EFFECT OF FEATURE SELECTION AND DATASET SIZE ON THE ACCURACY OF NAÏVE BAYESIAN CLASSIFIER AND LOGISTICS REGRESSION

BY

ANAKOBE, Muhammad Bashir

P14SCMT8017

Department of Statistics,

Faculty of Physical Sciences,

Ahmadu Bello University,

Zaria

SUPERVISORY COMMITTEE:

Dr. A. Yahaya (Chairman)

Dr. S.I.IS. Doguwa (Member)

June, 2018

DECLARATION

I declare that the work in this dissertation entitled "EFFECT OF FEATURE SELECTION AND DATASET SIZE ON THE ACCURACY OF NAÏVE BAYESIAN CLASSIFIER AND LOGISTICS REGRESSION" has been performed by me in the Department of Statistics under the supervision of Dr. A. Yahaya. The information derived from literature has been duly acknowledged in the text and a list of reference provided. No part of this was previously presented for another degree or diploma at any University or Institution.

ANAKOBE Muhammad Bashir	_____	_____
Name of Student	Signature	Date

CERTIFICATION

This Dissertation titled "**Effect Of Feature Selection and Dataset Size On the Accuracy Of Naïve Bayesian Classifier And Logistics Regression**" by Muhammad Bashir ANAKOBE (P14SCMT8003) meets the regulations governing the award of the Master of Science of Ahmadu Bello University, Zaria and is approved for its contribution to knowledge and literary presentation.

Dr. A. Yahaya	_____	_____
Chairman, Supervisory Committee	Signature	Date
Dr. S.I.S. Doguwa	_____	_____
Member, Supervisory Committee	Signature	Date
Dr. Y. Musa	_____	_____
External Examiner	Signature	Date
Dr. H.G. Dikko	_____	_____
Head of Department	Signature	Date
Prof. S.Z. Abubakar	_____	_____
Dean, School of Postgraduate Studies	Signature	Date

DEDICATION

To my beloved parents, Mrs Rekiyat Abdulkareem and Late Mr. Anakobe Abdulkareem, whose love and passion for growth and development have taken me this far.

ACKNOWLEDGEMENT

My profound gratitude goes to ALLAH, the Greatest on whom my life solely depends and by whose will this dream became a reality.

My sincere appreciation goes to my supervisors, Dr. A. Yahaya and Dr. S.I.S. Doguwa, for their efforts to make this work a success. The motivation and corrections have indeed given me a higher perspective on research and discipline.

I want to use this medium to appreciate my lecturers; Dr. H.G. Dikko, Prof. S.B. Junaidu, Dr. H.M. Jibril and all other lecturers in the Department of Statistics and their colleagues from other Departmrnts for their guidance, great collaboration and dedication towards a smooth and successful completion of this programme. I humbly acknowledge and appreciate the selflessness and wonderful love and care of my family members, my mother Mrs. Rekiyat Abdulkareem, Brothers and Sisters, Uncles and Aunties, Cousins, Nieces and Nephews. I would not forget my good friends, Yahaya J. Danjuma, Rita Tajan, Mrs Aisha, Zainab Izumafe, Bello Kizito Eneye, Musa Makama, Treasure Tani Yarima, Umar Kabir and to the entire PG (P14SCMT) students of Department of Statistics, thank you all.

ABSTRACT

Binary Logistics Regression and Naïve Bayesian classifier are two of the common classification modelling techniques that allow one to predict the category that a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. We studied the classification performances of the two linear classification under different feature (variable) selection criteria and dataset size conditions on a medical domain area were studied based on the datasets (breast cancer and heart diseases) obtained from the University of California, Irvine, online respiratory. The result indicated that logistics Regression for classification on relatively large datasets without the application of PCA (for variable selection) has the great accuracy (91.4%), while Naïve Bayesian classifier with PCA (for variable/ feature selection) tops the smaller dataset classification with an accuracy of 90.2%. These two accuracies are close enough and high enough, which is an indication of high relevance of their selections in solving classification problems on datasets from this kind of domain.

TABLE OF CONTENTS

TITLE PAGE.. i

DECLARATION.. ii

CERTIFICATION iii

DEDICATION iv

ACKNOWLEDGEMENT v

ABSTRACT vi

TABLE OF CONTENTS vi

LIST OF TABLES x

LIST OF FIGURES xi

CHAPTER ONE: INTRODUCTION 1

1.1 Background to the Study 1

1.2 Statement of the Problem 3

1.3 Aim and Objectives of the Study 3

1.4 Significance of the Study 4

1.5 Motivation 4

1.6 Scope and Limitation of the Study 4

1.7 Definition of Terms 5

CHAPTER TWO: LITERATURE REVIEW 8

CHAPTER THREE: RESEARCH METHODOLOGY AND MATERIALS 14

3.1 Introduction 14

3.2 Source of Data 14

3.3 Method of data analysis 14

3.4 Principal component analysis (PCA) 14

3.5 Logistics Regression 17

3.5.1 Binary Logit model from the Logistic Function 17

3.6 Naïve Bayesian Classifier 20
3.6.1 Formulation of the model 20
3.6.2 Learning the model: 21
3.7 Classification of New Data 22
3.8 Model's performance evaluation 22
3.8.1 Confusion Matrix 22
3.8.2 Comparing multiple models 23
CHAPTER FOUR: ANALYSIS AND DISCUSSION 25
4.1 Introduction 25
4.2 Model Building and evaluation for the Breast Cancer Dataset 25
4.2.1 Building and evaluating NB on the larger dataset, with no PCA 26
4.2.2 Building and evaluating NB on the smaller dataset, with no PCA 26
4.2.3 Building and evaluating LR on the larger dataset, with no PCA 27
4.2.4 Building and evaluating LR on the smaller dataset, with no PCA 28
4.2.5 Building and evaluating NB on the larger dataset, with PCA 29
4.2.6 Building and evaluating NB on the smaller dataset, with PCA 30
4.2.7 Building and evaluating LR on the larger dataset, with PCA 31
4.2.8 Building and evaluating LR on the smaller dataset, with PCA 31
4.3 Model Building and evaluation for the Heart Disease Dataset 32
4.3.1 Building and evaluating NB on the larger dataset, with no PCA 33
4.3.2 Building and evaluating NB on the smaller dataset, with no PCA 34
4.3.3 Building and evaluating LR on the larger dataset, with no PCA 34
4.3.4 Building and evaluating LR on the smaller dataset, with no PCA 35
4.3.5 Building and evaluating NB on the larger dataset, with PCA 36
4.3.6 Building and evaluating NB on the smaller dataset, with PCA 37
4.3.7 Building and evaluating LR on the larger dataset, with PCA 38
4.3.8 Building and evaluating LR on the smaller dataset, with PCA 38

4.4 Summary of Results 39

CHAPTER FIVE: SUMMARY, CONCLUSION AND RECOMMENDATION .. 42

5.1 Summary 42

5.2 Conclusion 43

5.3 Recommendation 43

5.4 Recommendation and Suggestion for future research 44

5.5 Contribution to knowledge 44

REFERENCES 44

APENDIX I 46

APENDIX II 62

LIST OF TABLES

Table 1. 1: Sample of a Confusion Matrix 6
Table 3. 1: 2x2 Confusion Matrix for Classification Result 23
Table 4. 1: Confusion Matrix for NB on the larger dataset, with no PCA 26
Table 4. 2: Confusion Matrix for NB on the smaller dataset, with no PCA 27
Table 4. 3: Confusion Matrix for LR on the larger dataset, with no PCA 27
Table 4. 4: Confusion Matrix for LR on the smaller dataset, with no PCA 28
Table 4. 5: Confusion Matrix for NB on the larger dataset, with PCA 30
Table 4. 6: Confusion Matrix for NB on the smaller dataset, with PCA 30
Table 4. 7: Confusion Matrix for LR on the larger dataset, with PCA 31
Table 4. 8: Confusion Matrix for LR on the smaller dataset, with PCA 31
Table 4. 9: Confusion Matrix for NB on the larger dataset, with no PCA 33
Table 4. 10: Confusion Matrix for NB on the smaller dataset, with no PCA 34
Table 4. 11: Confusion Matrix for LR on the larger dataset, with no PCA 35
Table 4. 12: Confusion Matrix for LR on the smaller dataset, with no PCA 35
Table 4. 13: Confusion Matrix for NB on the larger dataset, with PCA 37
Table 4. 14: Confusion Matrix for NB on the smaller dataset, with PCA 37
Table 4. 15: Confusion Matrix for LR on the larger dataset, with PCA 38
Table 4. 16: Confusion Matrix for LR on the smaller dataset, with PCA 38
Table 4. 17: A table showing the Classification Accuracies of NB and LR for the Breast Cancer Datasets, with their respective sensitivities and specificities. 39
Table 4. 18: A table showing the Classification Accuracies of NB and LR for the Breast Cancer Datasets, with their respective sensitivities and specificities. 40
Table 4. 19: A table showing the Average Classification Accuracies of NB and LR for the two datasets. 41

LIST OF FIGURES

Figure 4. 1: A scree plot showing the levels at which each components explains the variability. 29
Figure 4. 2: A scree plot showing the levels at which each components explains the variability. 36
Figure 4. 3: A multiple Bar Chart showing the Classification Accuracies of NB and LR for the Breast Cancer Datasets. 40
Figure 4. 4: A multiple Bar Chart showing the Classification Accuracies of NB and LR for the Heart Disease Datasets. 41

CHAPTER ONE: INTRODUCTION

1.1 Background to the Study

We start by considering the following problem: suppose you are a medical laboratory technologist, who has access to a patient's health records, who was admitted for heart disease diagnosis, the natural question that comes to mind is, does he or she has a heart disease or not? Or this one: suppose you are a bank, and given a person who wants to take out a loan, will she default on the loan? Or this: how can your email server tell which emails are spam and which ones are actual mail?

Intuitively, all of the above situations can be resolved by examining empirical data and taking out the factors that are important in each. For example, in the heart disease case, one might want to look through hospitalization records of patients who have had heart diseases and see if your patient resembles them in age, blood pressure, body temperature, diet and exercise habits, family history and other clinical measurements.

The above situations are examples of classification problems. Classification is a statistical method used to build predicative models to separate and classify new data points. In Machine Learning and Statistics, classification is the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known.

Feature (variable) selection is the process of identifying and removing as many irrelevant and redundant features as possible from dataset features (Yu and Liu, 2004). This reduces the dimensionality of the data and enables data mining algorithms to operate effectively. The fact that many features depend on one another often unduly influences the accuracy of models.

Classification models are affected by the choice of features (variables). Purkayastha*et al* (2014) stated that: selecting the *relevant features* for classification is significant for a variety of reasons like simplification of performance, computational efficiency, and feature interpretability.

A model which performs classification is known as a classifier. A classifier is a function which maps an input variable *X* to a class *C*. Classifiers are broadly divided into linear and non-linear classifiers: with the linear classifiers, models which are based on the linear combination of variables' values are built. Linear classifiers work well for practical problems such as medical diagnosis, document classification, and more generally for problems with many variables (features), reaching accuracy levels comparable to non-linear classifiers while taking less time to learn from the training dataset.

There are numerous classifiers today and the choice of which to use depends on a number of factors; for example, the simplicity, *accuracy and applicability to the domain and structure of dataset under consideration,* Kwon and Mun Sim (2013). In this research work, we focused on Discriminative-Generative pair of binary linear classifiers as typified by Binary Logistics Regression (LR) and Naïve Bayes Classifier (NBC).

Bayesian Classification represents a supervised learning method for classification. Naïve Bayes classifier is a kind of probabilistic classifier that is based on applying Bayes' theorem. It assumes that, all features are independent. Naïve Bayes classifier has the following three advantages. First, in some probability models, Naïve Bayes classifier can be effectively trained in supervised learning environment. Second, the amount of training

data used to estimate the necessary parameters for classification need not necessarily be large. Third, despite a simple design, Naïve Bayes classifier operates well in various complicated situations (Yoo and Yang, 2015). Given a class **C** and Variable vector **X**, we use the training data to determine the probabilities P(**X**/**C**) and P(**C**) for all values of ***X*** and ***C***. New **X** examples can then be classified using these estimated probability distributions with Bayes Rule. This type of classifier is called a generative classifier, because we can view the distribution P(**X**/**C**) as describing how to generate random instances of **X** conditioned on the target class **C**.

Logistic Regression is a model that uses training data to directly estimate the probability of an instance of some set of features or a variable vector **X** belonging to a class **C**, P(**C**/**X**), in contrast to Naive Bayes. In this sense, Logistic Regression is often referred to as a discriminative classifier because we can view the distribution P(C/X) as directly estimating the probabilistic value of the target C for any given instance of **X**.

