematic review and meta-analysis.," *BMJ*, vol. 334, no. 7588, p. 299, Feb. 2007.
- [2] W. H. Herman, T. J. Hoerger, M. Brandle, K. Hicks, and S. Sorensen, "The Cost-Effectiveness of Lifestyle Modification or Metformin in Preventing Type 2 Diabetes in Adults with Impaired Glucose Tolerance," *Ann Intern Med*, vol. 142, no. 5, pp. 323–332, 2009.
- [3] P. Zimmet, K. G. M. M. Alberti, and J. Shaw, "Global and societal implications of the diabetes epidemic," *Nature*, vol. 414, no. December 2001, pp. 782–787, 2001.
- [4] Diabetes UK, "Number of people with diabetes up 60 per cent in last decade," 2015. [Online]. Available: https://www.diabetes.org.uk/About_us/News/diabetes-up-60-per-cent-in-last-decade-/. [Accessed: 29-Jul-2017].
- [5] Diabetes UK, "Diabetes Prevalence," 2017. [Online]. Available: <http://www.diabetes.co.uk/diabetes-prevalence.html>. [Accessed: 29-Jul-2017].
- [6] Diabetes UK, "Diabetes: Facts and Stats," 2015.
- [7] D. L. Burnet, L. D. Elliott, M. T. Quinn, A. J. Plaut, M. A. Schwartz, and M. H. Chin, "Preventing diabetes in the clinical setting," *J. Gen. Intern. Med.*, vol. 21, no. 1, pp. 84–93, Jan. 2006.
- [8] T. Yates, J. Jarvis, J. Troughton, and M. J. Davies, "Preventing type 2 diabetes: applying the evidence in nursing practice.," *Nurs. Times*, vol. 105, no. 41, pp. 10–4, 2009.
- [9] K. Khavandi, H. Amer, B. Ibrahim, and J. Brownrigg, "Strategies for preventing type 2 diabetes: an update for clinicians," *Ther. Adv. Chronic Dis.*, vol. 4, no. 5, pp. 242–261, Sep. 2013.
- [10] The Diabetes Prevention Program Research Group, "The 10-Year Cost-Effectiveness of Lifestyle Intervention or Metformin for Diabetes Prevention," *Diabetes Care*, vol. 35, no. 4, pp. 723–730, 2012.
- [11] W. C. Knowler, E. Barrett-Connor, S. E. Fowler, R. F. Hamman, J. M. Lachin, E. a Walker, and D. M. Nathan, "Reduction in the incidence of type 2 diabetes with lifestyle intervention or metformin.," *N. Engl. J. Med.*, vol. 346, no. 6, pp. 393–403, Feb. 2002.
- [12] Diabetes Prevention Program Research Group, W. C. Knowler, S. E. Fowler, R. F. Hamman, C. A. Christophi, H. J. Hoffman, A. T. Brenneman, J. O. Brown-Friday, R. Goldberg, E. Venditti, and D. M. Nathan, "10-year follow-up of diabetes incidence and weight loss in the Diabetes Prevention Program Outcomes Study.," *Lancet (London, England)*, vol. 374, no. 9702, pp. 1677–86, Nov. 2009.

- [13] B. Paulweber, P. Valensi, J. Lindström, N. M. Lalic, C. J. Greaves, M. McKee, K. Kissimova-Skarbek, S. Liatis, E. Cosson, J. Szendroedi, K. E. Sheppard, K. Charlesworth, A.-M. Felton, M. Hall, A. Rissanen, J. Tuomilehto, P. E. Schwarz, M. Roden, M. Paulweber, A. Stadlmayr, L. Kedenko, N. Katsilambros, K. Makrilakis, Z. Kamenov, P. Evans, A. Gilis-Januszezwska, K. Lalic, A. Jotic, P. Djordevic, V. Dimitrijevic-Sreckovic, U. Hühmer, B. Kulzer, S. Puhl, Y. H. Lee-Barkey, A. AlKerwi, C. Abraham, W. Hardeman, T. Acosta, M. Adler, A. AlKerwi, N. Barengo, R. Barengo, J. M. Boavida, K. Charlesworth, V. Christov, B. Claussen, X. Cos, E. Cosson, S. Deceukelier, V. Dimitrijevic-Sreckovic, P. Djordjevic, P. Evans, A.-M. Felton, M. Fischer, R. Gabriel-Sanchez, A. Gilis-Januszezwska, M. Goldfracht, J. L. Gomez, C. J. Greaves, M. Hall, U. Handke, H. Hauner, J. Herbst, N. Hermanns, L. Herrebrugh, C. Huber, U. Hühmer, J. Huttunen, A. Jotic, Z. Kamenov, S. Karadeniz, N. Katsilambros, M. Khalangot, K. Kissimova-Skarbek, D. Köhler, V. Kopp, P. Kronsbein, B. Kulzer, D. Kyne-Grzebalski, K. Lalic, N. Lalic, R. Landgraf, Y. H. Lee-Barkey, S. Liatis, J. Lindström, K. Makrilakis, C. McIntosh, M. McKee, A. C. Mesquita, D. Misina, F. Muylle, A. Neumann, A. C. Paiva, P. Pajunen, B. Paulweber, M. Peltonen, L. Perrenoud, A. Pfeiffer, A. Pölönen, S. Puhl, F. Raposo, T. Reinehr, A. Rissanen, C. Robinson, M. Roden, U. Rothe, T. Saaristo, J. Scholl, P. E. Schwarz, K. E. Sheppard, S. Spiers, T. Stemper, B. Stratmann, J. Szendroedi, Z. Szybinski, T. Tankova, V. Telle-Hjellset, G. Terry, D. Tolks, F. Toti, J. Tuomilehto, A. Undeutsch, C. Valadas, P. Valensi, D. Velickiene, P. Vermunt, R. Weiss, J. Wens, and T. Yilmaz, “A European evidence-based guideline for the prevention of type 2 diabetes.” *Horm. Metab. Res.*, vol. 42 Suppl 1, pp. S3-36, Apr. 2010.
- [14] K. G. M. M. Alberti, P. Zimmet, and J. Shaw, “International Diabetes Federation: a consensus on Type 2 diabetes prevention.” *Diabet. Med.*, vol. 24, no. 5, pp. 451–63, May 2007.
- [15] Diabetes UK, “Type 2 Diabetes Know Your Risk,” 2017. [Online]. Available: <http://riskscore.diabetes.org.uk/start>. [Accessed: 02-Jun-2017].
- [16] A. Abbasi, L. M. Peelen, E. Corpeleijn, Y. T. van der Schouw, R. P. Stolk, A. M. W. Spijkerman, D. L. van der A, K. G. M. Moons, G. Navis, S. J. L. Bakker, and J. W. J. Beulens, “Prediction models for risk of developing type 2 diabetes: systematic literature search and independent external validation study,” *BMJ*, vol. 345, no. sep18 2, pp. e5900–e5900, Sep. 2012.
- [17] A. Bur, “Is fasting blood glucose a reliable parameter for screening for diabetes in hypertension?,” *Am. J. Hypertens.*, vol. 16, no. 4, pp. 297–301, Apr. 2003.
- [18] P. Patel and A. Macerollo, “Diabetes mellitus: Diagnosis and screening,” *Am. Fam. Physician*, vol. 81, no. 7, pp. 863–870, 2010.
- [19] D. M. Tisnado, J. L. Adams, H. Liu, C. L. Damberg, W.-P. Chen, F. A. Hu, D. M. Carlisle, C. M. Mangione, and K. L. Kahn, “What is the concordance between the medical record and patient self-report as data

sources for ambulatory care?,” *Med. Care*, vol. 44, no. 2, pp. 132–40, Feb. 2006.

