Classification of Breast Cancer using Logistic Regression

A Thesis Presented to the Department of Computer Science

African University of Science and Technology

In Partial Fulfilment of the Requirements for the Degree of Master of Science

By

Ude Anthony Anene

Abuja, Nigeria

July, 2019.

CERTIFICATION

This is to certify that the thesis titled “Classification of breast cancer using logistic regression” submitted to the school of postgraduate studies, African University of Science and Technology (AUST), Abuja, Nigeria for the award of the Master's degree is a record of original research carried out by Ude Anthony Anene in the Department of Computer Science.

CLASSIFICATION OF BREAST CANCER USING LOGISTIC REGRESSION

BY:

Ude Anthony Anene

A THESIS APPROVED BY THE COMPUTER SCIENCE DEPARTMENT

APPROVAL BY

Supervisor

Surname: Rajesh

First name: Prasad

Signature 

The Head of Department

Surname: DAVID

First name: Amos

Signature: 

ABSTRACT

Breast cancer is a prevalent disease that affects mostly women, an early diagnosis will expedite the treatment of this ailment. In recent times, Machine Learning (ML) techniques have been employed in biomedical and informatics to help fight breast cancer. This research work proposed an ML model for the classification of breast cancer. To achieve this we employed logistic regression (LR) and also compared our model's performance with other extant ML models namely, Support Vector Machine (SVM), Naïve Bayes (NB), and Multilayer Perceptron (MLP). The original Wisconsin Diagnostic Breast Cancer dataset (WDBC) was used. Our performance evaluation was done for two phases, i.e. Phase 1: when the WBCD is scaled (feature scaling) and Phase 2: when the dataset is not scaled. All models excluding MLP performed well when there is no feature scaling of dataset with f1-scores of (LR=97%, SVM = 97%, NB = 95%, MLP= 52%). However, when feature scaling is applied on dataset, the four models have f1-scores above 90% (SVM = 98%, LR = 97%, NB = 97%, MLP = 97%). Notably, the f1-score for LR in both cases did not change, hence to the best of our knowledge, we concluded that LR, given its simplicity and low time complexity is a good model to employ for binomial classification.

Keywords: Logistic regression, machine learning, supervised learning, features scaling, prediction models, and performance metrics*.*

Dedication

I dedicate this research work to my loving family, i.e. the family of Mr. & Mrs. Aaron A. Ude, for their unending and unmeasurable support in my academic pursuit and overall well-being.

Acknowledgement

God’s grace has kept me this far, I give all glory to him for his strength, mercy, and love which have ever been abundant in my life.

I greatly thank my supervisor, Dr. Rajesh Prasad, for his invaluable guidance, teaching, and contribution to making this work come to fruition. I am privileged to have worked under your experienced supervision, Sir.

I thank all my lecturers that have taught me in the duration of my Master’s degree, I appreciate and cherish their priceless knowledge and experience which they have shared with me.

I am thankful and grateful to African University of Science and Technology and its staff for this opportunity to be among the selected few in Africa to study in such a world-class university, I have become more knowledgeable than I was before I enrolled for this study.

Special thanks goes to my loving parents Mr. Aaron Ude & Mrs. Apollonia Ude and caring siblings, Chinedu, Nnamdi, Chidera and Ezinne for their encouraging words, prayers, financial and emotion support in every aspect of my life.

Finally, a big thanks to my friends and course mates, I am most grateful to all that have contributed in one way or the other to the success of my work. I wish you success in all your endeavours.

Table of content

List of Table

List of figures

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC	Area under Curve
BP	Backward Propagation
CSV	Comma Separated Values
DNA	Deoxyribonucleic acid
GRU	Gated Recurrent Unit
IDE	Integrated Development Environment
K-NN	K-Nearest Neighbour
LR	Logistic Regression
ML	Machine Learning
MLP	Multilayer Perceptron
NB	Naïve Bayes
ROC	Receiver Operating Characteristic
RF	Random Forest
SVM	Support Vector Machine
UCI	University of California, Irvine

CHAPTER ONE
INTRODUCTION

1.1 Research Background

Breast cancer is now one of the most prevailing cancers that affects humans, especially woman, and early diagnosis would go a long way to reducing the damage done by this cancer on its victims. Breast cancer's causes are multifactorial and involve family history, obesity, hormones, radiation therapy, and even reproductive factors. Every year, one million women are newly diagnosed with breast cancer, according to the report of the world health organization half of them would die, because it's usually late when doctors detect the cancer (Aaltonen et al., 1998). Breast cancer can be categorized into two, which are malignant breast cancer and benign breast cancer. The classification of breast cancer as either malignant or benign is possible by scientifically studying the features of breast tumours, lumps, or any abnormalities found in the breast. At the benign stage the cancer has less risk and is not life-threatening while cancer that is categorized as malignant is life-threatening (Huang, Chen, Lin, Ke, & Tsai, 2017). Malignant tumours expand to the neighbouring cells, which can spread to other parts, whereas benign masses can't expand to other tissues, the expansion is then only limited to the benign mass (Aaltonen et al., 1998; Huang et al., 2017).

To accurately classify breast cancer as benign or malignant, researchers have employed an aspect of Artificial intelligence (AI) which is machine learning. Machine learning algorithms are used to build models that accept as input, attributes that qualify a breast cancer case and produce as output a label for the type of the cancer, label 1 for being benign or label 2 for malignant.

Machine learning model such as Neutral network, Support vector machine (SVM), K Nearest Neighbour (KNN), Decision Tree, Naïve Bayes (NB), and logistic regression (LR), have all been used in the past to classify breast cancer. Accurate classification of breast cancer would translate to early detection, diagnosis, treatment and where possible full eradication of the cancer.

1.1.1 Data Mining

This can be seen as “mining knowledge in data” or rather as an extraction of information from a large or voluminous dataset (K.Srinivas, Rani, & A.Govrdhan, 2010). It is the most important aspect of machine learning (Kaymak, Helwan, & Uzun, 2017); whereas the salient focus aspect of data mining is the pattern recognition ability (Jothi, Rashid, & Husain, 2015). Data mining techniques can be applied to medical data records to trace and foresee salient pattern in order to save a life, increase treatment accuracy, reduce the cost of treatment and reduces human error (Manjusha, Sankaranarayanan, & Seena, 2015). Techniques such as abnormality detection, regression, clustering, summarization and association rule employ data mining. In data mining, there are various steps to be taken in finding meaningful patterns, namely:

i. Pre-processing – this involves cleaning, feature extraction, feature selection, and dimensionality reduction.
ii. Clustering – unsupervised learning technique by grouping a set of related data.
iii. Classification – this is a supervised machine learning technique; a data set (training data) is required in such a system to establish relationships between data items. Whenever a test data is supplied, it will classify such data based on the learnt relationship. In this research work, we will be focusing on classification.

1.1.2 Classification

Classification in data mining involves basically two processes: firstly it is the model training and with a test data to determine the class label of unknown test instances; secondly is the performance evaluation to check the accuracy of the classifier model, that is calculating the differences between the classified and actual values for each attribute tuple in the test dataset (Jouni, Issa, Harb, Jacquemod, & Leduc, 2016; Kaymak et al., 2017).

1.2 Problem statement

One of the problems of classification lies in the use of appropriate methods to fit the model depending on the nature of data. Which machine learning model would perform best in the presence of dependency among the data features, unbalanced data, and sparsely valued data features is still open research.

1.3 Research Aim and objectives

The aim is to develop a prediction system for detecting breast cancer.

The main objectives are:

1. Study and apply logistic regression for the classification of breast cancer.
2. Compare Logistic regression with other extant machine learning classification models on the same data set.
3. Performance analysis and conclusion.

1.4 Limitation of study

This paper is restricted to the study of logistic regression for the classification of breast cancer using Wisconsin Breast Cancer Dataset (WBCD) from UCI machine learning online repository. Performance of this model is measured using precision score, recall score and f1-score only.

1.5 Paper organization

This paper is broken into five chapters. Chapter one introduces the research essence, aim, and objective, in chapter two, a literature review of previous works related to this research work are discussed. Chapter three is all about the materials used and the methodology employed. In chapter four, performance analyses and discussion are done, finally, in chapter five, there is a summary of the work, conclusion, and recommendation for future work.

CHAPTER TWO

LITERATURE REVIEW

This chapter presents some basic concepts and terminologies such as: Data mining, Classification techniques. Furthermore, a review of previous related work done in this research topic is presented. This review is done to know the techniques, other authors employed for the classification of breast cancer. This review is cut through other machine learning algorithms that have been used for the classification of breast cancer and not only logistic regression. In the review, prediction accuracy is discussed as well as the techniques used in improving them.

2.1 Basic Terminologies and Concepts

Machine Learning (ML) is the science (and art) of programming computers so they can learn from data (Géron, 2017).

Machine learning can be defined in a more general way as:

ML as the field of study that gives computers the ability to learn without being explicitly programmed. – Arthur Samuel, 1959.

ML can also be defined in a more technical way as: A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T as measured by P, improves with experience E. – Tom Mitchell,1997.

There are several applications for ML, the most significant of which is data mining. People are often prone to making mistakes during analyses or, possibly, when trying to establish relationships between multiple features (Kotsiantis, Zaharakis, & Pintelas, 2006).

2.1.1 Data Pre-processing

Data pre-processing is one of the most data mining tasks which includes preparation and transformation of data into a suitable form of mining procedure. Data pre-processing aims to reduce the data size, find the relations between data, normalize data, remove outliers and extract features for data. It includes several techniques like data cleaning, integration, transformation and reduction (Alasadi & Bhaya, 2017).

2.1.2 Feature scaling

Feature scaling is a technique that is used to normalize the range of independent variables or features of data. In data pre-processing, it is also known as data normalization and is usually employed during the data pre-processing step.