The success of classifiers depends on the nature of the relationship between feature selection and Classification Accuracy. Researchers such as Kwon and Sim (2013), have tried to understand the nature of this relationship using some selected classifier models (algorithm). However, their explanation is too general and therefore not very informative. Ultimately, we would like to understand the performances of Naïve Bayesian Classifier and Logistic Regression when used as binary classifiers on the same domain area (In our case: for health problem diagnosis. Presence or absence of an ailment) under certain interactions of dataset sizes and variable selection methods.

1.2 Statement of the Problem

As the need to analyze big data sets grows exponentially, the role that classification algorithms play in data mining techniques also increases. As Kwon and Mun Sim (2013), noted that it is still a complex issue to determine which algorithm is strong or weak in relation to which data set, where in they experimentally examined how dataset characteristics affect a models performance. The key problem when dealing with classification problem is not whether a model is superior to others, but *under which conditions a particular method can significantly outperform others on a given application problem*. Naïve Bayesian Classifier and Logistics Regression have been reported to do well with a variety of datasets. This research proposes to find an optimal choice between these two classifiers.

1.3 Aim and Objectives of the Study

The aim of this research work is to study the classification performance of Naïve Bayesian Classifier and Logistic Regression under different feature (variable) selection criteria and

dataset size conditions on a domain area. The aim shall be achieved through the following objectives by:

i. building a Naïve Bayesian classifier model for each of the pre-determined conditions;
ii. building a Logistic Regression classifier model for each of the pre-determined conditions;
iii. testing the models in objectives (i) and (ii) on some datasets in order to measure their respective classification accuracies;
iv. performing a test of independence in the interaction of feature selection criteria, dataset size and choice of classifier model (algorithm).

1.4 Significance of the Study

This study helps to understand the optimal performances of Logistics Regression and Naïve Bayesian Classifier which are both linear statistical classification models that are fast becoming the choice of many researchers. Particularly, the results help to make optimal decisions on choice of model, and consequently improve the performances of classification algorithms.

1.5 Motivation

Our motivation stems from the resolution of Kwon and Sim (2013) that noted the complexity of having to determine which classifier model is strong or weak in relation to data sets from a specified domain of study, and concludes that the conditions under which a particular method significantly outperform the others on a giving application problem is the key to dealing with classification. In view of that, we were motivated to study the effect of feature selection and dataset size on the accuracy NBC and LR as limited to data sets from a medical domain.

1.6 Scope and Limitation of the Study

This research is limited to empirical data on medical records collected for breast cancer disease and heart disease which are suitable for classification. Data from non-medical domain areas are not considered.

1.7 Definition of Terms

a) Classification

Classification is a statistical method used to build predicative models to separate and classify new data points.

b) Classifier

The predicative model built to separate and classify new data points is known as a classifier.

c) Naïve Bayes

Naive Bayes is a classifier which is based on applying Bayes' theorem with the basic assumption of independence between every pair of features.

d) Logistics Regression (LR)

Logistic regression or logit model is a regression model whose dependent variable is categorical and takes only two values, such as pass or fail, win or lose, alive or dead, presence of disease or absence of disease. Multinomial Logistics regression has cases (in the form of dependent variables) with more than two categories.

e) Training set

A training set is a set of data used to discover potentially predictive relationships.

f) Testing set

A testing set is a set of data used to assess the strength and utility of a predictive relationship.

g) Confusion matrix

A confusion matrix is a table that is often used in the description of the performance of a model (classification models) on a set of data meant for the purpose of testing (usually called the test data set) for which the true values are known.

	Predicted (no disease)	Predicted (disease)
Actual (no disease)	TN	FP
Actual (disease)	FN	TP

Table 1. 1: Sample of a Confusion Matrix

h) Accuracy

Accuracy is the percentage of correct predictions made. In other words, the accuracy is the proportion of true results (that is, both true positives and true negatives) among the total number of cases examined.

i) Precision

$$(TP)/(TP+FP)$$

Precision gives information on the proportion of patients diagnosed as having a disease by the classifiers had it in the real case. It can be defined as the proportion of True Positive in the set of subjects diagnosed as positive to the condition been tested upon.

j) Sensitivity (Recall)

$$((TP)/(TP+FN))$$

Sensitivity computes the proportion of patients that actually had the disease who were diagnosed as having it. One should be careful not to mix up the meaning of sensitivity for precision. Sensitivity gives the proportion of True positive in the set of subjects having the condition in reality, like the proportion of patients who had actually had breast cancer and were diagnosed having it by the classifier model.

k) Specificity (True negative rate)

Specificity (SP) is calculated by dividing the number of correct negative predictionsby the total number of negative subjects (patients). Specificity may appear in other texts as true negative rate (TNR) or simply, Specificity, both terms mean the same thing. Specificity of 1.0 is considered the best, whereas 0.0 is considered the worst.

Specificity is calculated by dividing the number of correct negative predictions (TN) by the total number of negatives (N).

Specificity, $SP = TN/(TN + FP)$

$$SP = TN / N$$

Remember, $TNR = SR$

l) Principal Component Analysis

Principle Component Analysis (PCA) is a statistical technique used to examine the relation that exists among a set of variables in order to identify the structural pattern of those variables. PCA, also called factor analysis, is a non-parametric analysis and answers uniquely and independently of hypothesis about the data distribution.

m) Multicollinearity

In statistics, multicollinearity (sometimes called collinearity) is a phenomenon whereby two or more independent predictor variables in a multivariate regression model have a high correlation, which is an indication that one may be linearly predicted from the others with an acceptable degree of accuracy.

CHAPTER TWO: LITERATURE REVIEW

To provide an appropriate framework for this research, a review of some past related works in this field of study is undertaken to provide sufficient theoretical background to the study. These include review of some relevant empirical works, and an overview of some classification models.

Naïve Bayesian Classifier (NB) and logistics Regression (LR) are classification methods which uses historical data to construct classification models. Medical institutions can use the idea in evaluation of patients with an ailment and others. NB and LR are among the most popular approaches for representing classifiers. Researchers from Statistics and Data Mining have tried resolving problems with classification by comparing classification models.

Press and Wilson (1978) carried out two empirical studies of non-normal classification problems. They compared Logistics regression classical linear discriminant analysis methods, and found that *logistic regression with Maximum likelihood estimation (MLE) outperformed classical linear discriminant analysis in both cases*, but not by a large amount. They, thus, agreed with the conclusion of Halperin, Blackwelder, and Verter (1971) that the use of the maximum likelihood method is preferable in situations where the normality assumptions are not satisfied, especially in situations where many of the independent variables are not quantitative.

Steven, *et al* (2001) examines the use and reporting of Logistic regression in the medical literature by comprehensively assessing its use in a selected area of medical study. Through the Medline database, followed by bibliography query searches, the resulting list of articles was processed by hand to retain only articles that: (1) only contained results of original studies (that is, removing reviews), (2) measured patients' interest in the testing of genetics for cancer susceptibility, and (3) contained explicit mention of LR modelling. They identified 15 peer-reviewed English-language articles with original data, employing LR, which were

published between 1985 and 1999, pertaining to patient interest in genetic testing for cancer susceptibility. The articles were examined for each of 10 criteria for proper use and reporting of LR models. They found Substantial shortcomings in both use of LR and reporting of results. For many studies, they found that the ratio of the number of outcome events to predictor variables (that is, events per variable) was small enough to consider calling into question the accuracy of the regression model. Additionally, they found there are no studies that reported validation analysis, regression diagnostics, or even the goodness-of-fit measures. They concluded that Logistic regression is a type of multivariable analysis used that is being frequency used in health sciences researches because of its ability to model dichotomous (binary) outcomes, and that the proper use of such a powerful and yet sophisticated modelling technique requires a great care both in the specification of the form of the model and in the calculation and interpretation of the coefficients of the model, and thus recommended that authors, reviewers, and editors pay greater attention to guidelines established concerning the use and reporting of LR models.

As Naïve Bayesian classifier have shown significant performance in classification tasks, researchers began to find ways to make such a simple classifier even more efficient. Cerquides and DeMantaras (2003) introduced a classifier taking as its basis the *Tree Augmented Naïve Bayesian (TAN)* model, a modification of Naïve Bayesian Classifier, which takes into account uncertainty in model selection. They introduced decomposable distributions over TANs and show that they allow the expression resulting from the Bayesian model averaging of TAN models to be integrated into closed form. With the result, they constructed a classifier with a shorter learning time and a longer classification time than *TAN*. Empirical results show that *the classifier is, in most of the cases, more accurate than TAN and approximates the class probabilities better*.

Rish (2003) conducted a study to understand the data characteristics which affect the performance of naive Bayes. The approach used Monte Carlo simulations that allowed a

systematic study of classification accuracy for several classes of randomly generated problems. The work demonstrated that naive Bayes works well for certain nearly functional feature dependencies, thus reaching its best performance in two opposite cases: completely independent features (as expected) and functionally dependent features (which is surprising). Another finding was that the accuracy of naïve Bayes is not directly correlated with the degree of feature dependencies measured as the class conditional mutual information between the features. He however noted that a *deeper understanding of data characteristics that affect the performance of Naive Bayes* is still required.

The art of machine learning starts with the design of appropriate data representations. Guyon and Elisseeff (2003) highlighted the objective of variable selection in three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data. They noted that one of the potential benefits of variable/feature selection could be to defy *the curse of dimensionality* to improve prediction performance. Their paper focused mainly on *constructing and selecting variables that are useful to build a good predictor*. Steps that may be taken for variable (feature) selection were given. "Sophisticated variable selection methods improve predictor performance compared to simpler ones like correlation methods, but the improvements are not always significant."

Despite its simplicity, naive Bayes classifier surprises machine learning researchers by exhibiting good performance on a variety of learning problems. Encouraged by these results, researchers have looked to overcome naive Bayes' primary weakness, which is the assumption of attribute independence, and improve the performance of the algorithm. For instance, Frank *et al* (2004) presented a locally weighted version of naive Bayes that relaxes the independence assumption by learning local models at prediction time. They claimed, by experiment, that "locally weighted naive Bayes rarely degrades accuracy compared to

standard naïve Bayes and, in many cases, improves accuracy dramatically". They suggested exploring general locally-weighted classification as one direction for future work.

Zhang (2004) proposed a novel explanation on the superb classification performance of naive Bayes. The work explains that "*no matter how strong the dependences among attributes are, naive Bayes can still be optimal if the dependences distribute evenly in classes, or if the dependences cancel each other out*". The explanation was restricted to two attributes, X_1 andX_2 contained in$X$, and the assumption that the class density is a Multivariate Gaussian in both the positive and negative classes.

Kotsiantis (2007) described various supervised machine learning classification techniques. He observed that the classifier's evaluation is most often based on prediction accuracy (the percentage of correct prediction divided by the total number of predictions). He gave three techniques which are used to calculate a classifier's accuracy: one is to split the training set by using two-thirds for training and the other third for estimating performance; another technique, known as cross-validation, the training set is divided into mutually exclusive and equal-sized subsets and for each subset the classifier is trained on the union of all the other subsets. A common method for comparing supervised Machine Learning algorithms is to perform statistical comparisons of the accuracies of trained classifiers on specific datasets. However, in practice there is always certain sensitivity to the partitioning used. To measure "replicability", he said we need to repeat the same test several times on the same data with different random partitioning. Although, the comparison does not capture logistics regression or the models' sensitivity to feature selection criteria. He noted that, "The key question when dealing with ML classification is not whether a learning algorithm is superior to others, but under which conditions a particular method can significantly outperform others on a given application problem".

Janecek *et al* (2008) researched on the relationship between feature selection and classification accuracy. Dimensionality reduction and feature selection are two techniques for

reducing the space of a feature set. By experimenting on several classifiers, they investigated the relationship between several feature set reduction techniques and the resulting classification accuracy for two "very" different application areas: e-mail filtering and drug discovery problems. Their results showed that random forest (RandF) classification technique using principal component analysis (PCA) had the highest accuracy of about 99.77%. It should however be noted that the researchers used non-statistical classifier algorithms, neither naïve Bayesian network nor logistics regression was one of the algorithms studied, and they admittedly stated that "when looking specifically at the two data sets investigated in this paper, we note that the classification accuracy achieved with different feature reduction strategies is highly sensitive to the type of data".

Kwon and Mun Sim (2013), evaluated scenarios that examine which data set characteristics most affect the classification models' performances. They noted that it was a complex issue to determine. They experimentally examined how data set characteristics affect a model's performance, both in terms of accuracy and in elapsed time. The study was carried out on all benchmark data sets in UCI database that are fit to run the classification algorithm. The dataset characteristics were: class type,sample size, unlabelled class, missing values, continuous features, functional dependency of features, dimensionality (number of features), data imbalance, and domain area. The paper was focused on classification algorithms, and they examined which characteristics of a data set influenced their performances. They observed that the classification algorithms showed different performance about different kind of data structures, content and context. They suggested Context-aware selection of classification algorithms as an important field for further research.