- [20] D. L. Schacter and C. S. Dodson, “Misattribution, false recognition and the sins of memory,” *Philos. Trans. R. Soc. B Biol. Sci.*, vol. 356, no. 1413, pp. 1385–1393, Sep. 2001.
- [21] D. L. Schacter, “Memory: sins and virtues,” *Ann. N. Y. Acad. Sci.*, vol. 1303, no. 1, pp. 56–60, Nov. 2013.
- [22] A. Furnham, “Response bias, social desirability and dissimulation,” *Pers. Individ. Dif.*, vol. 7, no. 3, pp. 385–400, Jan. 1986.
- [23] T. D. Cook and D. T. Campbell, *Quasi-experimentation: Design and analysis issues for field settings*. Chicago: Rand McNally. Houghton Mifflin, 1979.
- [24] A. J. Nederhof, “Methods of coping with social desirability bias: A review,” *Eur. J. Soc. Psychol.*, vol. 15, no. 3, pp. 263–280, Jul. 1985.
- [25] NHS Choices, “NHS Health Check,” 2016. [Online]. Available: <http://www.nhs.uk/conditions/nhs-health-check/pages/what-is-an-nhs-health-check-new.aspx>. [Accessed: 02-Jun-2017].
- [26] A. Munoz, “Machine Learning and Optimization,” 2014. [Online]. Available: https://www.cims.nyu.edu/~munoz/files/ml_optimization.pdf. [Accessed: 29-Jul-2017].
- [27] L. K. Hansen and P. Salamon, “Neural Network Ensembles,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. October, pp. 993–1001, 1990.
- [28] T. G. Dietterich, “Ensemble Methods in Machine Learning,” in *Multiple Classifier Systems*, 2000.
- [29] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, “On combining classifiers,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998.
- [30] L. I. Kuncheva, *Combining pattern classifiers: methods and algorithms*, 2nd ed. Wiley, 2014.
- [31] L. Lam, “Multiple Classifier Systems: Implementations and Theoretical Issues,” in *Lecture Note in Computer Science*, Berlin Heidelberg: Springer, 2000, pp. 77–86.
- [32] G. I. Webb and Z. Zheng, “Multistrategy ensemble learning: reducing error by combining ensemble learning techniques,” *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 8, pp. 980–991, Aug. 2004.
- [33] T. G. Dietterich, “Multiple Classifier Systems,” in *Lecture Notes in Computer Science*, vol. 1857, Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 1–15.
- [34] N. Nnamoko, F. Arshad, D. England, and J. Vora, “Meta-classification Model for Diabetes onset forecast: a proof of concept,” in *IEEE International Conference on Bioinformatics and Biomedicine Workshops*, 2014, pp. 50–56.
- [35] V. Bhatnagar, M. Bhardwaj, S. Sharma, and S. Haroon, “Accuracy–

- diversity based pruning of classifier ensembles,” *Prog. Artif. Intell.*, vol. 2, no. 2–3, pp. 97–111, Jun. 2014.
- [36] L. Breiman, “Bagging predictors,” *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
 - [37] L. Breiman, “Random Forest,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
 - [38] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Blackwell, 2000.
 - [39] I. Kononenko and M. Kovacic, “Learning as Optimization: Stochastic Generation of Multiple Knowledge,” in *Ninth International Workshop (ML92)*, 1992, pp. 257–262.
 - [40] P. Smyth, R. M. Goodman, and C. M. Higgins, “A Hybrid Rule-Based/Bayesian Classifier,” in *ECAI*, 1990, pp. 610–615.
 - [41] G. Giacinto, F. Roli, and G. Fumera, “Design of effective multiple classifier systems by clustering of classifiers,” in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, 2000, vol. 2, pp. 160–163.
 - [42] G. . Partalas and I. Vlahavas, “Focussed ensemble selection: a diversity based method for greedy ensemble selection,” in *Proceedings of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, 2008, pp. 117–121.
 - [43] Z. Lu, X. Wu, X. Zhu, and J. Bongard, “Ensemble pruning via individual contribution ordering,” in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '10*, 2010, p. 871.
 - [44] D. D. Margineantu and T. G. Dietterich, “Pruning adaptive boosting,” in *ICML '97 Proceedings of the Fourteenth International Conference on Machine Learning*, 1997, pp. 211–218.
 - [45] G. Martínez-Muñoz and A. Suárez, “Pruning in ordered bagging ensembles,” in *Proceedings of the 23rd international conference on Machine learning - ICML '06*, 2006, pp. 609–616.
 - [46] G. Martinez-Muoz, D. Hernandez-Lobato, and A. Suarez, “An Analysis of Ensemble Pruning Techniques Based on Ordered Aggregation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 245–259, Feb. 2009.
 - [47] I. Partalas, G. Tsoumakas, and I. Vlahavas, “Pruning an ensemble of classifiers via reinforcement learning,” *Neurocomputing*, vol. 72, no. 7–9, pp. 1900–1909, Mar. 2009.
 - [48] C. Tamon and J. Xiang, “On the Boosting Pruning Problem,” in *Lecture Note in Computer Science*, 2003, pp. 404–412.
 - [49] S. W. Kwok and C. Carter, “Multiple decision trees,” in *Fourth Conference on Uncertainty in Artificial Intelligence*, 1990, pp. 213–220.
 - [50] W. Buntine, “Theory Refinement on Bayesian Networks,” in *Uncertainty in Artificial Intelligence*, 1991, vol. 1, no. 415, pp. 52–60.

- [51] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.
- [52] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory*. Wiley & Sons Inc, 1994.
- [53] Diabetes UK, "Diabetes in the UK 2011/2012: Key statistics on diabetes," 2012.
- [54] E. et. al. Loveman, "The clinical effectiveness of diabetes education models for Type 2 diabetes," *Health Technol. Assess. (Rockv)*, vol. 12, no. 9, 2008.
- [55] J. Tuomilehto, J. Lindstrom, J. G. Eriksson, T. T. Valle, H. Hamalainen, P. Ilanne-Parikka, S. Keinanen-Kiukaanniemi, M. Laakso, A. Louheranta, M. Rastas, V. Salminen, and M. Uusitupa, "Prevention of type 2 Diabetes Mellitus by changes in lifestyle among subjects with impaired glucose tolerance," *New English J. Med.*, vol. 344, no. 18, pp. 1343–1350, 2001.
- [56] Diabetes Prevention Program Outcomes Study Research Group, "Diabetes Prevention Program," 2011. [Online]. Available: http://www.bsc.gwu.edu/dpp/p3_1.pdf. [Accessed: 12-Dec-2012].
- [57] N. Unwin, K. G. M. M. Alberti, R. Bhopal, J. Harland, W. Watson, and M. White, "Comparison of the Current WHO and New ADA Criteria for the Diagnosis of Diabetes Mellitus in Three Ethnic Groups in the UK," *Diabetes Med.*, vol. 15, pp. 554–557, 1998.
- [58] NICE public health guidance 38, "Preventing type 2 diabetes: risk identification and interventions for individuals at high risk," 2012.
- [59] World Health Organisation, "Waist Circumference and Waist-Hip Ratio," Geneva, 2008.
- [60] American Diabetes Association, "Standards of Medical Care in," *Diabetes Care*, vol. 35, no. 1, pp. S11–S63, 2012.
- [61] T. R. Willemain and R. G. Mark, "Models of remote health care systems," *Biomed. Sci. Instrum.*, vol. 8, pp. 9–17, Jan. 1971.
- [62] J. Craig and V. Patterson, "Introduction to the practice of telemedicine.," *J. Telemed. Telecare*, vol. 11, no. 1, pp. 3–9, Jan. 2005.
- [63] Cesnik B and Kidd MR, "History of health informatics: a global perspective.," *Stud Heal. Technol Inf.*, vol. 151, pp. 3–8, 2010.
- [64] W. Einthoven, "Le télécardiogramme [The telecardiogram]," *Arch. Int. Physiol.*, vol. 4, pp. 132–164, 1906.
- [65] A. Jutras and G. Duckett, "Distant radiodiagnosis; telefluoroscopy & cinefluorography," *J. Union Med. Canada*, vol. 86, no. 11, pp. 1284–9, 1957.
- [66] C. Wittson, D. Affleck, and V. Johnson, "Two-way television in group therapy," *Ment. Hosp.*, vol. 12, no. 10, pp. 22–23, 1961.
- [67] N. Cunningham, C. Marshall, and E. Glazer, "Telemedicine in Pediatric Primary Care: Favorable Experience in Nurse-Staffed Inner-City Clinic," *J. Am. Med. Assoc.*, vol. 240, no. 25, pp. 2749–2751, 1978.