2.1.3 Supervised Learning

Supervised machine learning is the search for algorithms that cogitate from externally supplied instances to give general hypotheses, which then infer predictions about future instances. In other words, the goal of supervised learning is to build an incisive model of the distribution of class labels in terms of predictor features. The resulting classifier is then used to assign class labels to the testing instances where the values of the predictor features are known, but the value of the class label is unknown (Kotsiantis et al., 2006).

In supervised learning as shown in Fig. 2.1, the learner is provided with two sets of data, a training set, and a test set. The idea is for the learner to “learn” from a set of labelled examples in the training set so that it can identify unlabeled examples in the test set with the highest possible accuracy (Learned-miller, 2014).

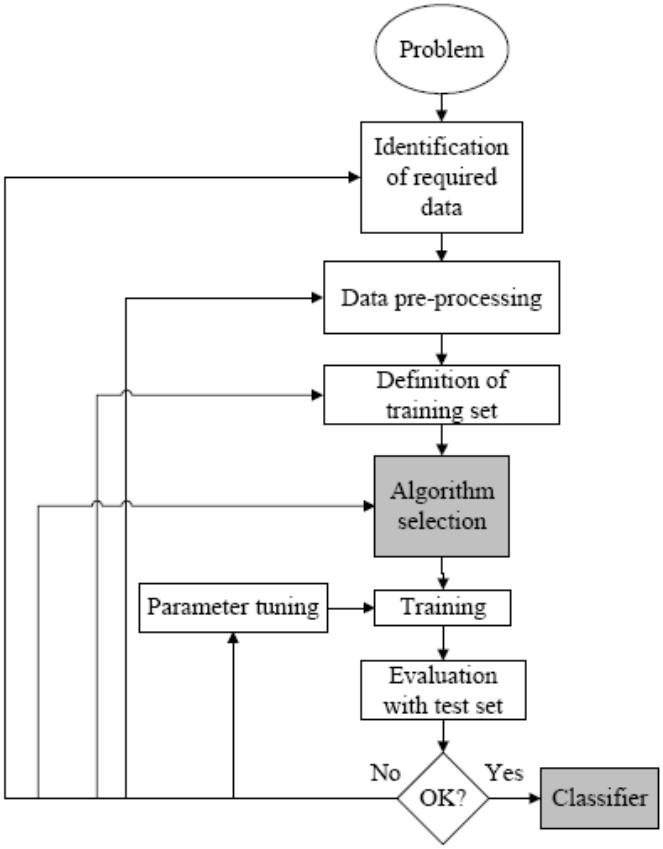


Figure 2. 1. The Processes of Supervised Machine Learning

2.1.4 Classification

We use the training dataset to get better boundary conditions which could be used to determine each target class. Once the boundary conditions are determined, the next task is to predict the target class. The whole process is known as classification. There exist two types of classification, the supervised and unsupervised.

In the supervised classification, available predefined knowledge is needed, whereas in the unsupervised classification sometimes referred to as clustering or exploratory data analysis, no predefined labelled data is needed(Agrawal, Gunopulos, & Leymann, n.d.; Tao, Faloutsos, Papadias, & Liu, 2004).

2.2 Literature Review

These are research work done in biomedical field related to the classification of cancer, especially breast cancer by using machine learning algorithms (supervised learning).

(Liu, 2018) in his paper “Research on logistic regression algorithm of breast cancer diagnose data by machine learning” used Logistic regression algorithm to classify dataset from breast cancer patients, the dataset was got from UCI Wisconsin repository. The author at first used the all 32 features of the dataset to train the model and at the end got an accuracy of 90%. Then, the author using a feature selection technique extracted two main features from the 32, which are maximum texture and maximum perimeter to achieve an accuracy of 96.5%, which is an improvement from the result got from the 32 features. In 2012, (Yusuff, Mohamad, Ngah, & Yahaya, 2012), data from mammogram was used by the logistic regression model to predict the risk’ factor of patient’s history, the prediction from logistic regression are used to verify to the prognosis made doctors and are also used to correct the incorrect predictions. The authors’ work can help be of assistance to radiologists to diagnose breast cancer correctly from using mammogram and referring to the patient’s history.

Using Naïve Bayesian as the classifier on Wisconsin dataset of 10 features (Rashmi, Lekha, & Bawane, 2016) tried to estimate the success and error of the algorithm for classification and prediction when data is chosen at random.

The Naïve Bayes model showed an approximated success rate of 85%-95% and an error rate of 10-15% for both classification and prediction.

(Amrane, Oukid, Gagaoua, & Ensari, 2018) in their research work, used two machine learning models: Naïve Bayesian and K Nearest Neighbour to classify the Original breast cancer dataset from UCI Machine Learning repository.

Their aim was to propose which of the two is the most effective. Using the same dataset, they applied the different algorithms on it and using cross validation as their performance measure. The result showed that KNN with 97.51% for accuracy lightly better than NB with 96.19%. However, the authors suggested that given a larger dataset that NB will likely perform better because KNN will be affected by its time complexity.

(Bazazeh & Shubair, 2016) did a comparative study for three popular machine learning algorithms for breast cancer classification namely: Support Vector Machine, Random Forest, and Bayesian Network. They also used the original Wisconsin breast cancer dataset from UCI Machine Learning Repository. The authors used K fold cross validation technique as the validation measure for the classifiers with k = 10. The parameters used for their comparison were accuracy, precision, recall, and AUC ROC and after doing their simulation on the dataset with the three classifiers, their result shows that SVM has the highest performance in terms of accuracy, precision, and specificity. However, they stated that in terms of correctly classifying tumours that RF scored the highest probability.

Furthermore, (Gupta & Gupta, 2018) did a comparative analysis of three widely used machine learning techniques namely: Multilayer perceptron (MLP), Decision Tree (C4.5), Support Vector Machine (SVM), K-Nearest Neighbour (KNN) performed on Wisconsin Breast Cancer dataset to predict the breast cancer recurrence.

The main objective of their work was to get the best classifier of the four in terms of accuracy, precision, recall, and R^2. In their work, they concluded that MLP performed better compared to other techniques, and in addition when 10-fold cross validation metric was used in used breast cancer prediction, MLP also performed better.

(Khourdifi & Bahaj, 2019) in their research work, applied four machine learning techniques namely SVM, RF, Naïve Bayes and K-NN on Wisconsin breast cancer dataset from UCI machine learning repository. The authors used Waikato Environment for Knowledge Analysis (Weka) software for the simulation of the algorithm. In their results, SVM had the overall performance in terms of effectiveness and efficiency.

(S Kharya, Dubey, & Soni, 2013) in their research work, carried out a comprehensive review of researches of some predictive models on breast cancer classification. The authors considered these machine learning algorithms Decision Tree, SVM, ANN, Bayesian Network and K-Nearest Neighbour. The authors restricted their review of research works from 2003 to 2013. The authors found ANN to be the most widely used predictive model in medical prediction, SVM is mainly used in computational biology such as microarray data analysis, translational initiation site recognition in DNA. The author noted that ANN and SVM are black box models hence, they have low acceptance in community working with large dataset due to their high time complexity in the training phase(Shweta Kharya, 2012). The authors found Bayesian Network to be very suitable to make predictions in uncertain circumstances coupled with incomplete data and also BN is suitable for the classification of breast cancer tumour.

(Agarap, 2017) proposed a new machine learning algorithm called GRU-SVM, this is a combination of gated recurrent unit (GRU) variant of recurrent neural network and (RNN) and the support vector machine (SVM) that is used on WBCD. The same dataset was also used on these algorithms Multilayer perceptron, Nearest Neighbour, Softmax Regression, and SVM.

The author split the dataset into two, 70% was kept for the training phase and 30% for the testing phase. The author's result showed that all the used Machine learning

models performed well (all met above 90% test accuracy) on the classification task. The MLP was outstanding with a test accuracy of approximately 99.04%.

(Ivančaková, Babič, & Butka, 2018) investigated six machine learning models namely, C4.5, SVM, K-NN, Random Forest, Neural Networks, Naïve Bayes on Wisconsin Diagnostic breast cancer Dataset, and the authors found their results to be plausible compared to prior works done before theirs. The authors proposed a new combination of machine learning algorithms such as using K-Means for the recognition of hidden patterns of the malignant and benign tumours separately, and SVM was then used in generating the new classifier within 10-fold cross validation. Their new approach got an accuracy of 97.38%, which was an improvement to the scores of the six algorithms.

(L. Li et al., 2017) in their research work, employed Backward propagation (BP) Neural Network and Logistic regression to classify heart sound signals into normal or abnormal. The authors used 3 feature sets which are formed from 40 extracted features, these 3 features set served as inputs to the models.

The authors found BP Neural network to have better classification performance with an accuracy of 88.56% (sensitivity 68.36%, specificity 94.01%) and Logistic Regression having an accuracy of 72.56% (sensitivity 15.68%, specificity 87.71%).

CHAPTER THREE

MATERIALS AND METHOD

In this chapter, we shall discuss the framework, algorithm used and explain various stages in the framework.

3.1 Concept of Classification Technique

Classification is one of the ways the machine learns. It has the specific goal of accurately classifying the unknown values of attribute of the target known values (Jhajharia *et al*., 2016; Aggarwal & Xhai, 2012; Mitra & Acharya, 2004). Classification is crucial in data mining and machine learning because it presents a clear distinction between the various classes by understanding deeply the relationship between the variables together with the class attribute (Aggarwal, 2015; Guo, Huang, & Zhang, 2014; Kriegel et al., 2007; T. Li, Ma, & Ogihara, 2005; Uppal, 2016).