Bhowmik (2015) compared the Naïve Bayes (NB) with Logistic regression (LR). He performed a set of experiments for both of the classifiers under different learning circumstances and their performances are compared and Statistical tests to measure the significance of comparison were conducted. From the experiments, the researcher observed

that LR learning with gradient ascent technique performs better than the general NB classifier. However, under Gaussian Naive Bayes assumption, both classifiers, NB and LR, performed similar. The aim of the work was to bring forth a complete study describing from the basic theoretical concepts of learning techniques to their implementation details. The aim was achieved by building several classifiers under different learning circumstances and measuring their performances. The researcher also noted that a question can arise about "which classifier would be appropriate for a real application?", and that "this is probably difficult to suggest as it completely depends on the data and the objective of the application as well".

The review of these works indicates that the researchers have only compared various classification models in terms of the superiority of their learning algorithms on general datasets without discrimination, and the theoretical backgrounds with which they were built. Also, they all seem to point to the fact that for any real application, context must be put to check. Our work therefore addresses the application of our choice models on the context of dataset domain, size and feature selection method as practicable for medical diagnosis.

CHAPTER THREE: RESEARCH METHODOLOGY AND MATERIALS

3.1 Introduction

In this chapter, we introduce the materials and methods by which this research is carried out. The source of data, the theory behind the models, and the statistical software packages that were used. We further give a framework for evaluating models performance, and conclude by stating the method of model comparison with regards to the size of the population under study. The blocks of codes written in R Package are documented in the appendix section of this report for the sake of clarity and organization, and hence, subsections of this chapter provide references to the codes as related to the two appendices.

3.2 Source of Data

The data used were sourced from the University of California Irvine (UCI) repository. As at the time this research was carried out, UCI was maintaining 360 data sets as a service to machine learning community. We have used medical datasets categorised for classification – the heart disease dataset and the breast cancer data, both of which are standard, high quality, real-world, and well understood machine learning data sets.

3.3 Method of data analysis

The data were analysed using one of the generative linear classifier called Logistic regression, and one of discriminative linear classifier called Naïve Bayes Classifier. R Software is used, with each model requiring its own R packages combination.

3.4 Principal component analysis (PCA)

PCA is a useful statistical technique that has found application in many fields(face recognition, etc.). PCA is a method to study the structure of a data set, with emphasis on determining the patterns of covariances among variables. Thus, PCA aims to understand the pattern of the variance-covariance matrix. Practically, PCA is a method used for the

identification of variable or sets of variables that have high correlations with each other. The results can be used for multiple purposes as follows:

- Construction of a new set of variables that are linear combinations of the original variables set and that contain absolutely the same information as the original variables, even though orthogonal to each other.
- Identification of patterns of feature pair correlations in a data set and use the results to address the multicolinearity problem in multiple linear regression.
- Identification of variables or factors, which underlie the original variables, that caused the variation in the data.
- Finding out the effective number of dimensions over which the data set exhibits variation, with the purpose of reducing the number of dimensions of the problem.
- Creating a few orthogonal features or variables that have most of the information in the data which simplifies the identification of groupings in the observations.

The application of PCA is made to a single group of variables; there is no distinction between predictor and response variables. In multiple linear regression (MLR), such as Logistics Regression involving multiple variable, PCA is applied only to the set of X variables, to study correlation that exists between sets of pair of variables.

Model and concept

No model is to be tested in regards to PCA, although it is assumed that the variables are linearly related. This is best understood by the thought that the same set of data is to be looked at from different perspective. The perspective is changed by moving the origin of the coordinate system to the centroid of the data and then rotating the axes.

More specifically, given a set of q variables (X_1, ..., Xq), PCA calculates a set of q linear combinations of the variables (PC_1, ..., PCq) such that:

- The total variation in the original variable is same as that of the new set of variables or principal components (PCs).

- The most variance possible is recorded in the first PC, e.g. as much variance as can be captured in a single axis.
- Second PC will be orthogonal to the first one (they have zero (0) correlation), and will contain as much of the remaining variance as possible.
- And the third PC is orthogonal to all previous PCs, and also contains the most variance possible. And others as the ones before.

The procedure is achieved by computing a matrix of coefficients with columns axis called eigenvectors of the variance-covariance, or eigenvectors of the correlation matrix of the data set. The following is some of the basic consequences of the procedure:

- The calculation of PC scores (that is, the location of individual observations in the newly generated axis formed by the PC's) require all original variables.
- The variances of the PC's when summed up equals the sum of the variances of the original variables, when PCA is based on the variance-covariance matrix.
- There are q eigenvalues (q =data set size), each one having association with one eigenvector and a Principal Component. These eigenvalues represents the variances of the data in each PC. Thus, eigenvalues summed up based on the variance-covariance matrix equals the sum of variances of the original variables.

PCA based on the correlation matrix is equivalent to using PCA based on the variance-covariance of the standardized variables. Because standardized variables have variance=1, the sum of eigenvalues is p, the number of variables.

Assumptions of PCA

- Linearity: this assumes the data set is a linear combinations of the variables.
- Importance of covariance and mean: There is no guarantee that the directions of maximum variance provides good features for discrimination.

- Also, that large variances have important dynamics: here, it assumes that components with larger variance correspond to interesting dynamics and lower ones correspond to noise.

Implementation of PCA in R

The default prcomp() function in R performs a principal components analysis on the given data matrix and returns the results as an object of class prcomp.The calculation of PCA is done by a singular value decomposition of the data matrix, not by using eigenvector on the covariance matrix. This method is generally preferred for numerical accuracy, (R documentation website). All PCA done in this research work is based on this implementation as seen documented in the appendix of this report.

3.5 Logistics Regression

Logistic regression is a statistical method used for analyzing a dataset having one or more independent variables that determine an outcome. The outcome is measured with a dichotomous (binary) variable.

In logistic regression, the dependent variable is always binary or dichotomous, that is. it only contains data coded as 1 (Success, presence, etc.) or 0 (Failure, absence, etc.).

Logistic regression aims at finding the best fitting (which is also biologically reasonable) model to describe the relationship between the binary characteristic of interest (dependent variable = response variable) and a set of independent (predictor or explanatory) variables. Logistic regression generates the coefficients (with its standard errors and significance levels) of a formula to predict a logit transformation of the presence probability of the characteristic of interest: the interests in our work is healthy and unhealthy, or ‘absence of disease’ and ‘presence of disease’. The following section gives the theoretical build of the binary logit model.

3.5.1 Binary Logit model from the Logistic Function

The common definition of Logistic Function is as follows:

$$P(x) = \frac{1}{1+\exp(-x)} \tag{3.1}$$

where $x \epsilon R$ is the variable of the function and $P(x) \epsilon [0,1]$. One importance of Equation 3.1 is that:

$$P(-x) = \frac{1}{1+\exp(x)} \tag{3.2}$$

$$P(-x) = \frac{1}{1+(1/\exp(-x))}$$

$$P(-x) = \frac{\exp(-x)}{1+\exp(-x)}$$

$$P(-x) = 1 - \frac{1}{1+\exp(-x)}$$

$$P(-x) = 1 - P(x) \tag{3.3}$$

The form of equation 3.1 is widely used as the form of Logistic Regression

$$P(y=1 \mid \overline{\beta}, x) = \frac{\exp(\overline{\beta}^T x)}{1+\exp(\overline{\beta}^T x)} \tag{3.4}$$

$$P(y=0 \mid \overline{\beta}, x) = \frac{1}{1+\exp(\overline{\beta}^T x)} \tag{3.5}$$

Where x is a feature vector and β is a coefficient vector. By using equation (3.4), we also have:

$$P(y=1 \mid \overline{\beta}, x) = 1 - P(y=0 \mid \overline{\beta}, x) \tag{3.6}$$

This formation of Logistic Regression is used where $y \epsilon \{0,1\}$ and the functional form of the probability to generate different labels is different. Another formalism unified the two forms into one single equation by integrating the label and the prediction together:

$$P(g=\pm 1 \mid \overline{\beta}, x) = \frac{1}{1+\exp(-g\overline{\beta}^T x)} \tag{3.7}$$

Where $g \epsilon \{\pm 1\}$ is the label for data item x. It is also easily to verify that $P(g=1 \mid \overline{\beta}, x) = 1 - P(g=-1 \mid \overline{\beta}, x)$

The equivalence of Two Forms of Logistic Regression

At first glance, the form (3.4) and the form (3.6) looks seem very different. However, the equivalence for these two forms can easily be established. Starting from the form (3.4), we can have:

$$P\left(y=1|\,\bar{\beta},x\right)=\frac{\exp\left(\bar{\beta}^{T}x\right)}{1+\exp\left(\bar{\beta}^{T}x\right)}$$

$$P\left(y=1|\bar{\beta},x\right)=\frac{1}{\left(1/\exp(\bar{\beta}^{T}x)\right)+1}$$

$$P\left(y=1|\,\bar{\beta},x\right)=\frac{1}{\exp\left(-\bar{\beta}^{T}x\right)+1}$$

$$P\left(y=1|\,\bar{\beta},x\right)=P\left(g=-1|\,\bar{\beta},x\right) \tag{3.8}$$

Another way to establish the equivalence is from the classification rule. For the form (3.4), we have the following classification rule:

$$\frac{P\left(y=1|\,\bar{\beta},x\right)}{P\left(y=0|\,\bar{\beta},x\right)}>1\rightarrow \text{y}=1 \tag{3.9}$$

$$\frac{\left(\exp\left(\bar{\beta}^{T}x\right)/\left(1+\exp\left(\bar{\beta}^{T}x\right)\right)\right)}{\left(1/\left(1+\exp\left(\bar{\beta}^{T}x\right)\right)\right)}>1\quad\rightarrow \text{y=1}$$

$$\exp\left(\bar{\beta}^{T}x\right)>1 \tag{3.10}$$

$$\left(\bar{\beta}^{T}x\right)>0 \tag{3.11}$$

Applying the model for Classification

We consider two classes in this research, the presence of disease (label 1) and the absence of disease (label 0). The class decision is based on the result from 3.9 in the previous section. That is

$$\frac{P\left(y=1|\,\bar{\beta},x\right)}{P\left(y=0|\,\bar{\beta},x\right)}>1\rightarrow \text{y}=1$$

The $P\left(y=1|\bar{\beta},x\right)$ is the probability of presence of the characteristic of interest (disease), while $P(y=0|\beta,x)$ is the probability of absence of the characteristic of interest (disease). The quotient can be called an odd ratio.

Implementation of LR in R

R makes it very easy to fit a logistic regression model. The function to be called is glm(). This is used to fit generalized linear models, which is specified by giving a symbolic description of the linear predictor and a description of the error distribution. The attribute 'family' of the glm() gives the link function. We have used binomial link function as it gives the glm a logistic regression definition.

3.6 Naïve Bayesian Classifier

Naïve Bayes classifier is a generative classifier model with a simplifying assumption that all input attributes are conditionally independent of each other given the class, the naïve Bayes classifier is formulated from the Bayes rule of probability.

3.6.1 Formulation of the model

Consider the following conditional probability:

$$P(\bar{X}|C)=\frac{P(\bar{X},C)}{P(C)} \tag{3.12}$$

Bayes theorem:

$$P(C|\bar{X})=\frac{P(\bar{X}|C)P(C)}{P(\bar{X})} \tag{3.13}$$

When faced with the problem of classification, we consider each attribute and class label as random variables. Given a record with attributes $(x_1,x_2,\ldots,x_n)$, the goal is to predict class C. Specifically, we want to find the value of C that maximizes $P(C|\,x_1,x_2,\ldots,x_n)$.

Can we estimate $P(C|x_1,x_2,\ldots,x_n)$ *directly from data?*

Approach:

Compute the posterior probability $P(C|x_1, x_2, \dots, x_n)$ for all values of C using the Bayes' theorem:

$$P(C \mid \mathrm{x}_1, x_2, \ldots, x_n) = \frac{P(\mathrm{x}_1, x_2, \ldots, x_n \mid \mathrm{C})}{P(\mathrm{x}_1, x_2, \ldots, x_n)} \tag{3.14}$$

Choose value of C that maximizes $P(C|x_1, x_2, \dots, x_n)$ which is equivalent to choosing the value of c that maximizes $P(x_1, x_2, \dots, x_n|C)P(C)$.

How to estimate $P(x_1, x_2, \dots, x_n|C)$*?*

We require all possible combinations of $x_1, x_2, \dots, x_n$ which generates the class C, however not all combinations are present.

Hence, additional assumption on the distribution is required – class independence.