- [68] R. Benschoter, M. Eaton, and P. Smith, "Use of videotape to provide individual instruction in techniques of psychotherapy," *J. Med. Educ.*, vol. 40, no. 12, pp. 1159–61, 1965.
- [69] F. J. Menolascino and O. R. G, "Psychiatric television consultation for the mentally retarded," *Am. J. Psychiatry*, vol. 127, no. 4, pp. 515–520, 1970.
- [70] T. F. Dwyer, "Telepsychiatry: psychiatric consultation by interactive television.," *Am. J. Psychiatry*, vol. 130, no. 8, pp. 865–9, 1973.
- [71] N. Straker, P. Mostyn, and C. Marshall, "The use of two-way TV in bringing mental health services to the inner city," *Am. J. Psychiatry*, vol. 133, no. 10, pp. 1202–5, 1976.
- [72] M. Snyder, "Self-monitoring of expressive behaviour," *J. Pers. Soc. Psychol.*, vol. 30, pp. 526–37, 1974.
- [73] S. Ryu, "Telemedicine: Opportunities and Developments in Member States: Report on the Second Global Survey on eHealth 2009 (Global Observatory for eHealth Series, Volume 2)," *Healthc. Inform. Res.*, vol. 18, no. 2, p. 153, 2012.
- [74] R. Currell, C. Urquhart, P. Wainwright, and R. Lewis, "Telemedicine versus face to face patient care : effects on professional practice and health care outcomes (Review)," 2000.
- [75] A. M. House and J. M. Roberts, "Telemedicine in Canada," *Can. Med. Association*, vol. 117, no. 4, pp. 386–388, 1977.
- [76] M. Maheu, P. Whitten, and A. Allen, "From Telemedicine and Telehealth to E-Health," in *E-Health, Telehealth, and Telemedicine: A Guide to Startup and Success*, 2001.
- [77] E. D. Lehmann, "Information technology in clinical diabetes care--a look to the future," *Diabetes Technol. Ther.*, vol. 6, no. 5, pp. 755–759, 2004.
- [78] S. Montani, P. Magni, R. Bellazzi, C. Larizza, A. V. Roudsari, and E. R. Carson, "Integrating model-based decision support in a multi-modal reasoning system for managing type 1 diabetic patients," *Artif. Intell. Med.*, vol. 29, no. 1–2, pp. 131–151, Sep. 2003.
- [79] E. D. Lehmann and T. Deutsch, "Compartmental models for glycaemic prediction and decision-support in clinical diabetes care: promise and reality.," *Comput. Methods Programs Biomed.*, vol. 56, no. 2, pp. 193–204, May 1998.
- [80] I. Bichindaritz and C. Marling, "Case-based reasoning in the health sciences: What's next?," *Artif. Intell. Med.*, vol. 36, no. 2, pp. 127–35, Feb. 2006.
- [81] D. Dazzi, F. Taddei, a Gavarini, E. Uggeri, R. Negro, and a Pezzarossa, "The control of blood glucose in the critical diabetic patient: a neuro-fuzzy method.," *J. Diabetes Complications*, vol. 15, no. 2, pp. 80–7, 2001.
- [82] P. P. San, S. H. Ling, and H. T. Nguyen, "Intelligent detection of hypoglycemic episodes in children with type 1 diabetes using adaptive

neural-fuzzy inference system,” *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, vol. 2012, pp. 6325–8, Jan. 2012.

- [83] E. D. Lehmann and T. Deutsch, “AIDA: An Automated Insulin Dosage Advisor,” in *Proc Annu Symp Comput Appl Med Care*, 1992, pp. 818–819.
- [84] E. D. Lehmann, T. Deutsch, E. R. Carson, and P. H. Sönksen, “AIDA: an interactive diabetes advisor,” *Comput. Methods Programs Biomed.*, vol. 41, no. 3–4, pp. 183–203, Jan. 1994.
- [85] G. Robertson, E. D. Lehmann, W. Sandham, and D. Hamilton, “Blood Glucose Prediction Using Artificial Neural Networks Trained with the AIDA Diabetes Simulator: A Proof-of-Concept Pilot Study,” *J. Electr. Comput. Eng.*, vol. 2011, pp. 1–11, 2011.
- [86] T. G. Dietterich, “Machine-Learning Research: Four Current Directions,” *AI Mag.*, vol. 18, no. 4, p. 97, 1997.
- [87] P. E. Black, “Greedy Algorithm,” *Dictionary of Algorithms and Data Structures*, 2005. [Online]. Available: <https://xlinux.nist.gov/dads/HTML/greedyalgo.html>. [Accessed: 15-Jan-2018].
- [88] A. Alessandri, C. Cervellera, and M. Sanguineti, “Design of Asymptotic Estimators: An Approach Based on Neural Networks and Nonlinear Programming,” *IEEE Trans. Neural Networks*, vol. 18, no. 1, pp. 86–96, Jan. 2007.
- [89] J. R. Quinlan, “Induction of decision trees,” *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986.
- [90] K. Hornik, M. Stinchcombe, and H. White, “Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks,” *Neural Networks*, vol. 3, no. 5, pp. 551–560, Jan. 1990.
- [91] T. M. Fragoso and F. L. Neto, “Bayesian model averaging: A systematic review and conceptual classification,” *arXiv preprint*, vol. 1509.08864, 2015.
- [92] M. F. Steel, “Bayesian model averaging and forecasting,” *Bull. EU US Inflat. Macroecon. Anal.*, vol. 200, pp. 30–41, 2011.
- [93] R. E. Schapire, “The strength of weak learnability,” *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, Jun. 1990.
- [94] Y. Freund and R. E. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.
- [95] M. Deckert, “Incremental Rule-Based Learners for Handling Concept Drift: An Overview,” *Found. Comput. Decis. Sci.*, vol. 38, no. 1, Jan. 2013.
- [96] L. Rokach and O. Maimon, “Data Mining with Decision Trees: Theory and Applications,” in *Machine Perception and Artificial Intelligence*, Singapore: World Scientific, 2008.
- [97] M. A. Arbib and S. Amari, “Dynamic Interactions in Neural Networks:

Model and Data,” in *Research Notes in Neural Computing*, B. Kosko, Ed. Berlin: Springer-Verlag, 1989.

- [98] B. Parmanto, P. W. Munro, and H. R. Doyle, “Reducing Variance of Committee Prediction with Resampling Techniques,” *Conn. Sci.*, vol. 8, no. 3–4, pp. 405–426, Dec. 1996.
- [99] R. E. Schapire and Y. Singer, “Improved Boosting Algorithms Using Confidence-rated Predictions,” *Mach. Learn.*, vol. 37, no. 3, pp. 297–336, 1999.
- [100] T. G. Dietterich and G. Bakiri, “Solving Multiclass Learning Problems via Error-Correcting Output Codes,” *Artif. Intell. Res.*, vol. 2, pp. 263–286, 1995.
- [101] R. E. Schapire, “Using output codes to boost multiclass learning problems,” in *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997, pp. 313–321.
- [102] J. F. Kolen and J. B. Pollack, “Back propagation is sensitive to initial conditions,” in *Proceedings of the 1990 conference on Advances in neural information processing systems*, 1990, pp. 860–867.
- [103] T. G. Dietterich, “An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization,” *Mach. Learn.*, vol. 40, no. 2, pp. 139–157, 2000.
- [104] Y. Raviv and N. Intrator, “Bootstrapping with noise: An effective regularization technique,” *Conn. Sci.*, vol. 8, no. 34, pp. 355–372, 1996.
- [105] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artif. Intell.*, vol. 97, no. 1–2, pp. 273–324, Dec. 1997.
- [106] G. H. John, R. Kohavi, and K. Pfleger, “Irrelevant features and the subset selection problem,” in *Proceedings of the 11th International Conference on Machine Learning*, 1994, pp. 121–129.
- [107] Huan Liu and Lei Yu, “Toward integrating feature selection algorithms for classification and clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005.
- [108] T. Hastie, T. Robert, and F. Jerome, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.
- [109] A. Jain and D. Zongker, “Feature selection: evaluation, application, and small sample performance,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 2, pp. 153–158, 1997.
- [110] D. Zongker and A. Jain, “Algorithms for feature selection: An evaluation,” in *Proceedings of 13th International Conference on Pattern Recognition*, 1996, pp. 18–22.
- [111] Y. Liao and J. E. Moody, “Constructing Heterogeneous Committees Using Input Feature Grouping: Application to Economic Forecasting,” *Adv. Neural Inf. Process. Syst.*, vol. 12, pp. 921–927, 2000.
- [112] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, *Cluster Analysis*, 5th ed. Wiley, 2011.
- [113] N. C. Oza and K. Tumer, “Input Decimation Ensembles: Decorrelation

through Dimensionality Reduction,” in *Multiple Classifier Systems*, 2001, pp. 238–247.

- [114] K. Turner and N. C. Oza, “Decimated input ensembles for improved generalization,” in *International Joint Conference on Neural Networks.*, 1999, vol. 5, pp. 3069–3074.
- [115] K. J. Cherkauer, “Human Expert-Level Performance on a Scientific Image Analysis Task by a System Using Combined Artificial Neural Networks,” in *Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms Wkshp, 13th Nat Conf on Artificial Intelligence*, 1996, pp. 15–21.
- [116] E. Stamatatos and G. Widmer, “Music Performer Recognition Using an Ensemble of Simple Classifiers,” in *Proceedings of the 15th European Conference on Artificial Intelligence*, 2002, pp. 335–339.
- [117] K. Tumer and J. Ghosh, “Error Correlation and Error Reduction in Ensemble Classifiers,” *Conn. Sci.*, vol. 8, no. 3–4, pp. 385–404, Dec. 1996.
- [118] Tin Kam Ho, “Random Decision Forests,” in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1995, vol. 1, pp. 278–282.
- [119] T. K. Ho, “The Random Subspace Method for Cosntructing Decision Forests,” *IEEE Trans. Pat- tern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, 1998.
- [120] R. J. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.
- [121] S. D. Bay, “Combining Nearest Neighbor Classifiers Through Multiple Feature Subsets,” in *Proceedings of the 17th International Conference on Machine Learning*, 1998, pp. 37–45.
- [122] F. M. Alkoot and J. Kittler, “Feature selection for an ensemble of classifiers,” in *Proceedings of the 4th Multiconference on Systematics, Cybernetics, and Informatics*, 2000, pp. 379–384.
- [123] S. Gunter and H. Bunke, “Creation of classifier ensembles for handwritten word recognition using feature selection algorithms,” in *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*, 2002, pp. 183–188.
- [124] P. Pudil, F. J. Ferri, J. Novovicova, and J. Kittler, “Floating Search Methods for Feature Selection with Nonmonotonic Criterion Functions,” in *Proceedings of the 12th IAPR International Conference on Pattern Recognition (Cat. No.94CH3440-5)*, 1994, vol. 2, pp. 279–283.
- [125] D. W. Opitz, “Feature Selection for Ensembles.,” in *Proceedings of the sixteenth national conference on Artificial intelligence*, 1999, pp. 379–384.
- [126] C. Guerra-Salcedo and D. Whitley, “Genetic approach to feature selection for ensemble creation,” in *Proceedings of Genetic and Evolutionary Computation Conference*, 1999, pp. 236–243.
- [127] L. J. Eshelman, “The CHC Adaptive Search Algorithm: How to Have

Safe Search When Engaging in Nontraditional Genetic Recombination,” in *Proceedings of the First Workshop on Foundations of Genetic Algorithms*, 1991, pp. 265–283.