It is established that there are some attributes that are slightly different from another or the difference is insignificant. Therefore if some of the insignificant attributes are ignored, results will be obtained at the minimum time (Garg, Beg, & Ansari, 2009). We developed a model for classifying breast cancer using Logistic Regression classifier. The model was trained and tested using a Wisconsin Breast Cancer Dataset (WBCD) obtained from UCI machine learning repositories.

3.2 Software Design Phase

The proposed model was implemented using Jupyter Notebook, a python programming environment, which has a machine learning library, Sci-Kit Learn.

Sci-Kit Learn has built-in support for all extant machine learning algorithms used for classification, and a good number of packages for data pre-processing techniques, machine learning performance measures.

This language has major advantages over others because of its flexibility, given output after the convergence of the learning stage, easy plotting of graphs and charts.

3.3 Hardware Requirement

The hardware requirements are:

1. Windows 7, 8 or 10, 64bits for PC and iOS 8, 10 for Macintosh operating system.
2. All CPUs
3. 4GB RAM and 40GB HDD free space

3.4 Proposed Framework

The proposed framework consists of the following modules: data collection, the pre-processing stage which involves the handling of missing data, the training, and testing of the machine learning models and lastly, performance analysis and comparison.

Fig. 3.1 depicts the proposed framework, the data is collected from UCI online machine learning repository. The data collected will be pre-processed, the pre-processing is done so as to handle the missing values in the data and a feature scaling technique is employed to normalize the data. The data is split into training set (80%) and test set (20%). The training is used to train the four prediction models, while the test set is used for validation purpose.

Using these performance metrics, which are precision, recall, and f1-score, the four prediction models are evaluated and compared.

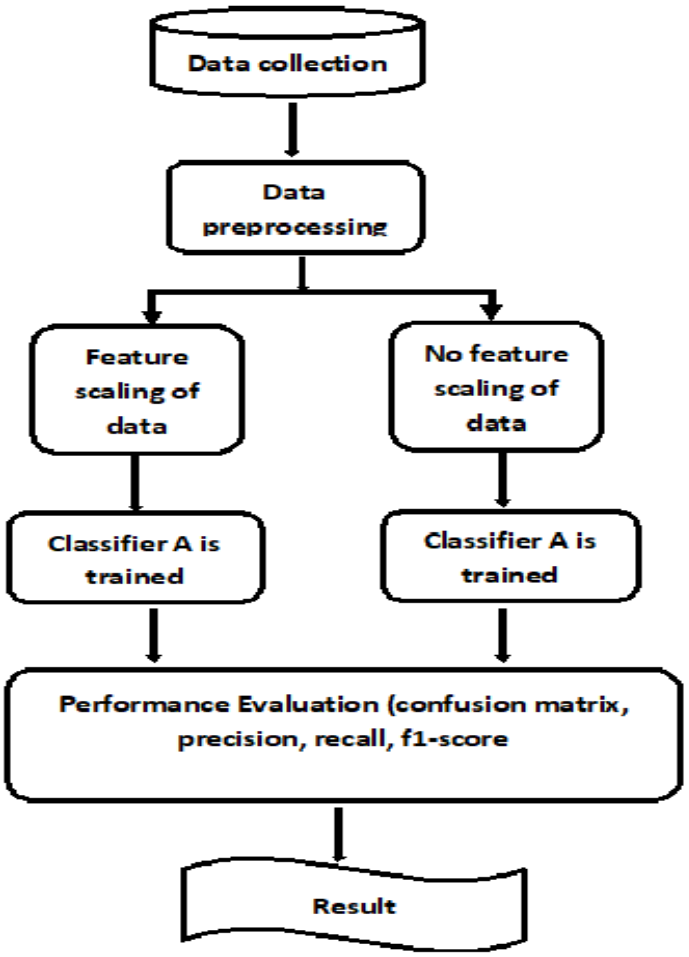


Figure 3. 1. Proposed Classification Framework

3.4.1 Experiments

The first step in the methodology is pre-processing the data, using the tools available in SciKit Learn library available in Python programming language (version 3.6.5). Taking into consideration the dataset adopted, the pre-processing will focus on managing the missing attributes, the unbalanced data and the number of attributes used to train the predicting model.

To handle the 16 missing values, we calculated the mean of the non-missing values and this mean value will be used to replace all missing values for all attributes having one or two missing values in the dataset.

When studying problems with imbalanced data, it is crucial to adjust either the classifier or the training set balance, or even both, to avoid the creation of an inaccurate classifier. A common practice for dealing with imbalanced datasets is to rebalance them artificially, which is called “up-sampling” (replicating features from the minority) and “down-sampling” (removing cased from the majority). There are plenty of studies demonstrating that this kind of technique does not have a great effect on the predictive performance of learned classifiers (Rodrigues, 2016).

In this paper, the problem with imbalanced data is solved by choosing machine learning methods that are insensitive to this kind of issue. This classifier is Logistic Regression.

3.4.2 Data collection

Data set was made available by Wisconsin Madison hospital via the UCI machine learning repository. The dataset is available at:

https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

The breast cancer dataset found at UCI repository was collected by Dr. Williams H. Wolberg (1989 - 1991) (Mangasarian, Street, & Wolberg, 2008). Wisconsin dataset is numerical in nature with 699 rows and 11 columns; each row of the data has a unique ID with nine attributes and one class attribute.

It comprises of 699 samples, 683 are complete data sample while 16 samples are having some missing values. The Wisconsin Breast Cancer Data (WBCD) data were obtained via fine needle aspirates of affected tissue with virtually assessed nuclear features from patients’ breasts.

The program uses a curve-fitting algorithm, as shown in Fig. 3.2, to compute ten features from each one of the cells in the sample, then it calculates the mean value, extreme value and standard error of each feature for the image, returning a 30 real-valuated vector (Rodrigues, 2016).

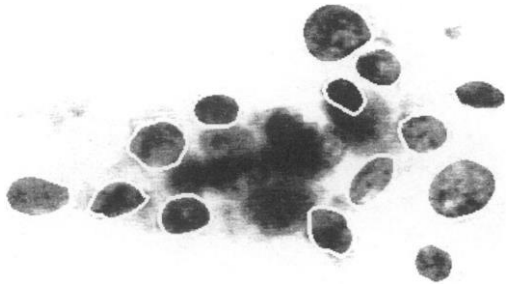


Figure 3. 2. A magnified image of a malignant breast fine needle aspirate (Mangasarian et al., 2008).

Each attribute is scaled on 1-10 which implies 10 as the most abnormal. The attribute class is represented by 2 or 4, *benign* and *malignant* respectively. Fig. 3.3 shows the dataset, and Table 3.1 shows the detailed dataset description of the attributes.

The data can be considered ‘noise-free‘ and has 16 missing values, which are the Bare Nuclei for 16 different instances.

Out[5]:

	Sample_code_number	Clump_Thickness	Uniformity_of_Cell_Size	Uniformity_of_Cell_Shape	Marginal_Adhesion	Single_Epithelial_Cell_Size	Bare_Nuclei	Bla
0	1000025	5	1	1	1	2	1.0	
1	1002945	5	4	4	5	7	10.0	
2	1015425	3	1	1	1	2	2.0	
3	1016277	6	8	8	1	3	4.0	
4	1017023	4	1	1	3	2	1.0	
5	1017122	8	10	10	8	7	10.0	
6	1018099	1	1	1	1	2	10.0	
7	1018561	2	1	2	1	2	1.0	
8	1033078	2	1	1	1	2	1.0	
9	1033078	4	2	1	1	2	1.0	
10	1035283	1	1	1	1	1	1.0	
11	1036172	2	1	1	1	2	1.0	
12	1041801	5	3	3	3	2	3.0	
13	1043999	1	1	1	1	2	3.0	
14	1044572	8	7	5	10	7	9.0	
15	1047630	7	4	6	4	6	1.0	
16	1048672	4	1	1	1	2	1.0	
17	1049815	4	1	1	1	2	1.0	
18	1050670	10	7	7	6	4	10.0	
19	1050718	6	1	1	1	2	1.0	

Figure 3. 3. Sample of the Wisconsin breast cancer dataset

Table 3. 1. Summary of the dataset

Dataset	No. of Attributes	No. of Instances	No. of Classes
WBCD	11	699	2
Features	Clump thickness	Numeric	1-10
	Uniformity of cell size	Numeric	1-10
	Uniformity of Cell Shape	Numeric	1-10
	Marginal Adhesion	Numeric	1-10
	Single Epithelial Cell Size	Numeric	1-10
	Bare Nuclei	Numeric	1-10
	Bland Chromatin	Numeric	1-10
	Normal Nucleoli	Numeric	1-10
	Mitoses	Nominal	1-10
	Class	Nominal	2 (Benign) or 4 (Malignant)
Class Distribution		Benign: 458 (65.5%)	
		Malignant: 241 (34.5%)	
Number of Missing Values		16	
Number of Instances		699	

Features of data set

- Clump Thickness: The nature of benign is monolayer grouping while cancerous is in multilayer.
- Uniformity of Cell Size/Shape: Cancer cells vary in shapes and sizes.
- Marginal Adhesion: The nature of normal is that they stick together while cancerous cell lose.
- Single Epithelial Size: Cells are significantly enlarged.
- Bare Nuclei: This term is usually in reference to nuclei that are not surrounded cytoplasm.
- Bland Chromatin: In cancer, the chromatin tends to be coarser.

- Normal Nucleoli: In a normal being, the nucleolus is very small if seen. These are small structure see in the nucleus.
- Mitoses: Cancer is generally known with the uncontrollable cell division.
- Diagnosis: 2 – *Benign* or 4 - *Malignant*

3.4.2 Data pre-processing

The data pre-processing technique employed in this work was done to handle the 16 missing values found in the 'Bare Nuclei' attribute of the data. To handle this problem, the mean of the non-missing values was calculated and this calculated mean is then used to fill up the 16 missing values. In this preprocessing stage, we also employed a feature scaling technique to normalize the data set.