Assuming independence among attributes Xi when a class is given:

$$P(\mathrm{x}_1, x_2, \ldots, x_n \mid \mathrm{C}) = \frac{P(\mathrm{x}_1 \mid \mathrm{C})\mathrm{P}(x_2 \mid \mathrm{C}), \ldots, P(x_n \mid \mathrm{C})P(C)}{P(\mathrm{x}_1, x_2, \ldots, x_n)} \tag{3.15}$$

Which becomes

$$P(\mathrm{x}_1, x_2, \ldots, x_n \mid \mathrm{C}) = \frac{P(C)\prod_{i=1}^{n} P(x_i \mid \mathrm{C})}{P(\mathrm{x}_1, x_2, \ldots, x_n)} \tag{3.16}$$

3.6.2 Learning the model:

For every class C:

Estimate the Prior, $P(C)$,

For every attribute x, for every attribute value v of x, estimate $P(x = v|C)$

3.6.3 Applying the model:

• Given an example $(v_1, v_2, \dots, v)$

• Pick the class C that maximizes

$P(C)\prod_{i=1}^{n} P(x_i \mid \mathrm{C})$

3.7 Classification of New Data

To test our models performance, we have to classify instances, the testing set (which still has the original number of attributes of the training set). We reserve 25% of the original data for this purpose, which would be treated as unlabelled instances. Matching the new class predictions with the actual class they belong gives an insight for measuring our models performance.

3.8 Model's performance evaluation

The evaluation of the performance of the binary classification models is based on some statistical criteria or performance metrics. The performance metrics employed in this work include accuracy, sensitivity and specificity. These counts are tabulated in a table called confusion matrix.

3.8.1 Confusion Matrix

In the confusion matrix for a binary classification problem each entry denotes the number of records from either classifying an observation correctly to its proper class, C0 to C0, C1to C1 or misclassifying an observation to another class that is C1 to C0, C0 to C1.

Creating a confusion matrix is a common technique to assess the performance of a classification model. There is just a fairly simple straightforward idea – taking statistics of actual class and predicted class membership distribution. Comparing the predicted classes with actuals results in 4 possible scenarios (for our binary models) which can be represented by a "2 by 2" matrix. We can then use the matrix to compute various performance measure such as accuracy, specificity, Precision, etc.

		Actual value		
		p	*N*	Total
Prediction	*p'*	True Positive	False Positive	P'

	n'	False Negative	True Negative	N'
	Total	P	N	

Table 3. 1: 2x2 Confusion Matrix for Classification Result

The performance of our models is evaluated by computing the percentages of Sensitivity (SE), Specificity (SP) and Accuracy (AC), the respective definitions are as follows:

In terms of medical diagnosis, a classifier's sensitivity (SE) measures its ability to correctly identify subjects with the disease (true positive rate, e.g., the percentage of breast cancer patients who are correctly identified as having breast cancer)

$$SE = \frac{TP}{(TP + FN)} \times 100$$

Its specificity (SP) is the ability to correctly identify those without the disease (true negative rate, e.g., the percentage of healthy people who are correctly identified as not having the condition).

$$SP = \frac{TN}{(TN + FP)} \times 100$$

$$AC = \frac{(TP + TN)}{(TN + TP + FN + FN)} \times 100$$

Where: TP refers to the number of true positives,

TN refers to the number of true negatives,

FN refers to the number of false negatives, and

FP is the number of false positives

3.8.2 Comparing multiple models

We have more than one candidate model and we would simultaneously evaluate and compare the performance of these models. For each of these candidate models, we would like to

simultaneously evaluate the impact to confusion matrix. Multiple bar charts were employed in this research for the sake of visual representation.

CHAPTER FOUR: ANALYSIS AND DISCUSSION

4.1 Introduction

This chapter contains the analyses and results obtained using the prescribed methods in the previous chapter. We have used the Breast Cancer and Heart disease data sets obtained from the University of California Irvine (UCI) open source repository, both of which are standard data sets with multiple independent variables and a dependent categorical response variable. A smaller set is selected from each data sets in order to examine the effect of dataset on the classification models. The resulting datasets are partitioned into training and testing sets on a 3:1 ratio, while maintaining an even distribution of class observations on each sets – to remove all forms of bias for class prediction. The Naïve Bayes (NB) and Binary Logistics Regression (LR) Classifiers were built for individual set before and after the application of Principal Component Analysis (PCA), used for dimension reduction. The criteria used for the evaluation of model's performance are the Accuracy, Sensitivity and Specificity.

4.2 Model Building and evaluation for the Breast Cancer Dataset

This dataset has 10 recorded features described as follows:

Variables (features)

V1 = Sample code number (Not a feature)

V2 = Clump Thickness

V3 = Uniformity of Cell Size

V4 = Uniformity of Cell Shape

V5 = Marginal Adhesion

V6 = Single Epithelial Cell Size

V7 = Bare Nuclei

V8 = Bland Chromatin

V9 = Normal Nucleoli

V10 = Mitoses

V11 = Class (2 for benign, 4 for malignant)

4.2.1 Building and evaluating NB on the larger dataset, with no PCA

We built a Naïve Bayes Classifier for the larger Breast Cancer dataset and evaluated its performance as illustrated with R codes in Apendix I, Code block label "a". The confusion matrix is as follows:

	Actual (no disease, 2)	Actual (4, disease)
Predicted (no disease, 2)	TP = 57	FP = 0
Predicted (disease, 4)	FN = 4	TN = 61

Table 4. 1: Confusion Matrix for NB on the larger dataset, with no PCA

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

Accuracy = 96.72131%

$$Sensitivity = TP / (TP + FN) = TP/P = 61/61 = 1.0$$

$$Specificity = TN / (TN + FP) = TN/N = 57/61 = 0.93$$

Building and evaluating NB on the smaller dataset, with no PCA

We built a Naïve Bayes Classifier for the smaller Breast Cancer dataset and evaluated its performance as illustrated with illustrated with R codes in Apendix I, Code block label "b". The confusion matrix is as follows:

	Actual (no disease, 2)	Actual (4, disease)

Predicted (no disease, 2)	TP = 17	FP = 0
Predicted (disease, 4)	FN = 1	TN = 18

Table 4. 2: Confusion Matrix for NB on the smaller dataset, with no PCA

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$Accuracy = 97.22222$$

$$Sensitivity = TP/(TP + FN) = TP/P = 18/18 = 1.0$$

$$Specificity = TN / (TN + FP) = TN/N = 17/21 = 0.81$$

3.8.3 Building and evaluating LR on the larger dataset, with no PCA

We built a Logistics Regression Classifier for the larger Breast Cancer dataset and evaluated its performance as illustrated with the R codes in Apendix I, Code block label "c". The confusion matrix is as follows:

	Actual (no disease, 2)	Actual (4, disease)
Predicted (no disease, 2)	TP = 52	FP = 1
Predicted (disease, 4)	FN = 4	TN = 60

Table 4. 3: Confusion Matrix for LR on the larger dataset, with no PCA

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$Accuracy = 95.90164$$

$$Sensitivity = TP/(TP + FN) = TP/P = 60/61 = 0.98$$

$$Specificity = TN / (TN + FP) = TN/N = 57/61 = 0.93$$

3.8.4 Building and evaluating LR on the smaller dataset, with no PCA

We built a Logistics Regression Classifier for the smaller Breast Cancer dataset and evaluated its performance as illustrated with R codes in Apendix I, Code block label "d". The confusion matrix is as follows:

	Actual (no disease, 2)	Actual (4, disease)
Predicted (no disease, 2)	TP = 17	FP = 1
Predicted (disease, 4)	FN = 1	TN = 17

Table 4. 4: Confusion Matrix for LR on the smaller dataset, with no PCA

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$Accuracy = 94.44444$$

$$Sensitivity = TP/(TP + FN) = TP/P = 17/18 = 0.94$$

$$Specificity = TN/(TN + FP) = TN/N = 17/18 = 0.94$$

DIMENSION REDUCTION WITH PCA

In order to reduce dimension, select fewer features, and examine the effect this reduction (by PCA) have on the classifiers performance, we apply PCA on the Breast Cancer dataset as shown in Apendix I, block code label "e".

The screeplot is show in the figure that follows:

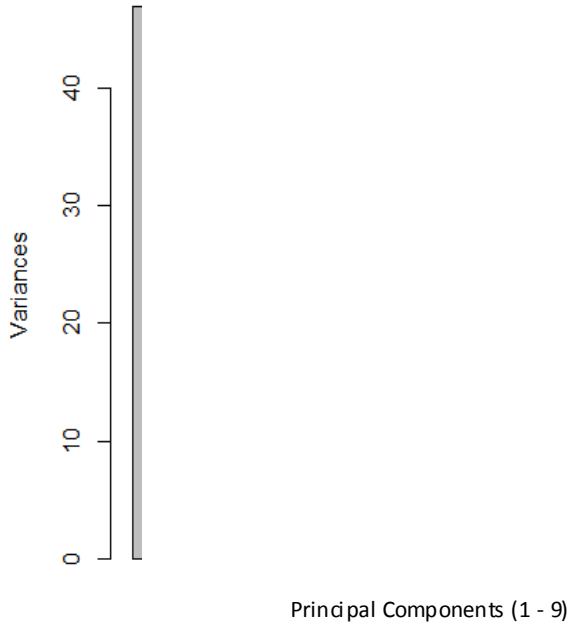


Figure 4. 1: A scree plot showing the levels at which each components explains the variability.

We looked at the proportion of variances explained by the PC components, we realized that the first 8 components explain 98.5% of the overall variability. The next step we performed was examining the rotations of these selected PC objects individually, in order to select the features that yields the PC's greatest magnitude. This feature (or variable) will be part of the predictors.

Thus, the predictor variables selected are V3, V5, V6, V7, V8, V9 and V10.

3.8.5 Building and evaluating NB on the larger dataset, with PCA

We built a Naïve Bayes Classifier for the larger Breast Cancer dataset, with predictor features as selected by the PCA, and evaluated its performance as illustrated with R codes in Apendix I, Code block label "f". The confusion matrix is as follows:

	Actual (no disease, 2)	Actual (4, disease)

Predicted (no disease, 2)	TP = 57	FP = 0
Predicted (disease, 4)	FN = 4	TN = 61

Table 4. 5: Confusion Matrix for NB on the larger dataset, with PCA

$Accuracy = (TP + TN)/(TP + TN + FP + \text{FN})$

Accuracy = 96.72131

$Sensitivity = TP/(TP + FN) = TP/P = 61/61 = 1.0$

$Specificity = TN / (TN + FP) = TN/N = 57/61 = 0.93$

3.8.6 Building and evaluating NB on the smaller dataset, with PCA

We built a Naïve Bayes Classifier for the smaller Breast Cancer dataset, with predictor features as selected by the PCA, and evaluated its performance as illustrated with R codes in Apendix I, Code block label "g". The confusion matrix is as follows:

	Actual (no disease, 2)	Actual (4, disease)
Predicted (no disease, 2)	TP = 17	FP = 0
Predicted (disease, 4)	FN = 1	TN = 18

Table 4. 6: Confusion Matrix for NB on the smaller dataset, with PCA

$Accuracy = (TP + TN)/(TP + TN + FP + FN)$

$Accuracy = 97.22222$

$Sensitivity = TP/(TP + FN) = TP/P = 18/18 = 1.0$

$Specificity = TN / (TN + FP) = TN/N = 17/18 = 0.94$

3.8.7 Building and evaluating LR on the larger dataset, with PCA

We built a Logistics Regression Classifier for the larger Breast Cancer dataset, with predictor features as selected by the PCA, and evaluated its performance as with R codes in Apendix I, Code block label "d". The confusion matrix is as follows:

	Actual (no disease, 2)	Actual (4, disease)
Predicted (no disease, 2)	TP = 58	FP = 1
Predicted (disease, 4)	FN = 3	TN = 60

Table 4. 7: Confusion Matrix for LR on the larger dataset, with PCA

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$Accuracy = 96.72131$$

$$Sensitivity = TP/(TP + FN) = TP/P = 60/61 = 0.98$$

$$Specificity = TN/(TN + FP) = TN/N = 58/61 = 0.95$$

3.8.8 Building and evaluating LR on the smaller dataset, with PCA

We built a Logistics Regression Classifier for the smaller Breast Cancer dataset, with predictor features as selected by the PCA, and evaluated its performance as illustrated with R codes in Apendix I, Code block label "i". The confusion matrix is as follows:

	Actual (no disease, 2)	Actual (4, disease)
Predicted (no disease, 2)	TP = 17	FP = 2
Predicted (disease, 4)	FN = 1	TN = 16

Table 4. 8: Confusion Matrix for LR on the smaller dataset, with PCA

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$Accuracy = 91.66667$$

$$Sensitivity = TP/(TP + FN) = TP/P = 16/18 = 0.89$$

$$Specificity = TN / (TN + FP) = TN/N = 17/18 = 0.94$$

3.9 Model Building and evaluation for the Heart Disease Dataset

This dataset has 14 recorded features described as follows:

Variables (Features)

V1 = age: age in years

V2 = sex: sex (1 = male; 0 = female)

V3 = cp: chest pain type

-- Value 1: typical angina

-- Value 2: atypical angina

-- Value 3: non-anginal pain

-- Value 4: asymptomatic

V4 = trestbps: resting blood pressure (in mm Hg on admission to the hospital)

V5 = chol: serum cholestoral in mg/dl

V6 = fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)

V7 = restecg: resting electrocardiographic results

-- Value 0: normal

-- Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)

-- Value 2: showing probable or definite left ventricular hypertrophy

by Estes' criteria

V8 = thalach: maximum heart rate achieved

V9 = exang: exercise induced angina (1 = yes; 0 = no)

V10 = oldpeak = ST depression induced by exercise relative to rest

V11 = slope: the slope of the peak exercise ST segment

-- Value 1: upsloping

-- Value 2: flat

-- Value 3: downsloping

V12 = ca: number of major vessels (0-3) colored by flourosopy

V13 = thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

V14 = Class (the predicted attribute): diagnosis of heart disease (angiographic disease status)

-- Value 0: < 50% diameter narrowing

-- Value 1: > 50% diameter narrowing

3.9.1 Building and evaluating NB on the larger dataset, with no PCA

We built a Naïve Bayes Classifier for the larger Heart Disease dataset and evaluated its performance as illustrated with R codes in Apendix II, Code block label "a". The confusion matrix is as follows:

	Actual (no disease, 0)	Actual (1, disease)
Predicted (no disease, 0)	TP = 54	FP = 13
Predicted (disease, 1)	FN = 7	TN = 48

Table 4. 9: Confusion Matrix for NB on the larger dataset, with no PCA

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$Accuracy = 83.60656\%$$

$$Sensitivity = TP/(TP + FN) = TP/P = 48/61 = 0.79$$

$$Specificity = TN/(TN + FP) = TN/N = 54/61 = 0.86$$

3.9.2 Building and evaluating NB on the smaller dataset, with no PCA

We built a Naïve Bayes Classifier for the smaller Heart Disease dataset and evaluated its performance as illustrated with R codes in Apendix II, Code block label "b". The confusion matrix is as follows:

	Actual (no disease, 0)	Actual (1, disease)
Predicted (no disease, 0)	TP = 16	FP = 5
Predicted (disease, 1)	FN = 2	TN = 13

Table 4. 10: Confusion Matrix for NB on the smaller dataset, with no PCA

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$Accuracy = 80.55556$$

$$Sensitivity = TP/(TP + FN) = TP/P = 13/18 = 0.72$$

$$Specificity = TN/(TN + FP) = TN/N = 16/18 = 0.89$$

3.9.3 Building and evaluating LR on the larger dataset, with no PCA

We built a Logistics Regression Classifier for the larger Heart Disease dataset and evaluated its performance as illustrated with R codes in Apendix II, Code block label "c". The confusion matrix is as follows:

	Actual (no disease, 0)	Actual (1, disease)

Predicted (no disease, 0)	TP = 56	FP = 11
Predicted (disease, 1)	FN = 5	TN = 50

Table 4. 11: Confusion Matrix for LR on the larger dataset, with no PCA

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$Accuracy = 86.88525$$

$$Sensitivity = TP/(TP + FN) = TP/P = 50/61 = 0.82$$

$$Specificity = TN / (TN + FP) = TN/N = 56/61 = 0.92$$

3.9.4 Building and evaluating LR on the smaller dataset, with no PCA

We built a Logistics RegressionClassifier for the smaller Heart Disease dataset and evaluated its performance as illustrated with the following R codes.

with R codes in Apendix II, Code block label "d". The confusion matrix is as follows:

	Actual (no disease, 0)	Actual (1, disease)
Predicted (no disease, 0)	TP = 16	FP = 6
Predicted (disease, 1)	FN = 2	TN = 12

Table 4. 12: Confusion Matrix for LR on the smaller dataset, with no PCA

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$accuracy = 77.77778$$

$$Sensitivity = TP/(TP + FN) = TP/P = 12/18 = 0.67$$

$$Specificity = TN / (TN + FP) = TN/N = 16/18 = 0.89$$

DIMENSION REDUCTION WITH PCA, FOR THE HEART DATA

PCA on Normalized Heart Dataset was done, as seen in Appendix II, block code label "e".

The following is a scree plot of the analysis.

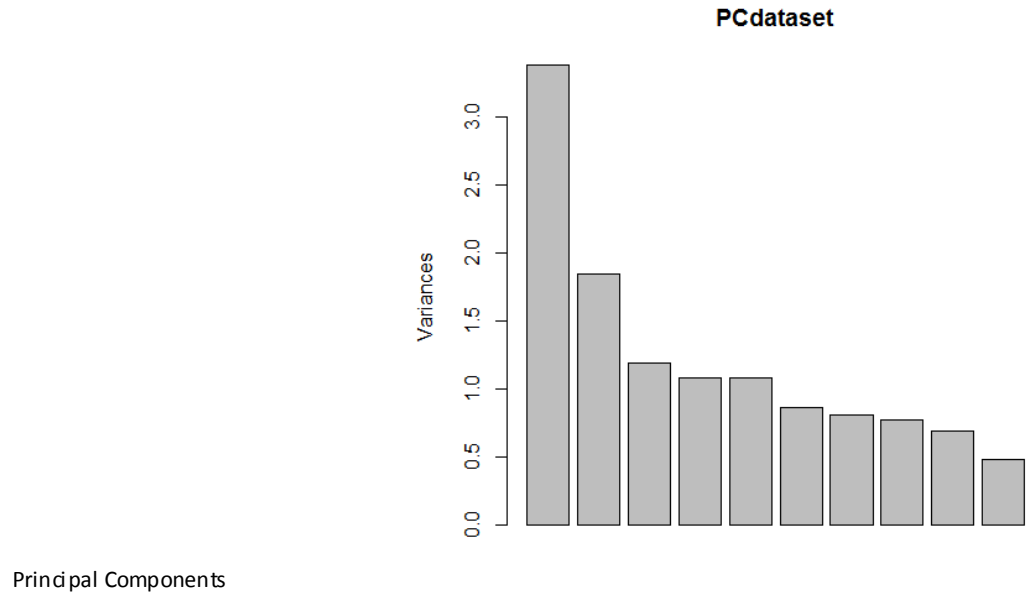


Figure 4. 2: A scree plot showing the levels at which each components explains the variability.

We looked at the proportion of variances explained by the PC components, we realized that the first 12 components explain the overall variability. The next step we performed was examining the rotations of these selected PC objects individually, in order to select the features that yields the PC's greatest magnitude. This feature (or variable) will be part of the predictors.

Thus, the predictor variables selected are V2, V4, V5, V6, V7, V8, V9 and V11.

3.9.5 Building and evaluating NB on the larger dataset, with PCA

We built a Naïve Bayes Classifier for the larger Heart Disease dataset, with predictor features as selected by the PCA, and evaluated its performance as illustrated with R codes in Apendix II, Code block label "f". The confusion matrix is as follows:

	Actual (no disease, 0)	Actual (1, disease)
Predicted (no disease, 0)	TP = 44	FP = 11
Predicted (disease, 1)	FN = 17	TN = 50

Table 4. 13: Confusion Matrix for NB on the larger dataset, with PCA

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$Accuracy = 77.04918$$

$$Sensitivity = TP/(TP + FN) = TP/P = 50/61 = 0.82$$

$$Specificity = TN/(TN + FP) = TN/N = 44/61 = 0.72$$

3.9.6 Building and evaluating NB on the smaller dataset, with PCA

We built a Naïve Bayes Classifier for the smaller Heart Disease dataset, with predictor features as selected by the PCA, and evaluated its performance as illustrated with R codes in Apendix II, Code block label "g". The confusion matrix is as follows:

	Actual (no disease, 0)	Actual (1, disease)
Predicted (no disease, 0)	TP = 17	FP = 5
Predicted (disease, 1)	FN = 1	TN = 13

Table 4. 14: Confusion Matrix for NB on the smaller dataset, with PCA

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$Accuracy = 83.33333$$

$$Sensitivity = TP/(TP + FN) = TP/P = 13/18 = 0.72$$

$$Specificity = TN/(TN + FP) = TN/N = 17/18 = 0.94$$

3.9.7 Building and evaluating LR on the larger dataset, with PCA

We built a Logistics Regression Classifier for the larger Heart Disease dataset, with predictor features as selected by the PCA, and evaluated its performance as illustrated with R codes in Apendix II, Code block label "h". The confusion matrix is as follows:

	Actual (no disease, 0)	Actual (1, disease)
Predicted (no disease, 0)	TP = 47	FP = 8
Predicted (disease, 1)	FN = 14	TN = 53

Table 4. 15: Confusion Matrix for LR on the larger dataset, with PCA

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$Accuracy = 81.96721$$

$$Sensitivity = TP/(TP + FN) = TP/P = 53/61 = 0.87$$

$$Specificity = TN/(TN + FP) = TN/N = 47/61 = 0.77$$

3.9.8 Building and evaluating LR on the smaller dataset, with PCA

We built a Logistics Regression Classifier for the smaller Heart Disease dataset, with predictor features as selected by the PCA, and evaluated its performance as illustrated with R codes in Apendix II, Code block label "i". The confusion matrix is as follows:

	Actual (no disease, 0)	Actual (1, disease)
Predicted (no disease, 0)	TP = 17	FP = 8
Predicted (disease, 1)	FN = 1	TN = 10

Table 4. 16: Confusion Matrix for LR on the smaller dataset, with PCA

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$Accuracy = 75$$

$$Sensitivity = TP/(TP + FN) = TP/P = 10/18 = 0.56$$

$$Specificity = TN / (TN + FP) = TN/N = 17/18 = 0.94$$

3.10 Summary of Results

BREAST CANCER DIAGNOSTIC MODEL EVALUATION

Model Description	Accuracy	Sensitivity	Specificity
Larger dataset, no PCA, NB	96.72131	1.0	0.93
Smaller dataset, no PCA, NB	97.22222	1.0	0.81
Larger dataset, no PCA, LR	95.90164	0.98	0.93
Smaller dataset, no PCA, LR	94.44444	0.94	0.94
Larger dataset, PCA, NB	96.72131	1.0	0.93
Smaller dataset, PCA, NB	97.22222	1.0	0.94
Larger dataset, PCA, LR	96.72131	0.98	0.95
Smaller dataset, PCA, LR	91.66667	0.89	0.94

Table 4. 17: A table showing the Classification Accuracies of NB and LR for the Breast Cancer Datasets, with their respective sensitivities and specificities.

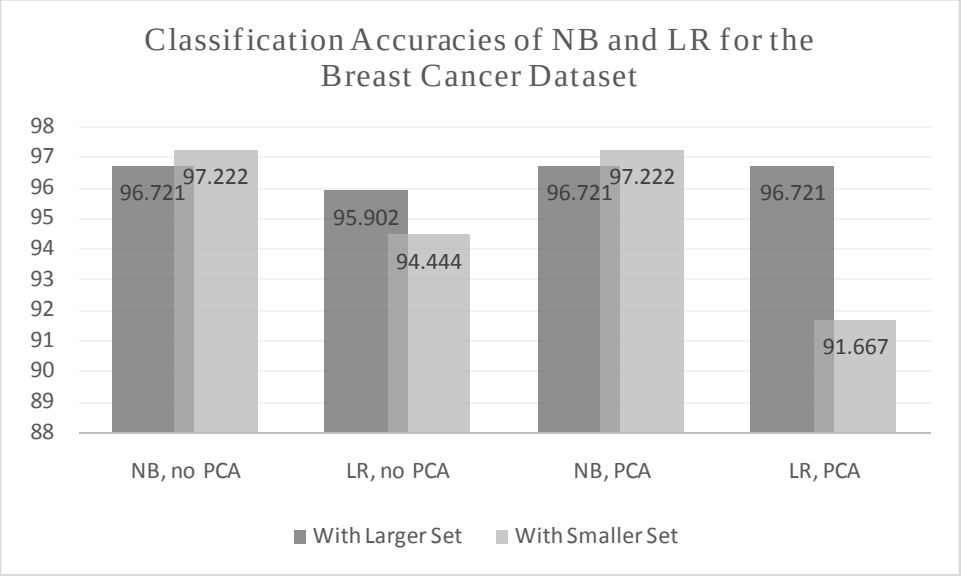


Figure 4. 3: A multiple Bar Chart showing the Classification Accuracies of NB and LR for the Breast Cancer Datasets.