- [128] C. Guerra-Salcedo and D. L. Whitley, “Genetic Search for Feature Subset Selection: A Comparison Between CHC and GENESIS,” in *Proceedings of the 3rd Conference on Genetic Programming*, 1998, pp. 504–509.
- [129] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, “Feature selection for ensembles: a hierarchical multi-objective genetic algorithm approach,” in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, 2003, vol. 1, pp. 676–680.
- [130] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, “Feature selection using multi-objective genetic algorithms for handwritten digit recognition,” in *Object recognition supported by user interaction for service robots*, 2002, vol. 1, pp. 568–571.
- [131] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley & Sons Inc, 1998.
- [132] J. C. Platt, “Fast Training of Support Vector Machines using Sequential Minimal Optimization,” in *Advances in Kernel Methods - Support Vector Learning*, MA, USA: MIT Press, 1999, pp. 185–208.
- [133] M. Pontil and a. Verri, “Support vector machines for 3D object recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 6, pp. 637–646, Jun. 1998.
- [134] V. Wan and W. M. Campbell, “Support vector machines for speaker verification and identification,” in *Neural Networks for Signal Processing X. Proceedings of the 2000 IEEE Signal Processing Society Workshop (Cat. No.00TH8501)*, vol. 2, no. C, pp. 775–784.
- [135] E. Osuna, R. Freund, and F. Girosit, “Training support vector machines: an application to face detection,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 130–136, 1997.
- [136] F. Schwenker, H. a. Kestler, and G. Palm, “Three learning phases for radial-basis-function networks,” *Neural Networks*, vol. 14, no. 4–5, pp. 439–458, May 2001.
- [137] L. Rokach and O. Maimon, “Decision Trees,” in *Data Mining and Knowledge Discovery Handbook*, Springer, 2005, pp. 165–192.
- [138] S. Kullback and R. A. Leibler, “On Information and Sufficiency,” *Ann. Math. Stat.*, vol. 22, no. 1, pp. 79–86, Mar. 1951.
- [139] W. N. H. W. Mohamed, M. N. M. Salleh, and A. H. Omar, “A comparative study of Reduced Error Pruning method in decision tree algorithms,” in *2012 IEEE International Conference on Control System, Computing and Engineering*, 2012, pp. 392–397.
- [140] M. Bramer, *Principles of Data Mining*. London: Springer London, 2007.
- [141] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, vol. 31. New York, NY: Springer New York, 1996.

- [142] W. W. Cohen, “Fast Effective Rule Induction,” in *Proceedings of the Twelfth International Conference on Machine Learning*, 1995, vol. 2435, pp. 115–123.
- [143] M. Lichman, “UCI Machine Learning Repository.” Irvine, CA: University of California, School of Information and Computer Science, 2013.
- [144] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes, “Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus,” in *Proc Annu Symp Comput Appl Med Care*, 1988, pp. 261–265.
- [145] M. Pazzani, C. Merz, P. Murphy, K. Ali, T. Hume, and C. Brunk, “Reducing Misclassification Costs,” in *International Conference of Machine Learning*, 1994, pp. 217–225.
- [146] P. Domingos, “MetaCost: A General Method for Making Classifiers Cost-Sensitive,” in *International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 155–164.
- [147] M. Kubat and S. Matwin, “Addressing the Curse of Imbalanced Training Sets: One-Sided Selection,” in *International Conference on Machine Learning*, 1997, vol. 4, pp. 179–186.
- [148] N. Japkowicz, “The Class Imbalance Problem: Significance and Strategies,” in *International Conference on Artificial Intelligence (IC-AI’2000): Special Track on Inductive Learning*, 2000, vol. 8.
- [149] D. D. Lewis, J. Catlett, and M. Hill, “Heterogeneous Uncertainty Sampling for Supervised Learning,” in *Eleventh International Conference of Machine Learning*, 1994, pp. 148–156.
- [150] C. X. Ling and C. Li, “Data Mining for Direct Marketing : Problems and Solutions,” in *American Association for Artificial Intelligence*, 1998, pp. 73–79.
- [151] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE : Synthetic Minority Over-sampling Technique,” *J. Artificial Intell. Res.*, vol. 16, pp. 321–357, 2002.
- [152] P. A. Lachenbruch and M. R. Mickey, “Estimation of error rates in discriminant analysis,” *Technometrics*, vol. 10, no. 1, pp. 1–12, 1968.
- [153] G. J. McLachlan, K.-A. Do, and C. Ambrose, *Analyzing Microarray Gene Expression Data*. Wiley, 2005.
- [154] S. Raschka, “K-fold Cross Validation question: statistics,” *Reddit.com*, 2016. [Online]. Available: <http://sebastianraschka.com/images/faq/evaluate-a-model/k-fold.png>. [Accessed: 26-Jul-2017].
- [155] G. Gong, *Cross-validation, the Jackknife, and the Bootstrap: Excess Error Estimation in Forward Logistic Regression*. Stanford University, 1982.
- [156] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

- [157] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
- [158] W. Zhu, N. Zeng, and N. Wang, "Sensitivity, Specificity, Accuracy, Associated Confidence Interval And ROC Analysis With Practical SAS Implementations," in *Northeast SAS Users Group*, 2010.
- [159] D. M. W. Powers, "Evaluation: From Precision, Recall and F-Measure To Roc, Informedness, Markedness & Correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.
- [160] J. A. Swets, "Measuring the accuracy of diagnostic systems.," *Science*, vol. 240, no. 4857, pp. 1285–93, Jun. 1988.
- [161] S. S. Lee, "Noisy replication in skewed binary classification," *Comput. Stat. Data Anal.*, vol. 34, no. 2, pp. 165–191, Aug. 2000.
- [162] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Pearson, 2013.
- [163] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [164] E. Fix and J. L. Hodges, "An important contribution to nonparametric discriminant analysis and density estimation," *Int. Stat. Rev.*, vol. 57, no. 3, pp. 233–247, 1951.
- [165] K. Hechenbichler and K. Schliep, "Weighted k-Nearest-Neighbor Techniques and Ordinal Classification," *Sonderforschungsbereich*, vol. 386, 2004.
- [166] R. Dechter and J. Pearl, "Generalized best-first search strategies and the optimality of A*," *J. ACM*, vol. 32, no. 3, pp. 505–536, Jul. 1985.
- [167] J. Pearl, *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Boston: Addison-Wesley, 1984.
- [168] M. A. Hall, "Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning," in *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp. 359–366.
- [169] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.
- [170] D. C. Klonoff, B. Buckingham, J. S. Christiansen, V. M. Montori, W. V. Tamborlane, R. a Vigersky, and H. Wolpert, "Continuous glucose monitoring: an Endocrine Society Clinical Practice Guideline.," *J. Clin. Endocrinol. Metab.*, vol. 96, no. 10, pp. 2968–79, Oct. 2011.
- [171] E. Deza and M. M. Deza, *Encyclopedia of Distances*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.
- [172] D. Coomans and D. L. Massart, "Alternative k-nearest neighbour rules in supervised pattern recognition," *Anal. Chim. Acta*, vol. 136, pp. 15–27, 1982.
- [173] B. Santoso, H. Wijayanto, K. A. Notodiputro, and B. Sartono, "Synthetic Over Sampling Methods for Handling Class Imbalanced Problems: A Review," in *IOP Conference Series on Earth and Environmental Sciences*, 2017, pp. 1–8.