Feature scaling

This method is widely used for normalization in many machine learning algorithms (e.g., support vector machines, logistic regression, and artificial neural networks) (Grus, 2015). The general method of calculation is to determine the distribution mean and standard deviation for each feature. Next, we subtract the mean from each feature. Then we divide the values (mean is already subtracted) of each feature by its standard deviation.

$$\text{Where} \quad x' = \frac{x - \bar{x}}{\sigma}$$

Where x is the original feature vector, $\bar{x} = average\ (x)$ is the mean of that feature vector, and σ is its standard deviation.

3.4.3 Machine learning classifiers

Logistic regression is the main machine learning model that is employed in this research work and for comparison purpose, three other extant machine learning models namely support vector machine, Naïve Bayes, and Artificial Neural Network are considered.

Logistic regression model

Logistic regression was developed in late the 1960s and early 1970s(Cabrera, 2007; Haigh, Cox, & Snell, 2007; Peng, Lee, & Ingersoll, 2002) and became popular among researches in various fields, particularly among health researchers(Abedin, Chowdhury, & Afzal, 2016).

Logistic regression is prevalent in almost every standard statistical software package. Owing to its wide popularity and usefulness in research it is important to comprehend the basics of logistic regression i.e., how does the model operate, what postulations are needed to be verified, how to report the results found, etc. (Abedin et al., 2016). In this paper, we study binomial logistic regression for the classification of breast cancer dataset.

The mathematical notion of logistic regression is to show the relationship between the outcome variable (dependent variable) and predictor variables (independent variables) in terms of logit: the natural logarithm of odds. Let’s take into consideration a simple case where Y is a dichotomous dependent variable categorized as “1” and “0” and X is a continuous independent variable. Now if we draw a scatter plot, as expected we will have two parallel lines analogous to each dependent variable category.

The relationship does not follow a linear trend and hence not possible to describe through a simple linear regression(Abedin et al., 2016; Peng et al., 2002).

Logistic regression remedies this little hiccup by logit transformation on the dependent variable Y. The simplest form of logistic regression model can be written as:

$$logit(Y) = \text{In}\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X \quad (1)$$

Here π is the probability of occurrence for the outcome Y and π/ (1−π) is the odds of success; the ratio of the probability of occurrence for the outcome Y and the probability of the outcome Y not occurring.

β_0 and β_1 are known as intercept and slope (regression coefficient) respectively (McGee, 2013).

By taking antilog on both sides of equation (1) we can estimate the probability of the occurrence of outcome Y for a given value of predictor X (Abedin et al., 2016):

$$\pi = P(Y|X = x) = \frac{e^{\beta_0+\beta_1}}{1+e^{\beta_0+\beta_1}} \quad (2)$$

The predictor variable X can be either continuous or categorical. Logistic regression can be extended for more than one predictor as well,

$$Logit(Y) = In\frac{\pi}{1-\pi} = \beta_0 + \beta_1 + \cdots + \beta_p X_p \quad (3)$$

Equation (3) is the general form of a logistic regression model for p number of predictors. Regression parameter βs (betas) can be estimated by either maximum likelihood (ML) method or weighted least square method (Abedin et al., 2016). The value of regression coefficients β1...βp points out the correspondence between X's and logit of Y. A coefficient value greater than 0 points to an increase in logit of Y with an increase in X and coefficient value lesser than 0 points to a decrease in logit of Y with an increase in X. When the coefficient value is 0, it indicates there is no linear relationship among logit of Y and predictors X (Fig. 3.4).

For the ease of interpretation, we usually report the odds ratio along with the regression coefficient. Odds ratio can be calculated by the following formula,

$$Odd\ ratio(OR) = e^{\beta} \qquad (4)$$

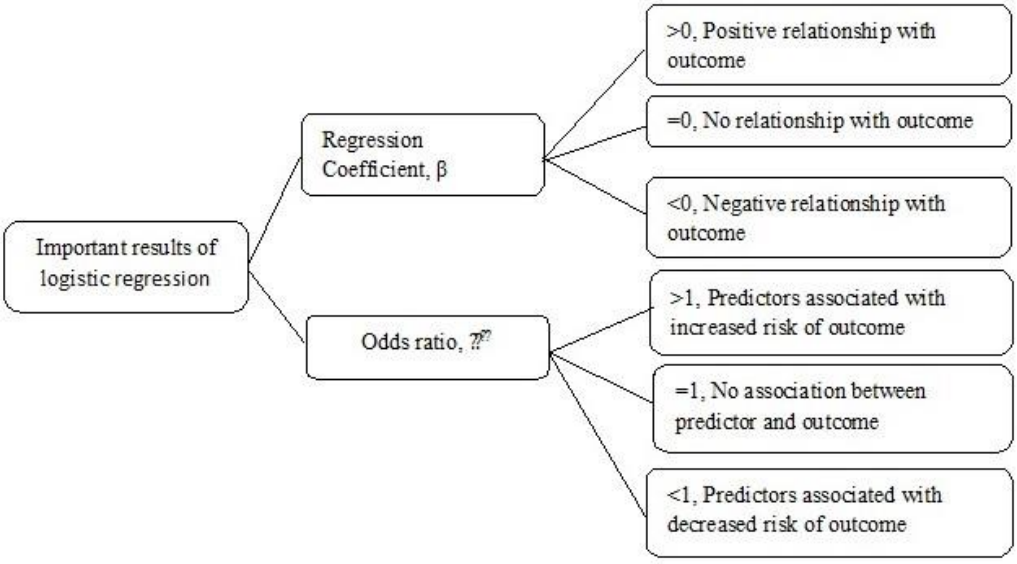


Figure 3. 4: Important estimates of Logistic regression and interpretation

Support Vector Machine Model

SVM is a supervised ML classification technique that is typically employed in the field of cancer diagnosis and prognosis. SVM operates by selecting critical samples from all classes known as support vectors and separating the classes by generating a linear function that divides them as broadly as possible using these support vectors (Bazazeh & Shubair, 2016).

Therefore, it can be said that a mapping between an input vector to a high dimensionality space is made using SVM that aims to find the most suitable hyperplane that divides the data set into classes (Williams & Williams, 2011). This linear classifier aims to maximize the distance between the decision hyperplane and

the nearest data point, which is called the marginal distance, by finding the best-suited hyperplane (Kourou, Exarchos, Exarchos, Karamouzis, & Fotiadis, 2015). Fig. 3.5 shows a scatter plot of two classes with two properties.

A linear hyperplane is defined as $ax_1 + bx_2$ and the aim is to find a, b, and c such that $ax_1 + bx_2 \leq c$ for class 1 and that $ax_1 + bx_2 > c$ for class 2 (Cleophas & Zwinderman, 2013; Williams & Williams, 2011). SVM depends on the support vectors, which are the data sets closest to the decision boundary, in their algorithms and this makes SVM different from other techniques. This is because removing other data points that are further away from the decision hyperplane will not change the boundary as much as if the support vectors were removed (Bazazeh & Shubair, 2016).

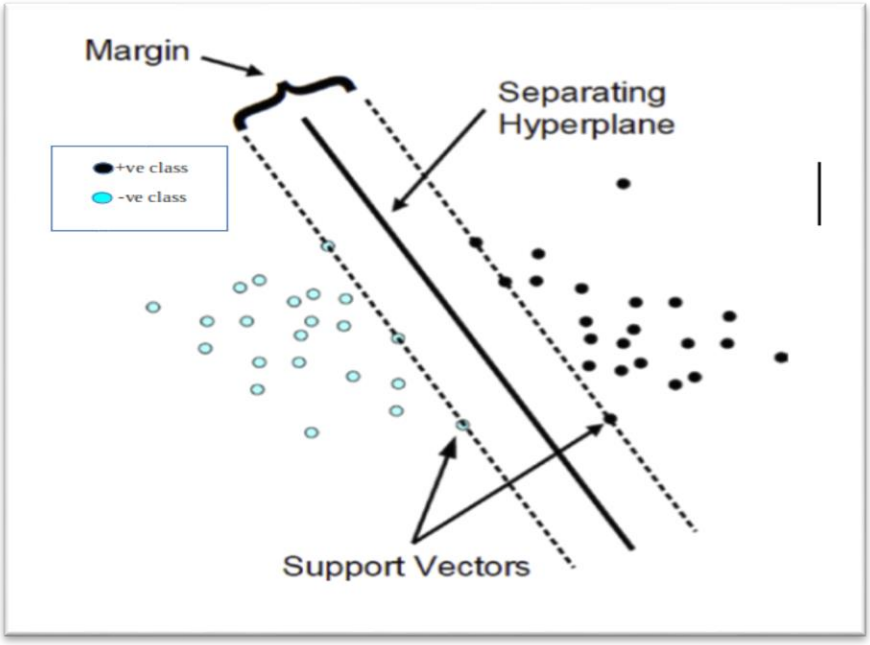


Figure 3. 5. SVM generated hyper-planes

Naïve Bayes Model

Naive Bayes relies on an assumption that is rarely valid in practical learning problems: that the attributes used for deriving a prediction are independent of each other, given the predicted value. By using this technique, the probability of an instance which belongs to a particular class is predicted.

All the features are presumed independent according to Bayes theorem which means there is no dependency among the attribute value on a given class and the other attributes (Frank, Trigg, Holmes, & Witten, 2000).