HEART DISEASE DIAGNOSTIC MODEL EVALUATION

Model Description	Accuracy	Sensitivity	Specificity
Larger dataset, no PCA, NB	83.60656	0.79	0.86
Smaller dataset, no PCA, NB	80.55556	0.72	0.89
Larger dataset, no PCA, LR	86.88525	0.82	0.92
Smaller dataset, no PCA, LR	77.77778	0.67	0.89
Larger dataset, PCA, NB	77.04918	0.82	0.72
Smaller dataset, PCA, NB	83.33333	0.72	0.94
Larger dataset, PCA, LR	81.96721	0.87	0.77
Smaller dataset, PCA, LR	75	0.56	0.94

Table 4. 18: A table showing the Classification Accuracies of NB and LR for the Breast Cancer Datasets, with their respective sensitivities and specificities.

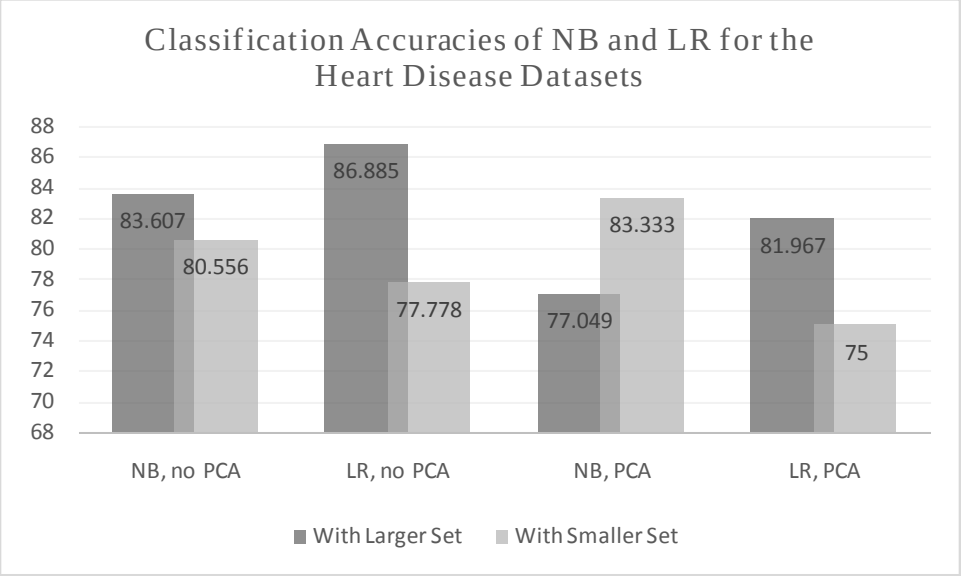


Figure 4. 4: A multiple Bar Chart showing the Classification Accuracies of NB and LR for the Heart Disease Datasets.

AVERAGE ACCURACY

Model Description	Breast Cancer	Heart Disease	Average Accuracy
Larger Dataset, no PCA, NB	96.721	83.607	90.164
Smaller Dataset, no PCA, NB	97.222	80.556	88.889
Larger Dataset, no PCA, LR	95.902	86.885	91.3935
Smaller Dataset, no PCA, LR	94.444	77.778	86.111
Larger Dataset, PCA, NB	96.721	77.049	86.885
Smaller Dataset, PCA, NB	97.222	83.333	90.2775
Larger Dataset, PCA, LR	96.721	81.967	89.344
Smaller Dataset, no PCA, LR	91.667	75	83.3335

Table 4. 19: A table showing the Average Classification Accuracies of NB and LR for the two datasets.

CHAPTER FIVE: SUMMARY, CONCLUSION AND RECOMMENDATION

4.1 Summary

The study aimed at studying the effect of variable selection and dataset size on the accuracy of Naïve Bayesian classifier (NB) and Logistics Regression (LR), in order to further understand the basic statistical techniques used for classification, and their underlying properties. The study was carried out to measure the performances of the classification models using the following evaluation metrics: accuracy, sensitivity and specificity.

These two models were used because they are a good representation of a linear discriminative-generative pair. NB has been reported to do well with a variety of classification problems, and because of its competitiveness with others (such as LR, etc.), it became a subject of debate as to which classifier performs better in the case of real life application. The information obtained carries the basic requirement that is needed to inform us about our expectation from a choice of any of the classifiers.

The study revealed that NB classifier is very strong when the learning data is small and reasonable enough to make judgment. PCA Variable selection, for the sake of dimension reduction, was applied to identify the variables that are statistically important for the prediction of respondents' outcomes (healthy or not-healthy), variables were the parameters through which we selected the most important predictor variables of the dependent variable. In the NB and LR classifiers, variables are modelled through the conditional probabilities and logistics function, respectively, of the independent variables.

We tabulated the resulting performances in terms of the classifier model used, relative size of datasets, and the use and non-use of PCA on the datasets. We have a mixed performances of the models that we conclude *almost certainly, that NB accuracy decreases as the datasets grow in size while LR's increases with dataset growth*. We also realized thatPCA for feature selection may not have a good effect on the model's performance, since information is lost on

the bases of features' likelihood to be a good predictor. PCA for feature extraction is not applied in this work because of problems with feature interpretability - the assessment of interpretability relies on a whole lot of datasets collection conditions (background, preferences, and priorities, etc.).

The average classification accuracies as shown in table 4.19 reveals that LR on relatively large datasets without PCA yield the highest accuracy (91.39%), while for relatively small datasets, the highest classification accuracy was the NB classifier with a rate of 90.28%.

4.2 Conclusion

The results of our Classification models' analysis indicated that for medical diagnosis involving breast cancer and heart disease,LR for classification on relatively large datasets without the application of PCA (for variable selection) has the greatest accuracy, while NB with PCA (for variable/feature selection) tops the smaller dataset classification. These two accuracies are close enough, and high enough, which is an indication of high relevance of their selections in solving classification problems on datasets from this kind of domain.

4.3 Recommendation

The following recommendations are offered to researchers and medical practitioners:

1. To build an automated system for medical diagnosis, one may consider the adoption of statistical learning algorithms, which have been proven to have accuracies high enough to make a good prediction.
2. Naïve Bayes Classifier and Logistics Regression are both powerful and simple, hence it is recommended for similar evaluation with categorical response.
3. Records on Patients' should be collected, with as much features (age, blood group, blood pressure, etc.) recorded as possible.

4.4 Recommendation and Suggestion for future research

This research is limited to datasets from medical domains, and the conclusions are drawn on the same base. The study reveals that an interaction between variable selection and dataset sizes affect the accuracy and precision of classifiers in a variety of ways, and therefore we recommend the following for future research

(i) An exploration of classifiers performance on datasets from other domains.

(ii) A better approach for variable selection for classification.

(iii) Comparison between linear and non-linear classifiers.

4.5 Contribution to knowledge

The work of other researchers have majorly been done on testing the superiority of a model over another on the basis of established theoretical concepts. But for real application, Bhowmik (2015), Kwon and Mun Sim (2013), and others agree in their research that the performance of a classifier differs across different varying datasets. This work sets a perspective for comparing the performances of linear classification models on the context of dataset features and size properties. This work specifically addresses the choice between naïve Bayesian classifier and Binary logistics regression for medical diagnosis of heart disease and breast cancer by considering obtainable dataset conditions from medical record keeping practices.

REFERENCES

Bhowmik, K. T. (2015). Naive Bayes vs Logistic Regression: Theory, Implementation and Experimental Validation. *Inteligencia Artificial Journal*. 18(56), 14-30.

Cerquides, J., and De Màntaras, R. L. (2003). Tractable Bayesian learning of tree augmented naïve Bayes models. *ICML*. 75-82.

Frank, E., Hall, M. and Pfahringer, B. (2002). Locally weighted naive bayes. *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*. 249-256.

Guyon, I., and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*. 3, 1157-1182.

Halperin, M., Blackwelder, W. C., and Verter, J. I. (1971). Estimation of the multivariate logistic risk function: a comparison of the discriminant function and maximum likelihood approaches. *Journal of chronic diseases*, 24(2), 125-158.

Janecek, A., Gansterer, W., Demel, M., and Ecker, G. (2008, September). On the relationship between feature selection and classification accuracy. *New Challenges for Feature Selection in Data Mining and Knowledge Discovery*, 90-105.

Kotsiantis, S. B., Zaharakis, I., and Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*. 160, 3-24.

Kwon, O. and Mun Sim, J. (2013). Effects of data set features on the performances of classification algorithms. *Journal of Expert Systems with Applications*, 40(5), 1847-1857.

Press, J. S., and Wilson, S. (1978). Choosing Between Logistic Regression and Discriminant Analysis. *Journal of the American Statistical Association*, 73(364), 699-705.

Purkayastha, P., Rallapalli, A., Murthy, N. B., Malapati, A., Yogeeswari, P., and Sriram, D. (2015). Effect of feature selection on kinase classification models. *Computational Intelligence in Medical Informatics*. 81-86.

Rish, I. (2003). An emperical study of the Naive Bayes classifier. T.J. *Watson Research Center Journal.*3(2), 41-46.

Steven C., Halbert W. and Beatrice A. (2001). Logistics Regression in the Medical Literature: Standards for Use and Reporting, with particular attention to one medical domain. *Journal of Clinical Epidemiology*. 54(10), 979-985.

Yoo JY and Yang D. (2015). Classification Scheme of Unstructured Text Document Using TF-IDF and Naïve Bayes Classifier. *Advance Science and Technology Letters*. 111(50), 263-266.

Yu, L. and Liu, H (2004). Efficient Feature Selection via Analysis of Relevance and Redundancy. *Journal of Machine Learning Research*. 5, 1205 – 1224.

Zhang, H. (2004). The Optimality of Naive Bayes. *American Society for Artificial Intelligence.*AA, 1(2), 3.

APENDIX I

BREAST CANCER DATASET ANALYSIS WITH R

a. NB on the larger dataset, with no PCA

```
> require(e1071)
> #reading the dataset into R
> train = read.table("TrainSet_C.txt")
> test = read.table("TestSet_C.txt")
> model = naiveBayes(as.factor(V11)~. , data = train)
> #predicting the testdata
> pred = predict(model, test[,-10])
> tab = table(pred, test$V11)
```

```
> tab

pred  2  4

   2 57  0

   4  4 61

> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)

> accuracy

[1] 96.72131

2- benign, or safe. 4- malignant, or dangerous

>#Sensitivity = TP/(TP+FN)= TP/P = 61/61 = 1.0

>#Specificity = TN / (TN + FP) = TN/N = 57/61 = 0.93
```

b. NB on the smaller dataset, with no PCA

```
> require(e1071)

> train = read.table("smallTrainSet_C.txt")

> small = read.table("smallTestSet_C.txt")

> test = read.table("smallTestSet_C.txt")

> model = naiveBayes(as.factor(V11)~. , data = train)

> #predicting the testdata

> pred = predict(model, test[,-10])

> tab = table(pred, test$V11)

> tab

pred  2  4

   2 17  0

   4  1 18
```

```
> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)

> accuracy

[1] 97.22222

>#Sensitivity = TP/(TP+FN)= TP/P = 18/18 = 1.0

>#Specificity = TN / (TN + FP) = TN/N = 17/21 = 0.81
```

c. LR on the larger dataset, with no PCA

```
> train = read.table("TrainSet_C.txt")

> test = read.table("TestSet_C.txt")

> lrmodel = glm(as.factor(V11)~., data=train, family='binomial')

> lrmodel

Call:  glm(formula = as.factor(V11) ~ ., family = "binomial", data = train)

Coefficients:

(Intercept)           V2           V3           V4           V5           V6
V71

   -16.7099       0.5827       0.7648       0.7312       0.2242      -0.1630
6.4279

       V710          V72          V73          V74          V75          V76
V77

    27.6596       6.7430       9.5115      10.5781      10.2134      26.8063
8.6468

        V78          V79           V8           V9          V10

    27.8197      23.7605      -0.1176       0.2262       0.4189


Degrees of Freedom: 359 Total (i.e. Null);  341 Residual

Null Deviance:      499.1
```

```
Residual Deviance: 36.53        AIC: 74.53

> #confusion matrix

> predict <- predict(lrmodel, test[,-10], type = 'response')

> tab = table(predict > 0.5, test$V11)

> tab

         2  4

  FALSE 57  1

  TRUE   4 60

> #accuracy

> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)

> accuracy

[1] 95.90164

>#Sensitivity = TP/(TP+FN)= TP/P = 60/61 = 0.98

>#Specificity = TN / (TN + FP) = TN/N = 57/61 = 0.93
```