- [174] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, and A. Hussain, "Comparing Oversampling Techniques to Handle the Class Imbalance Problem: A Customer Churn Prediction Case Study," *IEEE Access*, vol. 4, pp. 7940–7957, 2016.
- [175] R. Longadge, S. S. Dongre, and L. Malik, "Class Imbalance Problem in Data Mining: Review," *Int. J. Comput. Sci. Netw.*, vol. 2, no. 1, pp. 1–6, 2013.
- [176] X. Guo, Y. Yin, C. Dong, G. Yang, and G. Zhou, "On the Class Imbalance Problem," in *2008 Fourth International Conference on Natural Computation*, 2008, pp. 192–201.
- [177] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explor. Newsl.*, vol. 6, no. 1, p. 20, Jun. 2004.
- [178] I. Tomek, "Two Modifications of CNN," *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-6, no. 11, pp. 769–772, Nov. 1976.
- [179] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, "SMOTE-RSB *: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory," *Knowl. Inf. Syst.*, vol. 33, no. 2, pp. 245–265, Nov. 2012.
- [180] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," in *International Conference on Intelligent Computing*, 2005, pp. 878–887.
- [181] C. Bunkhumpornpat and C. Sinapiromsaran, Krung Lursinsap, "Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2009, pp. 475–482.
- [182] Diabetes UK, "Diabetes : Facts and Stats," 2014. [Online]. Available: <https://www.diabetes.org.uk/resources-s3/2017-11/diabetes-key-stats-guidelines-april2014.pdf>. [Accessed: 25-Jan-2018].
- [183] R. Parikh, A. Mathai, S. Parikh, G. Chandra Sekhar, and R. Thomas, "Understanding and using sensitivity, specificity and predictive values.," *Indian J. Ophthalmol.*, vol. 56, no. 1, pp. 45–50, 2008.
- [184] S. E. Chandra, N. Pavithra, and P. Saikumar, "Skin Fold Thickness in Diabetes Mellitus: A Simple Anthropometric Measurement May Bare the Different Aspects of Adipose Tissue.," *J. Dent. Med. Sci.*, vol. 15, no. 11, pp. 7–11, 2016.
- [185] D. S. Freedman, P. T. Katzmarzyk, W. H. Dietz, S. R. Srinivasan, and G. S. Berenson, "Relation of body mass index and skinfold thicknesses to cardiovascular disease risk factors in children: the Bogalusa Heart Study," *Am. J. Clin. Nutr.*, vol. 90, no. 1, pp. 210–216, Jul. 2009.
- [186] P. Zuchinali, G. C. Souza, F. D. Alves, K. S. M. D'Almeida, L. A. Goldraich, N. O. Clausell, and L. E. P. Rohde, "Triceps Skinfold as a Prognostic Predictor in Outpatient Heart Failure," *Arq. Bras. Cardiol.*, vol. 101, no. 5, pp. 434–441, 2013.

- [187] A. Wang, G. Chen, Z. Su, X. Liu, X. Liu, H. Li, Y. Luo, L. Tao, J. Guo, L. Liu, S. Chen, S. Wu, and X. Guo, "Risk scores for predicting incidence of type 2 diabetes in the Chinese population: the Kailuan prospective study," *Sci. Rep.*, vol. 6, no. 1, p. 26548, Sep. 2016.
- [188] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, Jun. 1947.
- [189] A. F. Clark and C. Clark, "Performance Characterization in Computer Vision," 2004. [Online]. Available: <http://www.music.mcgill.ca/~ich/classes/mumt611/Evaluation/tutorial.pdf>.
- [190] R. M. Haralick, "Performance Characterization in Computer Vision," in *BMVC92*, London: Springer London, 1992, pp. 1–8.
- [191] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman and Hall/CRC, 2011.
- [192] G. S. Collins, S. Mallett, O. Omar, and L.-M. Yu, "Developing risk prediction models for type 2 diabetes: a systematic review of methodology and reporting.," *BMC Med.*, vol. 9, no. 1, p. 103, Jan. 2011.
- [193] M. C. Monard and G. E. A. P. A. Batista, "Learning with Skewed Class Distributions," *LAPTEC*, pp. 1–9, 2003.
- [194] R. D. Wilson and T. R. Martinez, "Improved Heterogeneous Distance Functions," *J. Artif. Intell. Res.*, vol. 6, pp. 1–34, 1997.
- [195] G. Lemaitre, F. Nogueira, D. Oliveira, and C. Aridas, "SMOTE + Tomek," 2016. [Online]. Available: http://contrib.scikit-learn.org/imbalanced-learn/auto_examples/combine/plot_smote_tomek.html. [Accessed: 27-Jul-2017].
- [196] G. Lemaitre, F. Nogueira, D. Oliveira, and C. Aridas, "Tomek Links," 2016. [Online]. Available: http://contrib.scikit-learn.org/imbalanced-learn/auto_examples/under-sampling/plot_tomek_links.html. [Accessed: 27-Jul-2017].
- [197] G. Lemaitre, F. Nogueira, D. Oliveira, and C. Aridas, "SMOTE + ENN," 2016. [Online]. Available: http://contrib.scikit-learn.org/imbalanced-learn/auto_examples/combine/plot_smote_enn.html. [Accessed: 27-Jul-2017].
- [198] R. C. Prati, G. E. A. P. A. Batista, and M. C. Monard, "Data mining with imbalanced class distributions: concepts and methods," in *Indian International Conference on Artificial Intelligence*, 2009, pp. 359–376.
- [199] A. D. Pozzolo, O. Caelen, and G. Bontempi, "Comparison of balancing techniques for unbalanced datasets," *Machine Learning Group, Universite Libre Bruxelles Belgium*, 2013. [Online]. Available: http://www.ulb.ac.be/di/map/adalpozz/pdf/poster_unbalanced.pdf. [Accessed: 27-Jul-2017].

- [200] J. Bergstra and Y. Bengio, “Random Search for Hyper-Parameter Optimization,” *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.
- [201] N. Nnamoko, F. Arshad, D. England, J. Vora, and J. Norman, “Evaluation of Filter and Wrapper Methods for Feature Selection in Supervised Machine Learning,” in *PGNET*, 2014, pp. 63–67.
- [202] S. Das, “Filters , Wrappers and a Boosting-Based Hybrid for Feature Selection,” in *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001, pp. 74–81.
- [203] L. Talavera, “An evaluation of filter and wrapper methods for feature selection in categorical clustering,” in *Advances in Intelligent Data Analysis VI*, 2005, pp. 440–451.
- [204] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
- [205] R. Sedgewick, *Algorithms*. Addison-Wesley, 2011.
- [206] M. Mitchell, J. H. Holland, and S. Forrest, “When Will a Genetic Algorithm Outperform Hill Climbing?,” in *Advances in Neural Information Processing Systems 6*, J. D. Cowan, G. Tesauro, and J. Alspector, Eds. 1993, pp. 51–58.
- [207] B. Xi, Z. Liu, M. Raghavachari, C. H. Xia, and L. Zhang, “A smart hill-climbing algorithm for application server configuration,” in *Proceedings of the 13th conference on World Wide Web - WWW '04*, 2004, p. 287.

APPENDIX A.1

Detailed description of the experimental datasets, including the source information and data characteristics.

Pima Indians Diabetes Dataset

Source:

National Institute of Diabetes and Digestive and Kidney Diseases

Donor to UCI database:

Vincent Sigillito (vgs@aplcn.apl.jhu.edu)

Dataset Information:

This data contains 768 samples of diabetes examination results that can be used to judge the risk of developing diabetes within 5 years. The goal is to classify the patient into one of two categories, “positive and negative”. This data set includes 500 instances of “negative” and 268 instances of “positive”. The instances are described by 9 attributes.

Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

Attribute Information:

1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (μ U/ml)
6. Body mass index ($\text{weight in kg}/(\text{height in m})^2$)
7. Diabetes pedigree function
8. Age (years)
9. Class variable (0 or 1)

The providers indicated that there were no missing values in the dataset. However, this cannot be true as there are zeros in places where they are biologically impossible, such as the blood pressure attribute. It seems very likely that zero values encode missing data. Since the dataset donors made no such statement, users are encouraged to use their best judgement and state their assumptions.

APPENDIX A.2

Detailed description of SMOTE algorithm

```

Input: Number of minority class samples  $T$ ; Amount of SMOTE  $N\%$ ; Number
of nearest neighbours  $k$ 
Output:  $(N/100)*T$  synthetic minority class samples
(* If  $N$  is less than 100%, randomize the minority class samples as
only a random percent of them will be SMOTEd. *)
if  $N < 100$ 
    then Randomize the  $T$  minority class samples
         $T = (N/100) * T$ 
         $N = 100$ 
endif
 $N = \text{int}(N/100)$ 
(* The amount of SMOTE is assumed to be in integral multiples of 100.
*)
 $k$  = Number of nearest neighbours
 $\text{numattrs}$  = Number of attributes
 $\text{Sample}[] []$ : array for original minority class samples
 $\text{newindex}$ : keeps a count of number of synthetic samples generated,
initialized to 0
 $\text{Synthetic}[] []$ : array for synthetic samples
(* Compute  $k$  nearest neighbours for each minority class sample only.
*)
for  $j \leftarrow 1$  to  $T$ 
    Compute  $k$  nearest neighbors for  $j$ , and save the indices in the
     $\text{nnarray}$ 
    Populate( $N, j, \text{nnarray}$ )
endfor
Populate( $N, j, \text{nnarray}$ )
(* Function to generate the synthetic samples. *)
while  $N \neq 0$ 
    Choose a random number between 1 and  $k$ , call it  $\text{nn}$ . This step
    chooses one of the  $k$  nearest neighbours of  $j$ .
    for  $\text{attr} \leftarrow 1$  to  $\text{numattrs}$ 
        Compute:  $\text{dif} = \text{Sample}[\text{nnarray}[\text{nn}]][\text{attr}] - \text{Sample}[j][\text{attr}]$ 
        Compute:  $\text{gap} = \text{random number between } 0 \text{ and } 1$ 
         $\text{Synthetic}[\text{newindex}][\text{attr}] = \text{Sample}[j][\text{attr}] + \text{gap} * \text{dif}$ 
    endfor
     $\text{newindex}++$ 
     $N = N - 1$ 
endwhile
return
(* End of Populate. *)

```

Figure A.2.0.1: SMOTE algorithm (source: [151])

Provided below, is an example of how random synthetic samples are calculated from the sample vector.