Multilayer Perceptron model

The MLP is based on the supervised procedure as shown in Fig. 3.6, that is, the network builds a model based on examples in data with known outputs. A relation between problem and solution may be quite general, for example, the simulation of species richness or the abundance of animal (output) expressed by the quality of habitat (input) (Marwala, 2018).

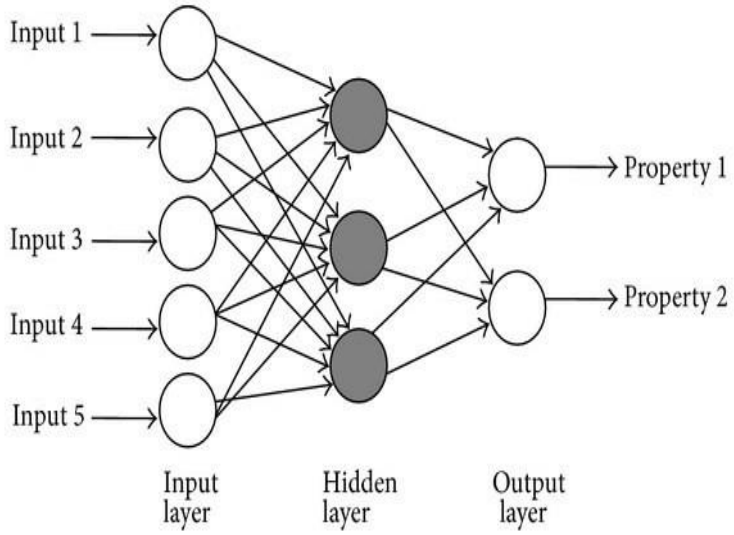


Figure 3. 6. Schematic of a three-layered feedforward neural network, with one input layer, one hidden layer, and one output layer (Marwala, 2018).

3.4.4 Training the logistic regression model

All experiments on the classifiers discussed in this paper were conducted using a machine learning library, SciKit Learn and we used Jupyter Notebook as our integrated development environment (IDE).

SciKit-Learn that is a built-in library in Python 3.6.5, that contains a collection of machine learning algorithms for data pre-processing, classification, regression, clustering and association rules.

Machine learning techniques implemented in SciKit-Learn are applied to a variety of real-world problems. The program offers a well-defined framework for experimenters and developers to build and evaluate their models.

In Binary Logistic Regression analysis methods the independent variables are dummy variables, and these independent variables consist of different size levels whereas dependent variables must be linear and fulfil the response that is needed for this method. A logistic regression model is the result of non-linear transformation of the linear regression model (Yusuff et al., 2012).

For a binary classification problem, where $Y \in \{0, 1\}$.

An Odds Ratio (OR) is defined as: **OR** = $\frac{(P(Y = 1|X)}{P(Y=0|X))} \in [0, \infty\}$

If **OR** > 1, P(Y=1|X) is more likely to occur

If **OR** < 1, P(Y=0|X) is more likely to occur

A logit is defined as, logit = In (OR) $\in (-\infty, \infty)$

Output = 0 or 1; Hypothesis => Z = WX + B

Activation function: hΘ(x) = sigmoid (Z)

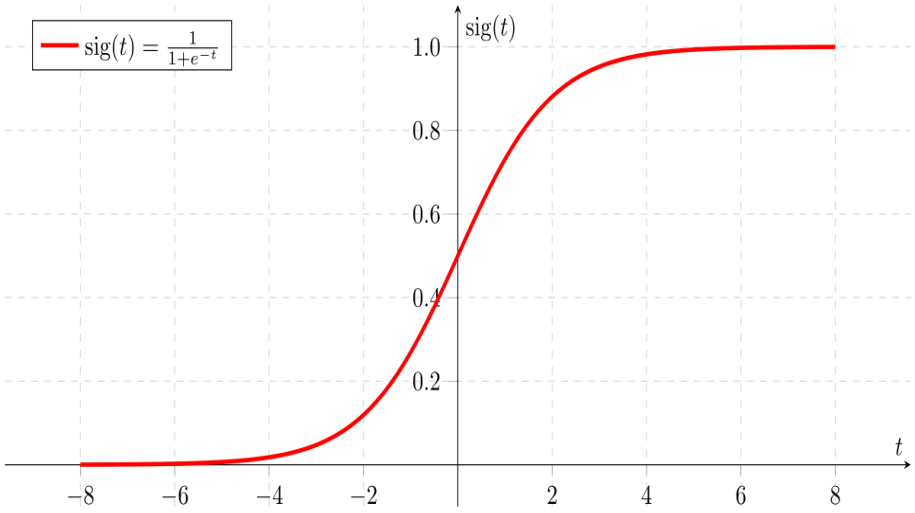


Figure 3.7. Sigmoid Activation Function

From Fig. 3.7, we can depict that if 'Z' goes to infinity, Y (predicted) will become 1 and if 'Z' goes to negative infinity, Y (predicted) will become 0. However, in this research work, 1 will be 2 and 0 will be 4 for our Y (predicted outcomes).

Analysis of the hypothesis

The output from the hypothesis is the estimated probability. This is used to infer how confident can predicted value be actual value when given an input X. Consider the example below.

X = [X0 X1 … X9] = [One row from our dataset, having the 9 attributes]

Based on the X value, assume we obtained the estimated probability to be 0.8. This denotes that there is an 80% chance that a breast cancer case is benign and in the event of a probability 0.45, then there is a 65% chance that a case is malignant.

The output from the hypothesis is the estimated probability. This is used to infer how confident can predicted value be actual value when given an input X.

Mathematically this can be written as shown in Fig. 3.8 below,

$h_\Theta(x) = P(Y=1 \mid X; theta)$

Probability that Y=1 given X which is parameterized by 'theta'.

$P(Y=1 \mid X; theta) + P(Y=0 \mid X; theta) = 1$

$P(Y=0 \mid X; theta) = 1 - P(Y=1 \mid X; theta)$

Figure 3. 8. Mathematical Representation of Logistic regression model

This justifies the name 'logistic regression'. Data is fitted into the linear regression model, which then be acted upon by a logistic function predicting the target categorical dependent variable.

3.5 Classifier Performance Evaluation Criteria

After the training phase, the classifiers are tested and their prediction accuracies are measured. To effectively evaluate the performance of these prediction models the following performance metrics are used in this work to achieve this aim.

3.5.1 Confusion matrix

A confusion matrix contains information about actual and predicted classifications done by a classification model. Performance of such model is commonly evaluated using the data in the matrix.

Table 3.2 shows the confusion matrix for a two class classifier (Goyal & Mehta, 2012). It classifies each instance into one of two classes.

The classes are true and false; this gives rise to four possible classifications for each instance as listed below.

- True-Positive(TP) means positive pattern seen as positives
- False-Positive(FP) means negative pattern seen as positive
- False-Negative(FN) means positive Pattern seen as negative

- True-Negative(TN) means negative Pattern seen as negative

Table 3. 2. A 2 × 2 Confusion Matrix for two Class Classifier

Confusion Matrix		**Actual Class**	
		Positive(2)	Negative(4)
Predicted class	Positive(2)	True Positive (TP)	False Negative (FN)
	Negative(4)	False Positive (FP)	True Negative (TN)

From the table above the classification that lies along the major diagonal that is TP and TN are the correct classifications/predictions.

While the remaining fields, FN and FP signify model error. From Confusion Matrix many model performance metrics can be derived, popular among the metrics is accuracy, which is defined as:

$$\boldsymbol{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Here accuracy rate is all the correctly classified patterns divided by total number of patterns.

Other performance metrics include precision, recall, and f1-score defined as follows:

3.5.2 Precision

This defines how exact the model is in terms of its prediction

$precision = \frac{TP}{TP+FP}$ For the benign case;

$precision = \frac{TN}{TN+FN}$ For the malignant case;

3.5.3 Recall (Sensitivity)

This performance metric implies how different values and independent variable affect a dependent variable.

$$recall = \frac{TP}{TP+FN}$$ For the benign case;

$$recall = \frac{TN}{TN+FP}$$ For the malignant case;

3.5.4 F1-Score

This conveys the balance between precision and recall; this is the harmonic mean of Precision and Recall.

$$\textbf{F1-score} = \frac{2}{\frac{1}{Precision}+\frac{1}{Recall}}$$

CHAPTER FOUR

RESULTS AND DISCUSSIONS

This chapter includes the implemented framework and results with the programming language used from the preprocessing phase to the training and validation phase of the prediction models. Screenshots of results are presented to support our proposed framework.

4.1 Presentation of Results

All the steps taken in this research work: handling missing values in the data, feature scaling of the data, training the prediction models and the evaluation of the models' performance in terms of accuracy, precision and sensitivity are presented; all the various stages of preprocessing, normalization, training and testing of data, classification, and measures of accuracy are implemented using SciKit Learn library in python programming. The results are all displayed and analyzed.

4.1.1 Reading the Textual File

The downloaded data from UCI machine learning repository is located on my local machine at this directory "C:\Users\SKITTISH\Desktop\THESIS WORK\code" with the file name WBCD_9_attributes.csv. Python programming has a library known as Pandas, which can be used to open and read comma separated valued (CSV) files and Fig. 4.1 shows the python code used to read in our data set file.

```
In [1]: import pandas as pd
        import numpy as np
        from matplotlib import pyplot as plt
        %matplotlib inline

In [2]: df = pd.read_csv("WBCD_9_attributes.csv")
```

Figure 4. 1. Reading in the WBCD file.