d. LR on the smaller dataset, with no PCA

```
> train = read.table("smallTrainSet_C.txt")

> lrmodel = glm(as.factor(V11)~., data=train, family='binomial')

> lrmodel

Call:  glm(formula = as.factor(V11) ~ ., family = "binomial", data = train)

Coefficients:

(Intercept)           V2           V3           V4           V5           V6
V71

   -32.0280       3.6737      -2.2647       1.6370       2.1221       5.8183     -
27.5855
```

```
       V710          V72          V73          V74          V75          V76
V77

     2.4036     -26.8837     -19.4968      23.4190     -11.8907      -6.6687
7.5359

        V78          V79           V8           V9          V10

     0.6853      33.9302       0.6507       1.8044      -4.7301

Degrees of Freedom: 103 Total (i.e. Null);  85 Residual

Null Deviance:      144.2

Residual Deviance: 1.363e-09    AIC: 38

> #confusion matrix

> predict <- predict(lrmodel, test[,-10], type = 'response')

> tab = table(predict > 0.5, test$V11)

> tab

         2  4

  FALSE 17  1

  TRUE   1 17

> #accuracy

> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)

> accuracy

[1] 94.44444

> tab=table(predict > 0.5, test$V11)

> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)

>#Sensitivity = TP/(TP+FN)= TP/P = 17/18 = 0.94

>#Specificity = TN / (TN + FP) = TN/N = 17/18 = 0.94
```

e. DIMENSION REDUCTION WITH PCA

```
> train = read.table("TrainSet_C.txt")

> test = read.table("TestSet_C.txt")

> dataset = rbind(train, test)

> #converting dataset to numeric

> dataset.numeric = data.matrix(dataset, rownames.force = NA)

> PCdataset = prcomp(dataset.numeric[,-10])

> summary(PCdataset)

Importance of components:

                          PC1     PC2    PC3     PC4     PC5     PC6     PC7

Standard deviation     6.8509 2.39067 2.2253 2.03136 1.91148 1.85087 1.50691

Proportion of Variance 0.6322 0.07698 0.0667 0.05558 0.04921 0.04614 0.03059

Cumulative Proportion  0.6322 0.70916 0.7759 0.83144 0.88066 0.92680 0.95738

                           PC8     PC9

Standard deviation     1.44072 1.04321

Proportion of Variance 0.02796 0.01466

Cumulative Proportion  0.98534 1.00000

> PCdataset

Standard deviations:

[1] 6.850891 2.390667 2.225272 2.031356 1.911483 1.850868 1.506914 1.440716

[9] 1.043210

Rotation:

          PC1         PC2          PC3         PC4        PC5          PC6

V2  0.3037476  0.55388944  0.556219230 -0.24801590  0.4189282 -0.19480717

V3  0.4508220 -0.01245428  0.146949603  0.07017973 -0.3555719  0.20570356

V4  0.4305944  0.04862667  0.140419457  0.04098995 -0.3169323  0.24523862

V5  0.3587592 -0.65360663  0.120508340  0.30053922  0.4920208 -0.20587277
```

```
V6  0.2800133 -0.03255804 -0.002708113  0.05594347 -0.4609183 -0.24198965
V7  0.1545061  0.49504701 -0.385322687  0.74375549  0.1661852 -0.04079866
V8  0.3191338 -0.11340414  0.028227850  0.02124146  0.1421660  0.40587901
V9  0.4053433  0.05978288 -0.695762487 -0.53329633  0.2089085 -0.01673618
V10 0.1481488 -0.03031734 -0.037721910 -0.02002116 -0.2272007 -0.76939538
             PC7           PC8         PC9
V2   0.03609591  0.1220049416 -0.02033300
V3  -0.23605304 -0.0959634299 -0.73239991
V4  -0.40597365 -0.1383569348  0.66804252
V5  -0.16156447  0.1683452273  0.01824472
V6   0.45455393  0.6553918617  0.10115889
V7  -0.00571384  0.0007804222 -0.00975761
V8   0.72619271 -0.4083750610  0.07049878
V9  -0.10462887  0.0898134378 -0.02638042
V10  0.08428158 -0.5692515366  0.02385205
> screeplot(PCdataset)
```

f. NB on the larger dataset, with PCA

```
> train = read.table("TrainSet_C.txt")
> test = read.table("TestSet_C.txt")
> #now fitting a Naive Bayes
> require(e1071)
Loading required package: e1071
Warning message:
package 'e1071' was built under R version 3.2.5
```

```
> model = naiveBayes(as.factor(V11)~V3+V5+V6+V7+V8+V9+V10 , data = train)

> model

Naive Bayes Classifier for Discrete Predictors

Call:

naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:

Y

  2   4

0.5 0.5

Conditional probabilities:

   V3

Y       [,1]      [,2]

  2 1.272222 0.7233577

  4 6.600000 2.7636912



   V5

Y       [,1]      [,2]

  2 1.355556 0.9837917

  4 5.694444 3.2752623



   V6

Y       [,1]      [,2]

  2 2.083333 0.9965023

  4 5.205556 2.3151124



   V7
```

```
Y             ?           1          10           2           3           4
  2 0.038888889 0.833333333 0.000000000 0.044444444 0.038888889 0.011111111
  4 0.005555556 0.066666667 0.522222222 0.038888889 0.061111111 0.050000000
   V7
Y             5           6           7           8           9
  2 0.027777778 0.000000000 0.005555556 0.000000000 0.000000000
  4 0.083333333 0.022222222 0.027777778 0.083333333 0.038888889


   V8
Y       [,1]     [,2]
  2 2.100000 1.233155
  4 5.944444 2.348712


   V9
Y       [,1]      [,2]
  2 1.238889 0.9879944
  4 5.738889 3.3798018


   V10
Y       [,1]      [,2]
  2 1.050000 0.4759583
  4 2.722222 2.6299851


> pred = predict(model, test[,-10])
> tab = table(pred, test$V11)
> tab
```

```
pred  2  4
   2 57  0
   4  4 61
> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)
> accuracy
[1] 96.72131
>#Sensitivity = TP/(TP+FN)= TP/P = 61/61 = 1.0
>#Specificity = TN / (TN + FP) = TN/N = 57/61 = 0.93
```

g. NB on the smaller dataset, with PCA

```
> train = read.table("SmallTrainSet_C.txt")
> test = read.table("SmallTestSet_C.txt")
> #now fitting a Naive Bayes
> require(e1071)
Loading required package: e1071
Warning message:
package 'e1071' was built under R version 3.2.5
> model = naiveBayes(as.factor(V11)~V3+V5+V6+V7+V8+V9+V10 , data = train)
> model


Naive Bayes Classifier for Discrete Predictors


Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)
```

A-priori probabilities:

Y

2 4

0.5 0.5

Conditional probabilities:

V3

Y [,1] [,2]

2 1.269231 0.6892845

4 7.057692 2.7037625

V5

Y [,1] [,2]

2 1.230769 0.546482

4 5.538462 3.250711

V6

Y [,1] [,2]

2 1.980769 0.5045046

4 5.653846 2.5196064

V7

Y ? 1 10 2 3 4

2 0.01923077 0.84615385 0.00000000 0.03846154 0.05769231 0.00000000

4 0.01923077 0.03846154 0.59615385 0.00000000 0.00000000 0.07692308

```
   V7

Y            5          6          7          8          9

  2 0.03846154 0.00000000 0.00000000 0.00000000 0.00000000

  4 0.07692308 0.01923077 0.01923077 0.13461538 0.01923077


   V8

Y       [,1]     [,2]

  2 2.057692 1.092101

  4 5.961538 2.019139


   V9

Y       [,1]      [,2]

  2 1.173077 0.9846101

  4 5.884615 2.9944645


   V10

Y       [,1]      [,2]

  2 1.057692 0.3076452

  4 2.480769 2.5628598


> pred = predict(model, test[,-10])

> tab = table(pred, test$V11)

> tab


pred  2  4

   2 17  0
```

```
   4  1 18

> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)

> accuracy

[1] 97.22222


#Sensitivity = TP/(TP+FN)= TP/P = 18/18 = 1.0

#Specificity = TN / (TN + FP) = TN/N = 17/18 = 0.94
```

h. LR on the larger dataset, with PCA

```
> train = read.table("TrainSet_C.txt")

> test = read.table("TestSet_C.txt")

> lrmodel = glm(as.factor(V11)~V3+V5+V6+V7+V8+V9+V10, data=train,
family='binomial')

Warning message:

glm.fit: fitted probabilities numerically 0 or 1 occurred

> lrmodel


Call:  glm(formula = as.factor(V11) ~ V3 + V5 + V6 + V7 + V8 + V9 +

    V10, family = "binomial", data = train)


Coefficients:

(Intercept)           V3           V5           V6          V71         V710

 -13.841448     1.233617     0.079479     0.006885     5.875982    27.309273

       V72          V73          V74          V75          V76          V77

   7.068758     8.103414    10.190768     8.575202    29.319275     7.176867
```

```
       V78          V79           V8           V9          V10
  26.539851    23.966472     0.174457     0.349940     0.497550

Degrees of Freedom: 359 Total (i.e. Null);  343 Residual
Null Deviance:      499.1
Residual Deviance: 47.15        AIC: 81.15
> #confusion matrix
> predict <- predict(lrmodel, test[,-10], type = 'response')
> tab = table(predict > 0.5, test$V11)
> tab

        2  4
  FALSE 58  1
  TRUE   3 60
> #accuracy
> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)
> accuracy
[1] 96.72131

#Sensitivity = TP/(TP+FN)= TP/P = 60/61 = 0.98
#Specificity = TN / (TN + FP) = TN/N = 58/61 = 0.95
```

i. LR on the smaller dataset, with PCA

```
> train = read.table("SmallTrainSet_C.txt")
> test = read.table("SmallTestSet_C.txt")
```

```
> lrmodel = glm(as.factor(V11)~V3+V5+V6+V7+V8+V9+V10, data=train,
family='binomial')
Warning messages:
1: glm.fit: algorithm did not converge
2: glm.fit: fitted probabilities numerically 0 or 1 occurred
> lrmodel


Call:  glm(formula = as.factor(V11) ~ V3 + V5 + V6 + V7 + V8 + V9 +
    V10, family = "binomial", data = train)


Coefficients:
(Intercept)           V3           V5           V6          V71         V710
   -29.9412       3.8626       2.4999       6.2892     -24.6433      27.9622
        V72          V73          V74          V75          V76          V77
   -16.0111     -19.5448      60.7841      -5.7644      -7.4076      -6.1253
        V78          V79           V8           V9          V10
    30.4546      32.1634      -0.4375       2.0204      -8.9551

Degrees of Freedom: 103 Total (i.e. Null);  87 Residual
Null Deviance:      144.2
Residual Deviance: 2.24e-09     AIC: 34
> #confusion matrix
> predict <- predict(lrmodel, test[,-10], type = 'response')
> tab = table(predict > 0.5, test$V11)
> tab
```

```
         2  4
  FALSE 17  2
  TRUE   1 16
> #accuracy
> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)
> accuracy
[1] 91.66667
> #Sensitivity = TP/(TP+FN)= TP/P = 16/18 = 0.89
> #Specificity = TN / (TN + FP) = TN/N = 17/18 = 0.94
```

APENDIX II

THE HEART DISEASE DATASET ANALYSIS WITH R

a. NB on the larger dataset, with no PCA

```
> require(e1071)

> train = read.table("bHeart_Train.txt")

> test = read.table("bHeart_Test.txt")

> model = naiveBayes(as.factor(V14)~. , data = train)

> #predicting the testdata

> pred = predict(model, test[,-14])

> tab = table(pred, test$V14)

> tab

pred  0  1

   0 54 13

   1  7 48

> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)

> accuracy

[1] 83.60656



#Sensitivity = TP/(TP+FN)= TP/P = 48/61 = 0.79

#Specificity = TN / (TN + FP) = TN/N = 54/61 = 0.86
```

b. NB on the smaller dataset, with no PCA

```
> require(e1071)

> train = read.table("sHeart_Train.txt")

> test = read.table("sHeart_Test.txt")

> model = naiveBayes(as.factor(V14)~. , data = train)

> #predicting the testdata

> pred = predict(model, test[,-14])

> tab = table(pred, test$V14)

> tab

pred  0  1

   0 16  5

   1  2 13

> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)

> accuracy

[1] 80.55556

> #Sensitivity = TP/(TP+FN)= TP/P = 13/18 = 0.72

> #Specificity = TN / (TN + FP) = TN/N = 16/18 = 0.89
```

c. LR on the larger dataset, with no PCA

```
> train = read.table("bHeart_Train.txt")

> test = read.table("bHeart_Test.txt")

> lrmodel = glm(as.factor(V14)~., data=train, family='binomial')

> lrmodel
```

```
Call:  glm(formula = as.factor(V14) ~ ., family = "binomial", data = train)


Coefficients:
(Intercept)           V1           V2           V3           V4           V5
  -4.088963    -0.004946     1.382505     0.667588     0.013232     0.003784
         V6           V7           V8           V9          V10          V11
  -0.001748    -0.120558    -0.015618     0.550191     0.746267     0.049134
        V12          V13
  -0.046614     0.073577


Degrees of Freedom: 363 Total (i.e. Null);  350 Residual
Null Deviance:      504.6
Residual Deviance: 317.5        AIC: 345.5
> #confusion matrix
> predict <- predict(lrmodel, test[,-14], type = 'response')
> tab = table(predict > 0.5, test$V14)
> tab

         0  1
  FALSE 56 11
  TRUE   5 50
> #accuracy
> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)
> accuracy
[1] 86.88525
```

```
#Sensitivity = TP/(TP+FN)= TP/P = 50/61 = 0.82

#Specificity = TN / (TN + FP) = TN/N = 56/61 = 0.92
```