Consider a sample (6,4) and let (4,3) be its nearest neighbour.

(6,4) is the sample for which k-nearest neighbours are being identified.
(4,3) is one of its k-nearest neighbours.

Let:

$f1_1 = 6$ $f2_1 = 4$ so $f2_1 - f1_1 = -2$
 $f1_2 = 4$ $f2_2 = 3$ so $f2_2 - f1_2 = -1$

The new samples will be generated as
 $(f1', f2') = (6,4) + \text{rand}(0-1) * (-2, -1)$

Note: $\text{rand}(0-1)$ generates a random number between 0 and 1.

Here, synthetic samples are generated, by taking the difference between the feature vector (sample) under consideration and its nearest neighbour and multiplying it by a random number between 0 and 1. The resultant value is then added to the feature vector under consideration. This approach effectively forces the decision region of the minority class to become more general by creating larger and less specific decision regions; rather than smaller and more specific regions created through sampling with replacement. As a result, more general regions are now learned for the minority class samples instead of those being subsumed previously by the majority class samples around them. The effect is that classifiers generalize better on the dataset.

APPENDIX A.3

Performance with unbalanced vs balanced dataset for each base classifier. Note that unbalanced dataset consists of 419 instances of which 279 tested negative and 140 tested positive. The balanced dataset consists of 559 instances of which 279 tested negative and 280 tested positive. Comparison with McNemar's test is impossible due to the difference in data size.

Table A.3.0.1: Tabular representation of Naïve Bayes performance on balanced vs unbalanced dataset

	Unbalanced data	Balanced data
True positive (count)	91	201
False negative (count)	49	79
True negative (count)	229	216
False Positive (count)	50	63
Classified correctly (count)	320	417
Classified incorrectly (count)	99	142
Accuracy (%)	76	75
Sensitivity (%)	65	72
Specificity (%)	82	77
AUC (%)	84	83

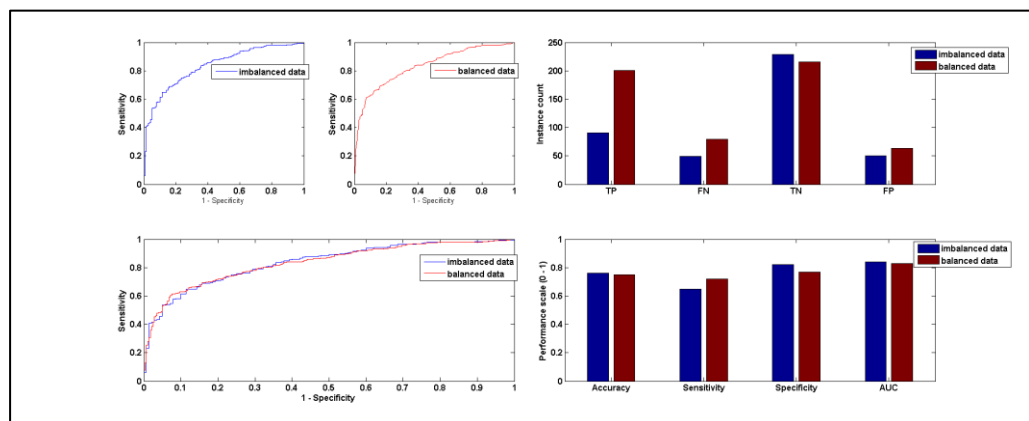


Figure A.3.0.1: Graphic representation of Naïve Bayes performance on balanced vs unbalanced dataset

Table A.3.0.2: Tabular representation of RBF performance on balanced vs unbalanced dataset

	Unbalanced data	Balanced data
True positive (count)	82	229
False negative (count)	58	51
True negative (count)	242	205
False Positive (count)	37	74
Classified correctly (count)	324	434
Classified incorrectly (count)	95	125
Accuracy (%)	77	78
Sensitivity (%)	59	82
Specificity (%)	87	73
AUC (%)	83	85

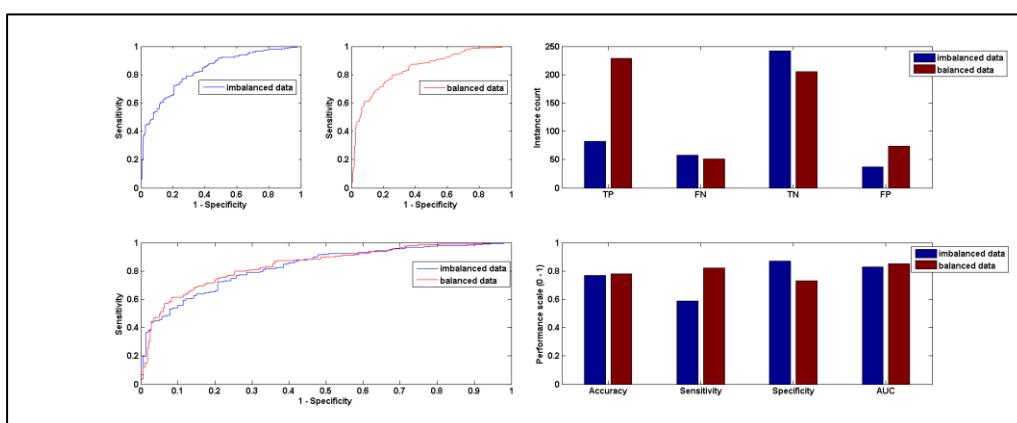


Figure A.3.0.2: Graphic representation of RBF performance on balanced vs unbalanced dataset

Table A.3.0.3: Tabular representation of SMO performance on balanced vs unbalanced dataset

	Unbalanced data	Balanced data
True positive (count)	81	206
False negative (count)	59	74
True negative (count)	246	219
False Positive (count)	33	60
Classified correctly (count)	327	425
Classified incorrectly (count)	92	134
Accuracy (%)	78	76
Sensitivity (%)	58	74
Specificity (%)	88	78
AUC (%)	84	85

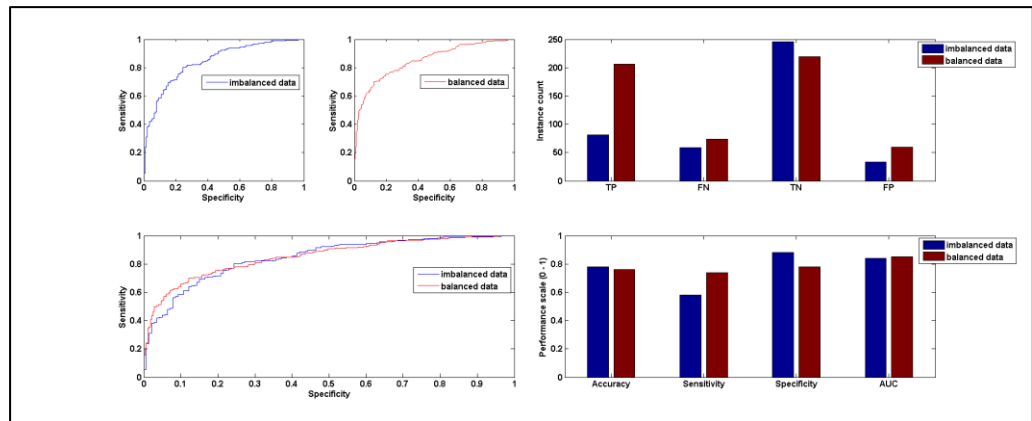


Figure A.3.0.3: Graphic representation of SMO performance on balanced vs unbalanced dataset

Table A.3.0.4: Tabular representation of C4.5 performance on balanced vs unbalanced dataset

	Unbalanced data	Balanced data
True positive (count)	100	226
False negative (count)	40	54
True negative (count)	228	207
False Positive (count)	51	72
Classified correctly (count)	328	433
Classified incorrectly (count)	91	126
Accuracy (%)	78	77
Sensitivity (%)	71	81
Specificity (%)	82	74
AUC (%)	76	79