4.1.2 Data pre-processing

From Fig. 4.2, we can see that for 'Bare_Nuclei' there are 683 data points, whereas others have 699 data points. There are 16 missing values in Bare_Nuclei, and without handling these missing values, it will be difficult to train the classifiers.

```
In [4]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 699 entries, 0 to 698
Data columns (total 11 columns):
Sample_code_number            699 non-null int64
Clump_Thickness               699 non-null int64
Uniformity_of_Cell_Size       699 non-null int64
Uniformity_of_Cell_Shape      699 non-null int64
Marginal_Adhesion             699 non-null int64
Single_Epithelial_Cell_Size   699 non-null int64
Bare_Nuclei                   683 non-null float64
Bland_Chromatin               699 non-null int64
Normal_Nucleoli               699 non-null int64
Mitoses                       699 non-null int64
Class                         699 non-null int64
dtypes: float64(1), int64(10)
memory usage: 60.1 KB
```

Figure 4. 2. WBCD information.

Handling missing value

To handle the missing, we adopted a strategy whereby we use the calculated mean from the 683 non-missing values in the Bare_Nuclei to fill up the 16 missing values. To implement this, we used an imputer method in SciKit Learn to handle the missing value situation and Fig. 4.3 below shows the python code used to handle the missing values.

```
In [ ]: from sklearn.preprocessing import Imputer

In [ ]: imputer = Imputer(missing_values='NaN',strategy='mean',axis=0)

In [ ]: df.iloc[:,[6]]= imputer.fit_transform(df.iloc[:,[6]])
```

Figure 4. 3. Handling the missing values

After running the code as shown in Fig. 4.3, all the missing values are filled up. Fig. 4.4 below shows the information of our dataset after the missing values have been filled up. From Fig. 4.4 we can see that Bare_Nuclei now has 699 data points.

```
In [13]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 699 entries, 0 to 698
Data columns (total 11 columns):
Sample_code_number            699 non-null int64
Clump_Thickness               699 non-null int64
Uniformity_of_Cell_Size       699 non-null int64
Uniformity_of_Cell_Shape      699 non-null int64
Marginal_Adhesion             699 non-null int64
Single_Epithelial_Cell_Size   699 non-null int64
Bare_Nuclei                   699 non-null int32
Bland_Chromatin               699 non-null int64
Normal_Nucleoli               699 non-null int64
Mitoses                       699 non-null int64
Class                         699 non-null int64
dtypes: int32(1), int64(10)
memory usage: 57.4 KB
```

Figure 4. 4. WBCD information with no missing values

Feature scaling

Fig. 4.5 shows the python code used to implement the feature scaling of the dataset using a StandardScalar method in the preprocessing class of SciKit Learn library.

```
In [11]: from sklearn.preprocessing import StandardScaler
         scaler = StandardScaler()
         scaler.fit(x_train)

Out[11]: StandardScaler(copy=True, with_mean=True, with_std=True)

In [13]: x_train = scaler.transform(x_train)

In [14]: x_test = scaler.transform(x_test)
```

Figure 4. 5. Feature scaling of WBCD

4.1.3 Training of classifiers and classification task

For classification, the preprocessed data is fed to the logistic regression model. The data is split into two parts; the training set (80% of data) and test set (20% of data). To train the model we used the training set and to evaluate the performance of the model we used the test set. Fig. 4.6 shows the python programming code used to implement this. The training set has 569 data points and the test set is 140.

```
In [ ]: x = df.iloc[:,1:10]
        y = df.iloc[:,10:11]

In [ ]:

In [ ]: bc_x = bc.iloc[:,2:32]
        bc_y = bc.diagnosis

In [ ]: x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)
        x_train2,x_test2,y_train2,y_test2 = train_test_split(bc_x,bc_y,test_size=0.2)
```

Figure 4. 6. WBCD split into training and test set.

Fig. 4.7 depicts the python code used to implement the training of the Logistic Regression model.

```
In [37]: lr2 = LogisticRegression()

In [38]: lr2.fit(x_train2,y_train2)

Out[38]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                            intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                            penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                            verbose=0, warm_start=False)
```

Figure 4. 8. Training LR model.

Fig. 4.8 depicts the python code used to implement the training of the Naïve Bayes model.

```
In [13]: gnb = GaussianNB()

In [14]: gnb.fit(x_train,y_train)

Out[14]: GaussianNB(priors=None)
```

Figure 4. 9. Training the NB model

Fig. 4.9 depicts the python code used to implement the training of the Support Vector Machine model.

```
In [20]: svm_clf = LinearSVC(C=1,loss='hinge')

In [21]: svm_clf.fit(x_train,y_train)

Out[21]: LinearSVC(C=1, class_weight=None, dual=True, fit_intercept=True,
              intercept_scaling=1, loss='hinge', max_iter=1000, multi_class
         ='ovr',
              penalty='l2', random_state=None, tol=0.0001, verbose=0)
```

Figure 4. 10. Training the SVM model

Fig. 4.10 depicts the python code used to implement the training of the Multilayer Perceptron Model.

```
In [15]: nn_clf = MLPClassifier( solver='adam', alpha=1e-5,max_iter=10000,
                                 hidden_layer_sizes=(5,2), random_state=1)

In [16]: nn_clf.fit(x_train,y_train)

Out[16]: MLPClassifier(activation='relu', alpha=1e-05, batch_size='auto', beta_1=0.9,
                       beta_2=0.999, early_stopping=False, epsilon=1e-08,
                       hidden_layer_sizes=(5, 2), learning_rate='constant',
                       learning_rate_init=0.001, max_iter=10000, momentum=0.9,
                       nesterovs_momentum=True, power_t=0.5, random_state=1, shuffle=True,
                       solver='adam', tol=0.0001, validation_fraction=0.1, verbose=False,
                       warm_start=False)
```

Figure 4. 11. Training the MLP model

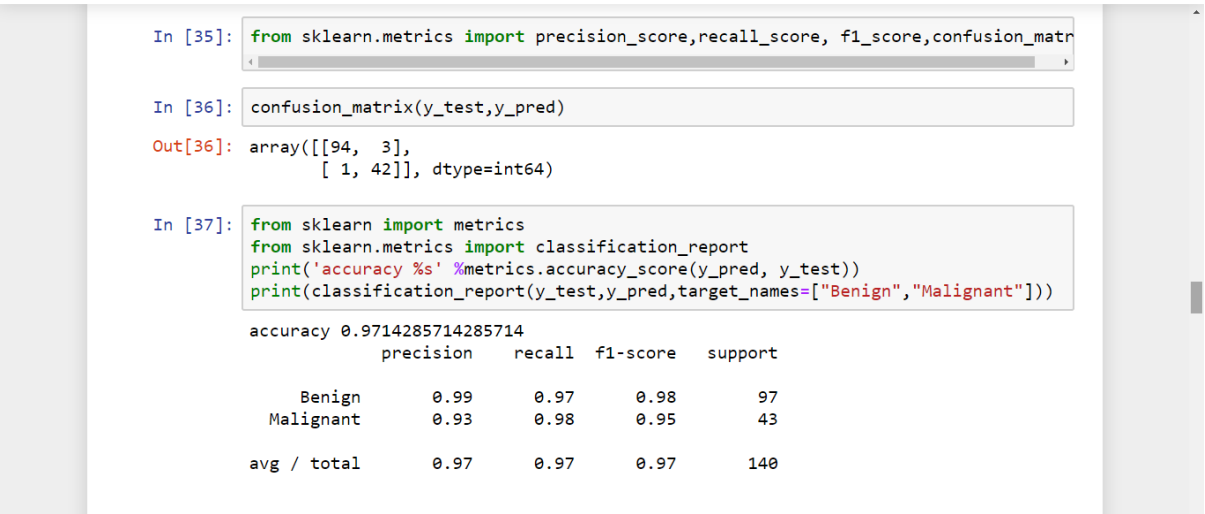
After the learning and training phase, the next step is to test the intelligence of the model, for this purpose the test data is used. The test set has 140 data points with 9 independent features and one target label. To test the trained model, the test set with the exclusion of the target label is fed to the model for the model to make some predictions. The predictions (predicted outcome) from the model will be used to match the actual outcomes of the test set. Fig. 4.11 shows the python code used to test the logistic regression model and the predicted outcomes are shown.

```
In [28]: y_pred = lr.predict(x_test)

In [29]: y_pred

Out[29]: array([2, 2, 2, 4, 2, 2, 2, 2, 4, 2, 2, 2, 2, 4, 4, 2, 2, 2, 2, 4, 2, 2,
                2, 2, 4, 4, 4, 2, 2, 2, 2, 2, 4, 4, 4, 2, 2, 4, 2, 4, 4, 4, 2, 2,
                2, 2, 4, 2, 2, 2, 2, 2, 2, 2, 4, 2, 2, 2, 4, 2, 2, 4, 2, 4, 2, 4,
                2, 2, 4, 4, 4, 4, 2, 2, 2, 2, 2, 4, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2,
                2, 4, 4, 2, 2, 2, 2, 4, 2, 2, 4, 4, 2, 4, 2, 2, 2, 2, 4, 2, 2, 2,
                2, 2, 2, 4, 4, 2, 2, 2, 2, 4, 4, 2, 2, 2, 4, 4, 2, 2, 4, 4, 2, 2,
                4, 2, 4, 2, 2, 2, 4, 2], dtype=int64)
```