d. LR on the smaller dataset, with no PCA

```
> test = read.table("sHeart_Test.txt")

> train = read.table("bHeart_Train.txt")

> lrmodel = glm(as.factor(V14)~., data=train, family='binomial')

> lrmodel


Call:  glm(formula = as.factor(V14) ~ ., family = "binomial", data = train)


Coefficients:

(Intercept)           V1           V2           V3           V4           V5

  -4.088963    -0.004946     1.382505     0.667588     0.013232     0.003784

         V6           V7           V8           V9          V10          V11

  -0.001748    -0.120558    -0.015618     0.550191     0.746267     0.049134

        V12          V13

  -0.046614     0.073577


Degrees of Freedom: 363 Total (i.e. Null);  350 Residual

Null Deviance:      504.6

Residual Deviance: 317.5        AIC: 345.5

> #confusion matrix

> predict <- predict(lrmodel, test[,-14], type = 'response')

> tab = table(predict > 0.5, test$V14)
```

```
> tab

         0  1

  FALSE 16  6

  TRUE   2 12

> #accuracy

> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)

> accuracy

[1] 77.77778


#Sensitivity = TP/(TP+FN)= TP/P = 12/18 = 0.67

#Specificity = TN / (TN + FP) = TN/N = 16/18 = 0.89
```

e. DIMENSION REDUCTION WITH PCA

```
> train = read.table("bHeart_Train.txt")

> test = read.table("bHeart_Test.txt")

> dataset = rbind(train, test)

> normdataset = scale(dataset)

> PCdataset = prcomp(normdataset[,-14])

> summary(PCdataset)

Importance of components:

                          PC1    PC2     PC3    PC4     PC5     PC6     PC7

Standard deviation     1.838 1.3591 1.09127 1.0394 1.03854 0.92763 0.89874

Proportion of Variance 0.260 0.1421 0.09161 0.0831 0.08297 0.06619 0.06213

Cumulative Proportion  0.260 0.4021 0.49369 0.5768 0.65976 0.72595 0.78808

                           PC8     PC9    PC10   PC11    PC12    PC13
```

```
Standard deviation     0.87785 0.83116 0.69617 0.6224 0.54833 0.34747

Proportion of Variance 0.05928 0.05314 0.03728 0.0298 0.02313 0.00929

Cumulative Proportion  0.84736 0.90050 0.93778 0.9676 0.99071 1.00000


> PCdataset

Standard deviations:

 [1] 1.8384505 1.3591235 1.0912747 1.0393654 1.0385394 0.9276291 0.8987412

 [8] 0.8778452 0.8311625 0.6961701 0.6224357 0.5483324 0.3474706


Rotation:

            PC1          PC2         PC3          PC4         PC5          PC6

V1   0.32099390  0.047200544 -0.20918043 -0.019184895 -0.46622546 -0.245833988

V2   0.03640132  0.229902991 -0.08353086  0.332450460  0.57695398 -0.585524887

V3   0.24025582  0.368235384  0.20747747  0.161566312  0.15394874  0.062821596

V4   0.12498853  0.243819195 -0.50732214 -0.574797491 -0.04320642 -0.063994195

V5   0.12510355  0.044881821  0.43648101 -0.500574330 -0.05521924 -0.563777778

V6   0.07935355 -0.071028754 -0.63571117  0.270279762 -0.04947446 -0.187131989

V7   0.28432532 -0.205937039  0.04173372 -0.009111864 -0.03598388 -0.181196013

V8  -0.07542257 -0.473228264 -0.12466910 -0.346750840  0.48559322  0.130612343

V9   0.20884981  0.388387941 -0.12937408 -0.265136570  0.39214436  0.284029854

V10  0.32566322  0.296321067  0.06636813  0.095240480 -0.06810698  0.238498892

V11  0.44749985 -0.007316093  0.12059657  0.031785642  0.02316520  0.219174244

V12  0.43172512 -0.361556682 -0.01014442  0.061376522  0.06255995  0.017256923

V13  0.42130519 -0.327152845  0.03355904  0.067247804  0.13686592 -0.006266921

            PC7          PC8         PC9          PC10         PC11         PC12

V1   0.35407010  0.16463050  0.25479308  0.241130249 -0.537659587 -0.05343155
```

```
V2   0.34591159 -0.03039656 -0.15365947 -0.008262923 -0.023194317 -0.08447007

V3  -0.35688651  0.34181359  0.40054061 -0.498804180 -0.232840350 -0.04710720

V4   0.17145871  0.09854707 -0.11987381 -0.451481163  0.268799204 -0.04032020

V5  -0.29911548 -0.33814400  0.08598121  0.073489217 -0.004199266  0.05014414

V6  -0.59928962 -0.31989375  0.08989853  0.058881311 -0.011413315 -0.01939453

V7  -0.32404311  0.61312657 -0.57827928  0.144422230  0.025472923 -0.03722963

V8  -0.03386342 -0.04690155 -0.02211531 -0.124737925 -0.595274287 -0.07249201

V9  -0.09869949  0.10882405  0.16750036  0.641872328  0.065391334  0.12651714

V10  0.01937342 -0.39159693 -0.52025036 -0.144591934 -0.325413139  0.42176177

V11  0.04618570 -0.28910149 -0.09940128  0.021124273  0.099218047 -0.78088089

V12  0.11708957 -0.01062679  0.16892530 -0.066612877  0.173058631  0.13024749

V13  0.13222962 -0.04176173  0.22172740 -0.077920782  0.280722390  0.39497392

             PC13

V1  -0.073110037

V2   0.025925023

V3   0.002425808

V4  -0.002605618

V5   0.034826692

V6  -0.021834822

V7  -0.050692569

V8  -0.052209770

V9   0.046130540

V10  0.045542103

V11 -0.141080441

V12  0.762298514

V13 -0.617883534
```

```
> screeplot(PCdataset)
```

f. NB on the larger dataset, with PCA

```
> train = read.table("bHeart_Train.txt")

> test = read.table("bHeart_Test.txt")

> #now fitting a Naive Bayes

> require(e1071)

Loading required package: e1071

Warning message:

package 'e1071' was built under R version 3.2.5

> model = naiveBayes(as.factor(V14)~V2+V4+V5+V6+V7+V8+V9+V11 , data = train)

> model


Naive Bayes Classifier for Discrete Predictors

Call:

naiveBayes.default(x = X, y = Y, laplace = laplace)


A-priori probabilities:

Y

 0  1

0.5 0.5


Conditional probabilities:
```

V2

Y [,1] [,2]

0 0.6098901 0.4891203

1 0.8406593 0.3670031

V4

Y [,1] [,2]

0 128.0604 19.27818

1 134.8791 18.83093

V5

Y [,1] [,2]

0 232.6593 76.07981

1 255.1429 73.87216

V6

Y [,1] [,2]

0 -0.02197802 0.9858448

1 0.01648352 1.0053738

V7

Y [,1] [,2]

0 0.4890110 0.7987491

1 0.6813187 1.1646650

```
  V8
Y      [,1]     [,2]
 0 149.5385 23.59362
 1 135.4835 22.54394

  V9
Y       [,1]     [,2]
 0 0.1043956 0.7689287
 1 0.5769231 0.4954103

  V11
Y        [,1]     [,2]
 0 -3.4175824 5.258124
 1  0.4505495 3.717255

> pred = predict(model, test[,-14])
> tab = table(pred, test$V14)
> tab

pred  0  1
   0 44 11
   1 17 50
> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)
> accuracy
[1] 77.04918
```

```
> #Sensitivity = TP/(TP+FN)= TP/P = 50/61 = 0.82

> #Specificity = TN / (TN + FP) = TN/N = 44/61 = 0.72
```

g. NB on the smaller dataset, with PCA

```
> train = read.table("sHeart_Train.txt")

> test = read.table("sHeart_Test.txt")

> #now fitting a Naive Bayes

> require(e1071)

Loading required package: e1071

Warning message:

package ‘e1071’ was built under R version 3.2.5

> model = naiveBayes(as.factor(V14)~V2+V4+V5+V6+V7+V8+V9+V11 , data = train)

> model


Naive Bayes Classifier for Discrete Predictors


Call:

naiveBayes.default(x = X, y = Y, laplace = laplace)


A-priori probabilities:

Y

  0   1

0.5 0.5


Conditional probabilities:
```

```
   V2

Y        [,1]      [,2]

  0 0.7307692 0.4478876

  1 0.8653846 0.3446423


   V4

Y       [,1]     [,2]

  0 131.8462 15.29242

  1 132.6731 18.93200


   V5

Y       [,1]     [,2]

  0 217.9423 81.39673

  1 249.6346 58.97458


   V6

Y        [,1]      [,2]

  0 0.1153846 0.3226025

  1 0.1153846 0.3226025


   V7

Y        [,1]      [,2]

  0 0.5769231 0.8710182

  1 0.9038462 0.9952754


   V8
```

```
Y       [,1]     [,2]
  0 150.1538 23.69196
  1 136.1154 23.43590

   V9
Y        [,1]      [,2]
  0 0.1923077 0.3979586
  1 0.6538462 0.4803845

   V11
Y         [,1]     [,2]
  0 -2.3846154 5.088063
  1  0.2307692 3.998114

> pred = predict(model, test[,-14])
> tab = table(pred, test$V14)
> tab

pred  0  1
   0 17  5
   1  1 13
> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)
> accuracy
[1] 83.33333

#Sensitivity = TP/(TP+FN)= TP/P = 13/18 = 0.72
```

```
#Specificity = TN / (TN + FP) = TN/N = 17/18 = 0.94
```

h. LR on the larger dataset, with PCA

```
> train = read.table("bHeart_Train.txt")

> test = read.table("bHeart_Test.txt")

> dataset = rbind(train, test)

> lrmodel = glm(as.factor(V14)~V2+V4+V5+V6+V7+V8+V9+V11, data=train,
family='binomial')

> lrmodel


Call:  glm(formula = as.factor(V14) ~ V2 + V4 + V5 + V6 + V7 + V8 +
    V9 + V11, family = "binomial", data = train)


Coefficients:
(Intercept)           V2           V4           V5           V6           V7
  -0.491498     1.433162     0.014678     0.004265    -0.004708     0.010545
         V8           V9          V11
  -0.025576     1.006954     0.153995


Degrees of Freedom: 363 Total (i.e. Null);  355 Residual
Null Deviance:      504.6
Residual Deviance: 359.6        AIC: 377.6
> #confusion matrix
> predict <- predict(lrmodel, test[,-14], type = 'response')
> tab = table(predict > 0.5, test$V14)
```

```
> tab


         0  1

  FALSE 47  8

  TRUE  14 53

> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)

> accuracy

[1] 81.96721


#Sensitivity = TP/(TP+FN)= TP/P = 53/61 = 0.87

#Specificity = TN / (TN + FP) = TN/N = 47/61 = 0.77
```

i. LR on the smaller dataset, with PCA

```
> train = read.table("sHeart_Train.txt")

> test = read.table("sHeart_Test.txt")

> dataset = rbind(train, test)

> lrmodel = glm(as.factor(V14)~V2+V4+V5+V6+V7+V8+V9+V11, data=train,
family='binomial')

> lrmodel


Call:  glm(formula = as.factor(V14) ~ V2 + V4 + V5 + V6 + V7 + V8 +

    V9 + V11, family = "binomial", data = train)


Coefficients:

(Intercept)           V2           V4           V5           V6           V7
```

```
  -0.597559     1.260145    -0.006329     0.005081    -0.678738     0.350516

         V8           V9          V11

  -0.011470     1.915981     0.061356


Degrees of Freedom: 103 Total (i.e. Null);  95 Residual

Null Deviance:      144.2

Residual Deviance: 107.4        AIC: 125.4

> #confusion matrix

> predict <- predict(lrmodel, test[,-14], type = 'response')

> tab = table(predict > 0.5, test$V14)

> tab

         0  1

  FALSE 17  8

  TRUE   1 10

> accuracy = 100 * sum(tab[row(tab)==col(tab)]) / sum(tab)

> accuracy

[1] 75

> #Sensitivity = TP/(TP+FN)= TP/P = 10/18 = 0.56

> #Specificity = TN / (TN + FP) = TN/N = 17/18 = 0.94
```