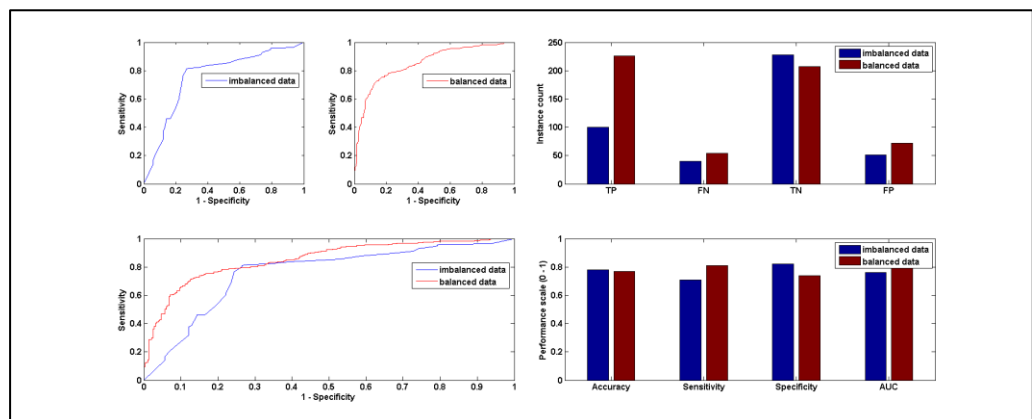


Figure A.3.0.4: Graphic representation of c4.5 performance on balanced vs unbalanced dataset

Table A.3.0.5: Tabular representation of RIPPER performance on balanced vs unbalanced dataset

	Unbalanced data	Balanced data
True positive (count)	95	223
False negative (count)	45	57
True negative (count)	231	214
False Positive (count)	48	65
Classified correctly (count)	326	437
Classified incorrectly (count)	93	122
Accuracy (%)	78	78
Sensitivity (%)	68	80
Specificity (%)	83	77
AUC (%)	74	79

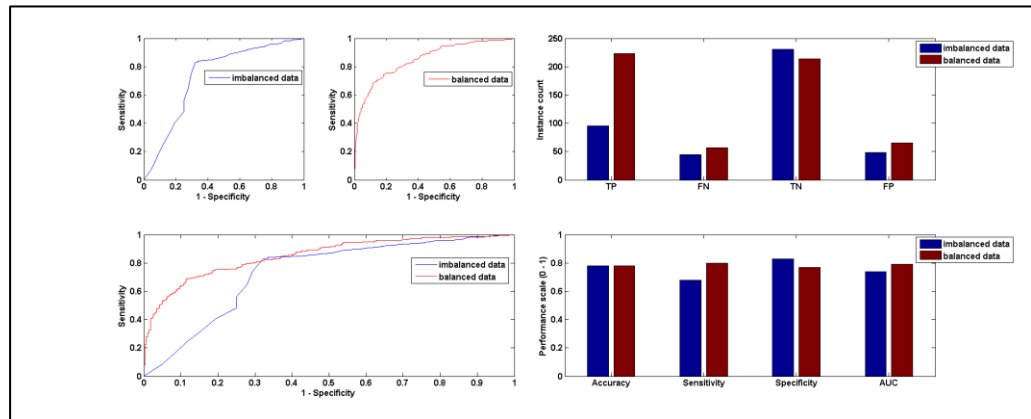


Figure A.3.0.5: Graphic representation of RIPPER performance on balanced vs unbalanced dataset

APPENDIX A.4

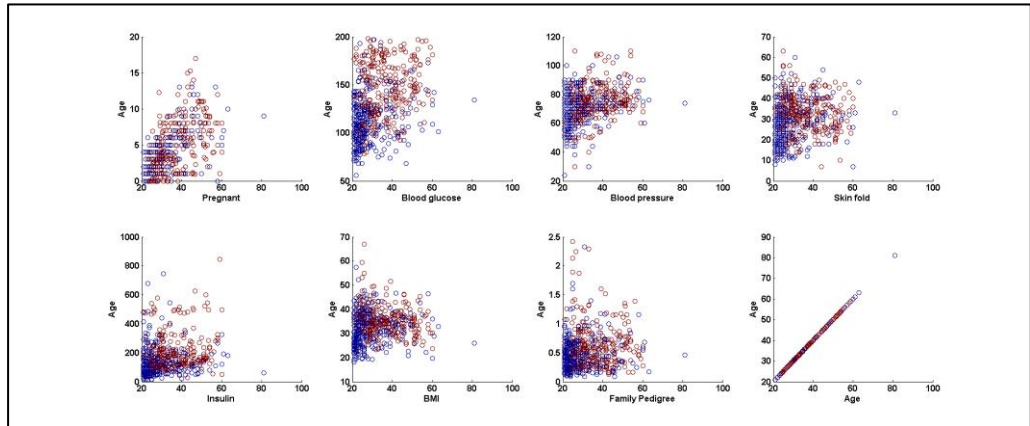


Figure A.4.0.1: Data cluster of 'age' and other features of the training dataset

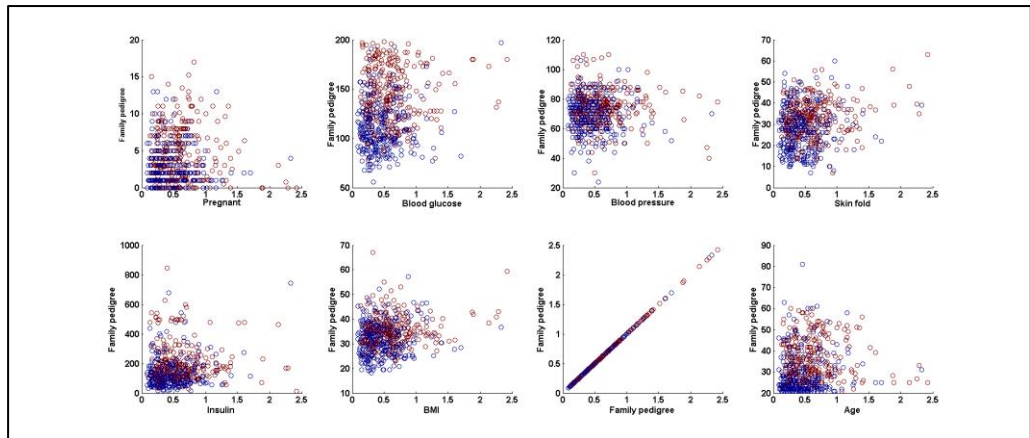


Figure A.4.0.2: Data cluster of 'family pedigree' and other features of the training dataset

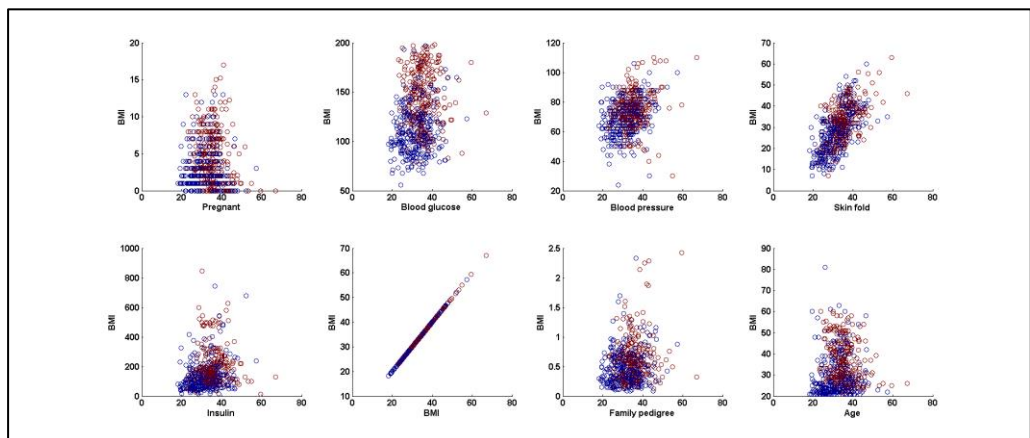


Figure A.4.0.3: Data cluster of 'bmi' and other features of the training dataset

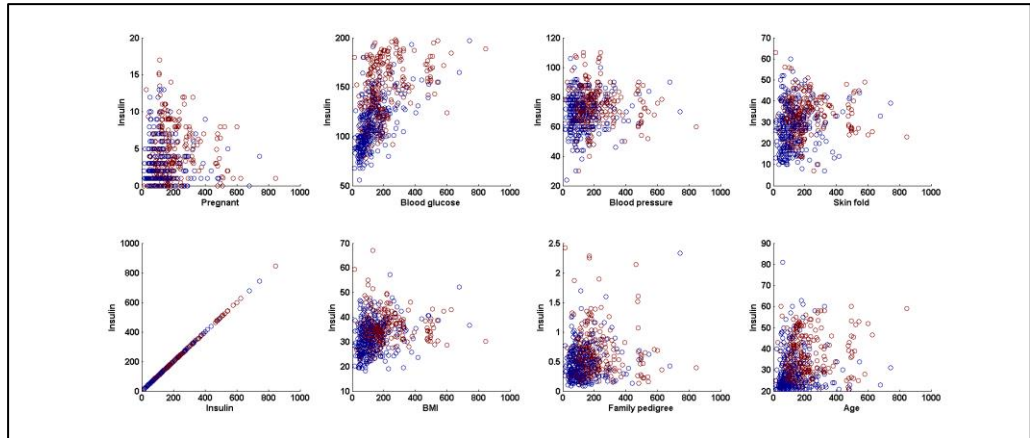


Figure A.4.0.4: Data cluster of 'insulin' and other features of the training dataset