Figure 4. 12. LR model is tested

4.1.4 Performance Analysis

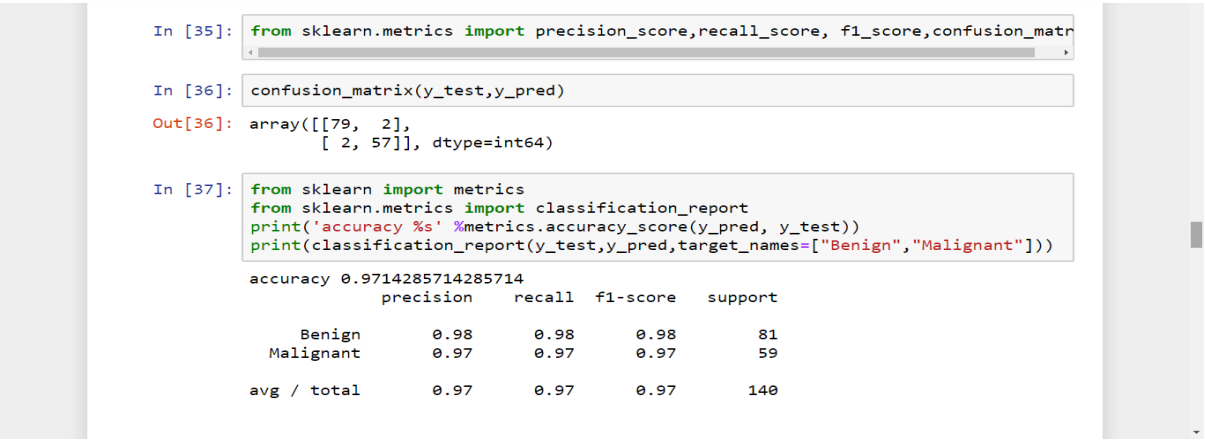
Four performance metrics namely confusion matrix, precision, recall, and f1-score are used to evaluate the performance of the trained models. Then, their performances are discussed, analyzed and hypotheses are made. Fig. 4.12 shows confusion matrix, accuracy score, precision score, recall score and f1-score for the LR model when feature scaling is not applied on the WBCD.



```
In [35]: from sklearn.metrics import precision_score,recall_score, f1_score,confusion_matr

In [36]: confusion_matrix(y_test,y_pred)
Out[36]: array([[94,  3],
                [ 1, 42]], dtype=int64)

In [37]: from sklearn import metrics
         from sklearn.metrics import classification_report
         print('accuracy %s' %metrics.accuracy_score(y_pred, y_test))
         print(classification_report(y_test,y_pred,target_names=["Benign","Malignant"]))

accuracy 0.9714285714285714
             precision    recall  f1-score   support

     Benign       0.99      0.97      0.98        97
  Malignant       0.93      0.98      0.95        43

avg / total       0.97      0.97      0.97       140
```

Figure 4. 13. Performance metrics for the LR model (1)

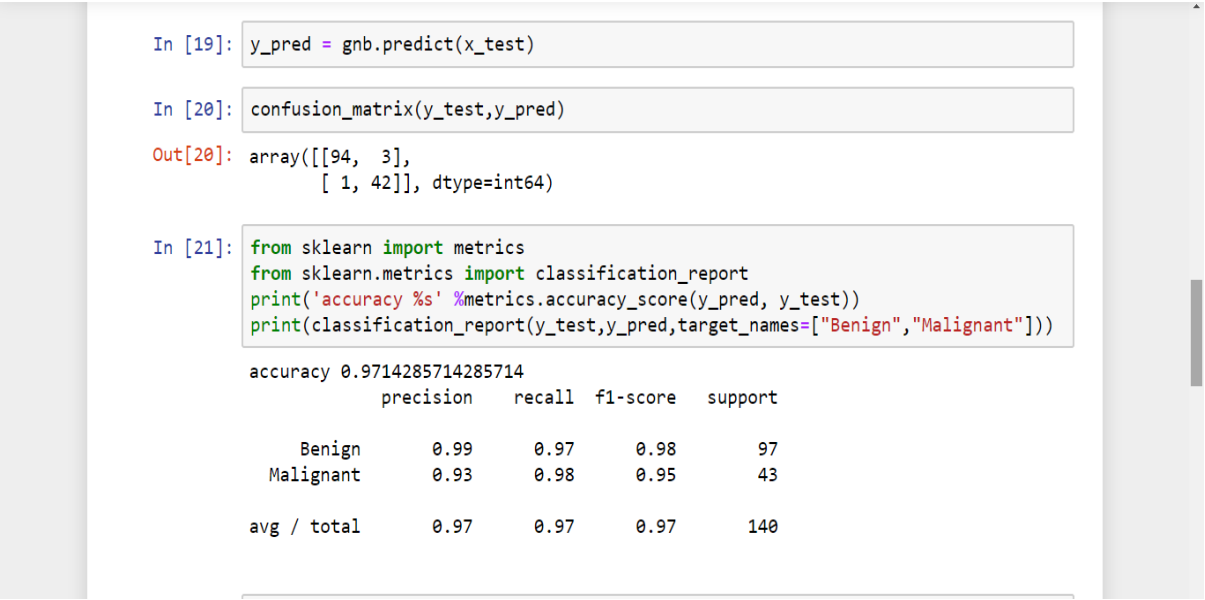
Fig. 4.13 shows confusion matrix, accuracy score, precision score, recall score and f1-score for the LR model when feature scaling is applied on the WBCD.



```
In [35]: from sklearn.metrics import precision_score,recall_score, f1_score,confusion_matr

In [36]: confusion_matrix(y_test,y_pred)
Out[36]: array([[79,  2],
                [ 2, 57]], dtype=int64)

In [37]: from sklearn import metrics
         from sklearn.metrics import classification_report
         print('accuracy %s' %metrics.accuracy_score(y_pred, y_test))
         print(classification_report(y_test,y_pred,target_names=["Benign","Malignant"]))

accuracy 0.9714285714285714
             precision    recall  f1-score   support

     Benign       0.98      0.98      0.98        81
  Malignant       0.97      0.97      0.97        59

avg / total       0.97      0.97      0.97       140
```

Figure 4. 14. Performance metrics for the LR model (2)

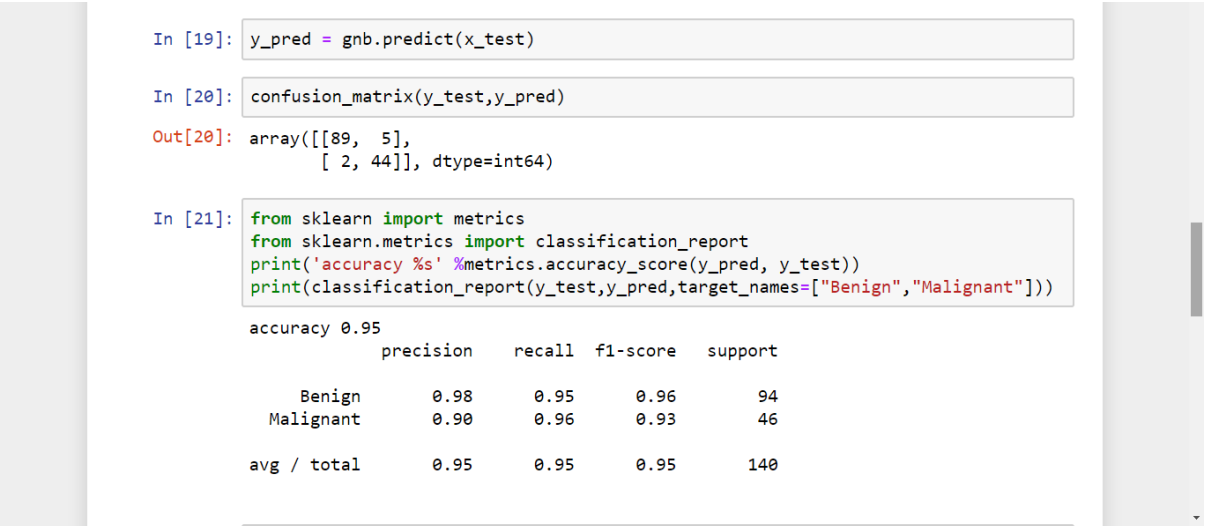
Fig. 4.14 shows confusion matrix, accuracy score, precision score, recall score and f1-score for the NB model when feature scaling is not applied on the WBCD.



```
In [19]: y_pred = gnb.predict(x_test)

In [20]: confusion_matrix(y_test,y_pred)

Out[20]: array([[94,  3],
                [ 1, 42]], dtype=int64)

In [21]: from sklearn import metrics
         from sklearn.metrics import classification_report
         print('accuracy %s' %metrics.accuracy_score(y_pred, y_test))
         print(classification_report(y_test,y_pred,target_names=["Benign","Malignant"]))

accuracy 0.9714285714285714
             precision    recall  f1-score   support

     Benign       0.99      0.97      0.98        97
  Malignant       0.93      0.98      0.95        43

avg / total       0.97      0.97      0.97       140
```

Figure 4. 15. Performance metrics for the NB model (1)