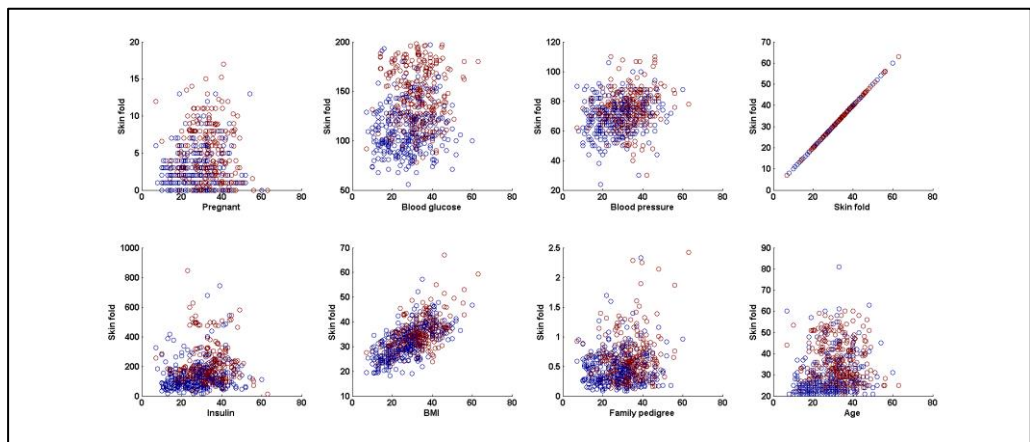


Figure A.4.0.5: Data cluster of 'skin fold' and other features of the training dataset

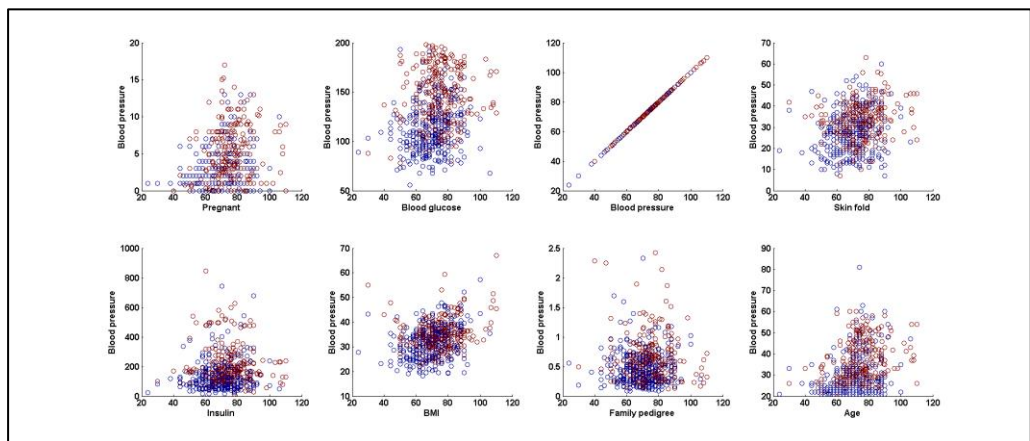


Figure A.4.0.6: Data cluster of 'blood pressure' and other features of the training dataset

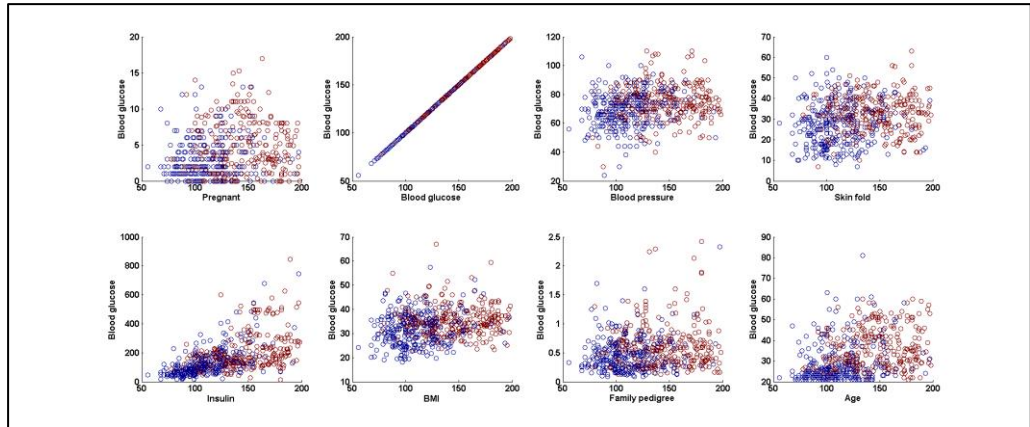


Figure A.4.0.7: Data cluster of 'blood glucose' and other features of the training dataset

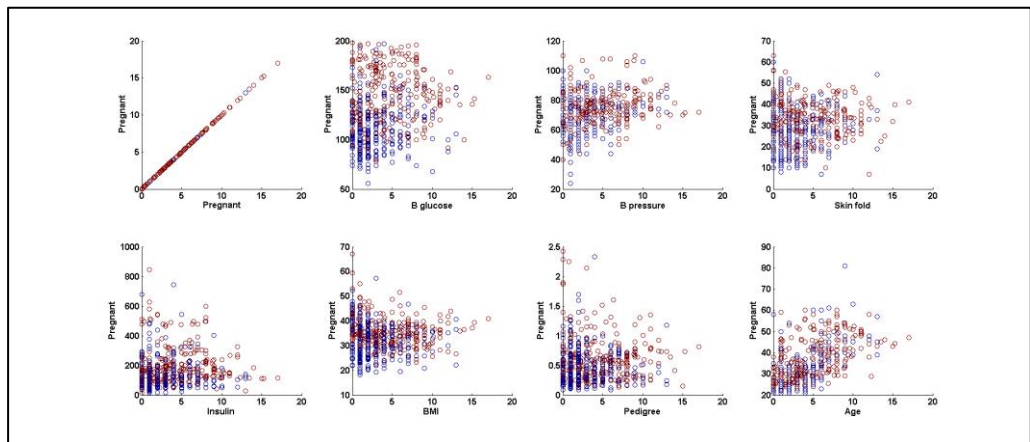


Figure A.4.0.8: Data cluster of 'pregnant' and other features of the training dataset

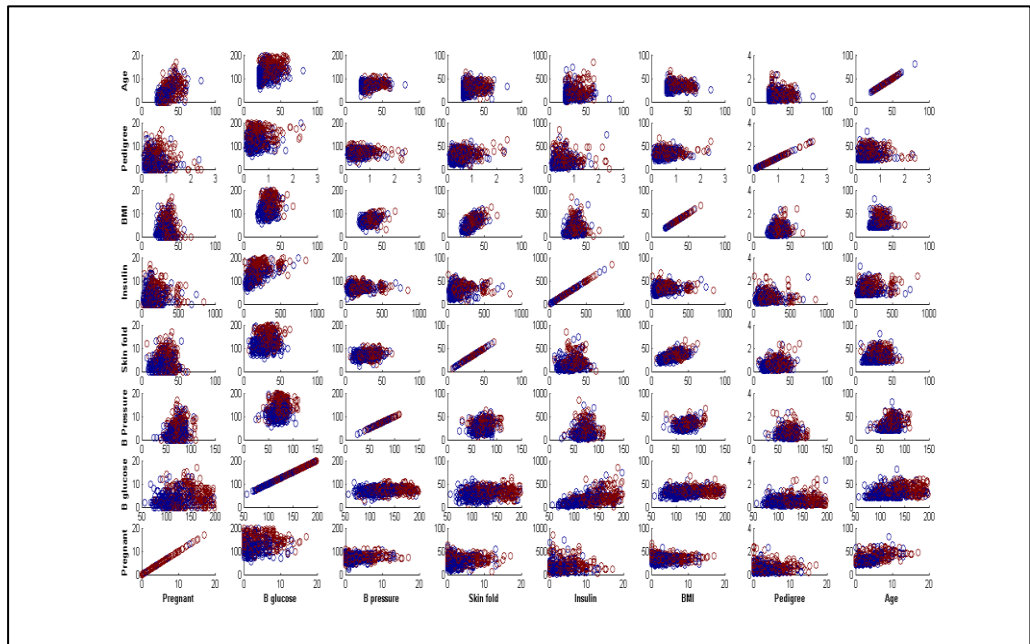


Figure A.4.0.9: Scatter plot of the experimental dataset showing class distribution and density