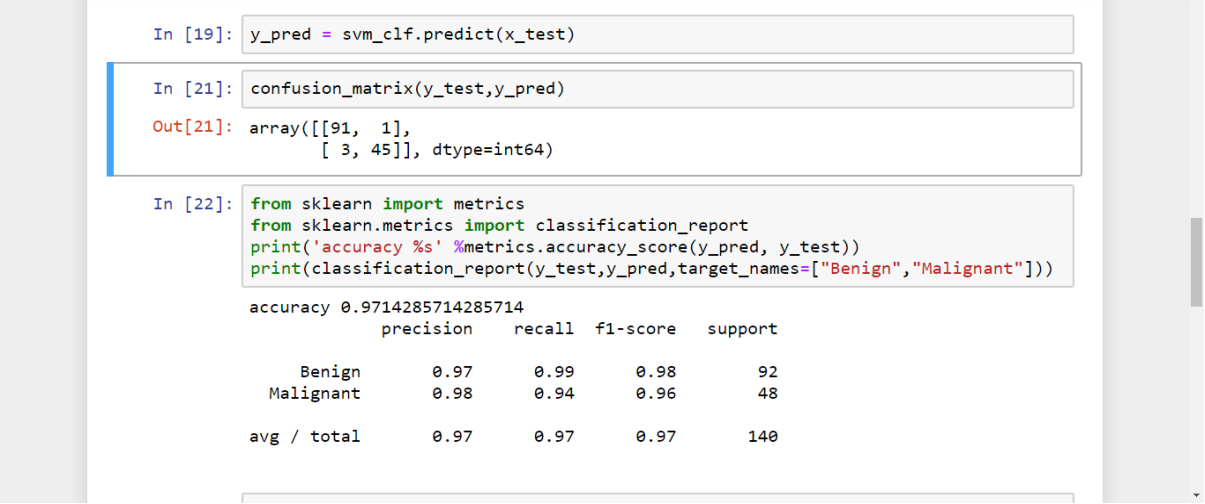
Fig. 4.15 shows confusion matrix, accuracy score, precision score, recall score and f1-score for the NB model when feature scaling is applied on the WBCD.



```
In [19]: y_pred = gnb.predict(x_test)

In [20]: confusion_matrix(y_test,y_pred)

Out[20]: array([[89,  5],
                [ 2, 44]], dtype=int64)

In [21]: from sklearn import metrics
         from sklearn.metrics import classification_report
         print('accuracy %s' %metrics.accuracy_score(y_pred, y_test))
         print(classification_report(y_test,y_pred,target_names=["Benign","Malignant"]))

accuracy 0.95
             precision    recall  f1-score   support

     Benign       0.98      0.95      0.96        94
  Malignant       0.90      0.96      0.93        46

avg / total       0.95      0.95      0.95       140
```

Figure 4. 16. Performance metrics for the NB model (2)

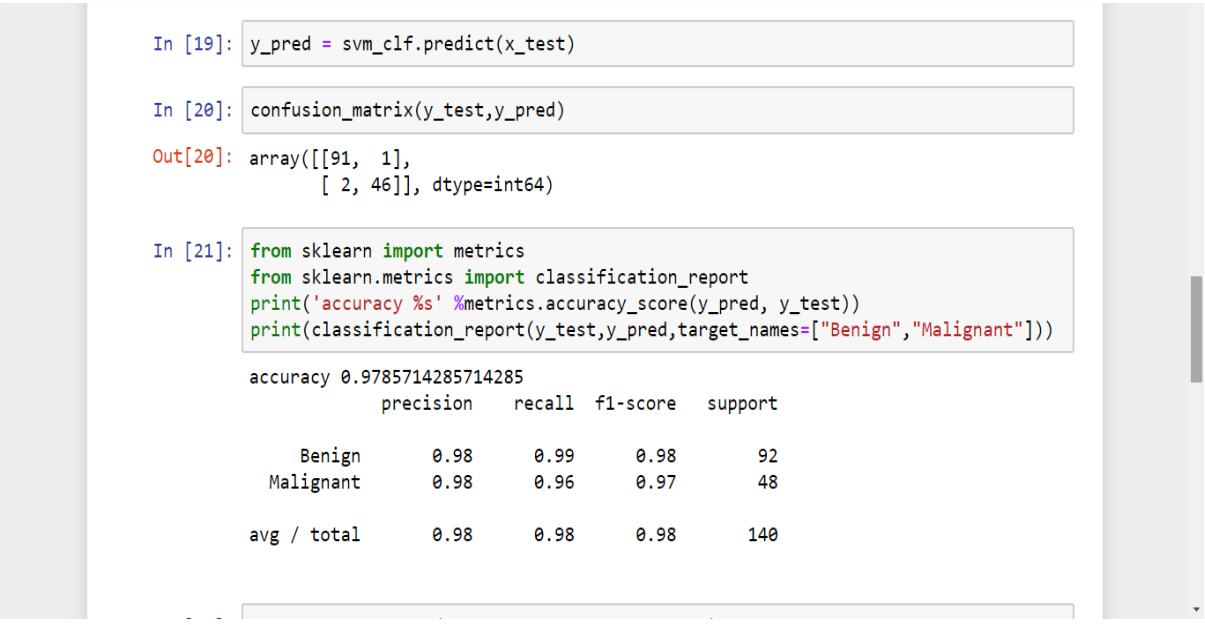
Fig. 4.16 shows confusion matrix, accuracy score, precision score, recall score and f1-score for the SVM model when feature scaling is not applied on the WBCD.



```
In [19]: y_pred = svm_clf.predict(x_test)

In [21]: confusion_matrix(y_test,y_pred)

Out[21]: array([[91,  1],
                [ 3, 45]], dtype=int64)

In [22]: from sklearn import metrics
         from sklearn.metrics import classification_report
         print('accuracy %s' %metrics.accuracy_score(y_pred, y_test))
         print(classification_report(y_test,y_pred,target_names=["Benign","Malignant"]))

accuracy 0.9714285714285714
             precision    recall  f1-score   support

     Benign       0.97      0.99      0.98        92
  Malignant       0.98      0.94      0.96        48

avg / total       0.97      0.97      0.97       140
```

Figure 4. 17. Performance metrics for the SVM model (1)

Fig. 4.17 shows confusion matrix, accuracy score, precision score, recall score and f1-score for the SVM model when feature scaling is applied on the WBCD.



```
In [19]: y_pred = svm_clf.predict(x_test)

In [20]: confusion_matrix(y_test,y_pred)

Out[20]: array([[91,  1],
                [ 2, 46]], dtype=int64)

In [21]: from sklearn import metrics
         from sklearn.metrics import classification_report
         print('accuracy %s' %metrics.accuracy_score(y_pred, y_test))
         print(classification_report(y_test,y_pred,target_names=["Benign","Malignant"]))

accuracy 0.9785714285714285
             precision    recall  f1-score   support

     Benign       0.98      0.99      0.98        92
  Malignant       0.98      0.96      0.97        48

avg / total       0.98      0.98      0.98       140
```

Figure 4. 18. Performance metrics for the SVM model (2)

Fig. 4.18 shows confusion matrix, accuracy score, precision score, recall score and f1-score for the MLP model when feature scaling is not applied on the WBCD.

```
In [22]: y_pred = nn_clf.predict(x_test)

In [23]: confusion_matrix(y_test,y_pred)

Out[23]: array([[92,  0],
                [48,  0]], dtype=int64)

In [24]: from sklearn import metrics
         from sklearn.metrics import classification_report
         print('accuracy %s' %metrics.accuracy_score(y_pred, y_test))
         print(classification_report(y_test,y_pred,target_names=["Benign","Malignant"]))

accuracy 0.6571428571428571
             precision    recall  f1-score   support

     Benign       0.66      1.00      0.79        92
  Malignant       0.00      0.00      0.00        48

avg / total       0.43      0.66      0.52       140
```

Figure 4. 19. Performance metrics for the MLP model (1)

Fig. 4.19 shows confusion matrix, accuracy score, precision score, recall score and f1-score for the MLP model when feature scaling is applied on the WBCD.

```
In [21]: from sklearn.metrics import precision_score,recall_score, f1_score, confusion_matrix

In [22]: y_pred = nn_clf.predict(x_test)

In [23]: confusion_matrix(y_test,y_pred)

Out[23]: array([[84,  2],
                [ 2, 52]], dtype=int64)

In [24]: from sklearn import metrics
         from sklearn.metrics import classification_report
         print('accuracy %s' %metrics.accuracy_score(y_pred, y_test))
         print(classification_report(y_test,y_pred,target_names=["Benign","Malignant"]))

accuracy 0.9714285714285714
             precision    recall  f1-score   support

     Benign       0.98      0.98      0.98        86
  Malignant       0.96      0.96      0.96        54

avg / total       0.97      0.97      0.97       140
```

Figure 4. 20. Performance metrics for the MLP model (2)

Fig. 4.12 to Fig.4.19 show the code implementation of the performance evaluation using for the four prediction models. This performance evaluation is done for the four models in two cases:

Case 1: No feature scaling technique is employed (on WBCD) in the data preprocessing stage. Case 2: Feature scaling technique is employed (on WBCD) in the data preprocessing stage.

Table 4. 1. Performance metric for LR, SVM, NB, MLP (WBCD is not feature scaled)

NO FEATURE SCALING OF DATA			
Classifier	**Precision**	**Recall**	**F1-Score**
LR	**97%**	**97%**	**97%**
NB	95%	95%	95%
SVM	**97%**	**97%**	**97%**
MLP	43%	66%	52%

Table 4. 2. Performance metric for LR, SVM, NB, MLP (WBCD is feature scaled)

FEATURE SCALING OF DATA			
Classifier	**Precision**	**Recall**	**F1-Score**
LR	97%	97%	97%
NB	97%	97%	97%
SVM	**98%**	**98%**	**98%**
MLP	97%	97%	97%

Table 4.1 and Table 4.2 show the performance comparison of three prediction models with the Logistic regression model. When the data fed to the models is not normalized, we can see that SVM has the best prediction performance, LR and NB also performed

well, but MLP has a poor performance, owing to the fact that artificial neural networks perform badly in the presence of unbalanced data.

However, when the data is normalized, all the models performed very well, with SVM having 98% and the rest 97%. Notably, the performance of LR did not change, LR has f1-score of 97% for both cases, i.e. when data is scaled and when the data is not.

4.2 Our Contribution

Evaluating the performance of logistic regression model for the classification of breast cancer. In this research, we checked how logistic regression model handles cases with unscaled data and scaled data. The performance of our LR model is compared with three prediction models namely SVM, NB, MLP.

The implementation in this research work was done with machine learning libraries in Python programming language (version 3.6.5).

CHAPTER FIVE

SUMMARY, CONCLUSION AND FUTURE WORK

5.1 Summary

The need for an accurate predictor for the prediction of breast cancer cannot be overemphasized. Breast cancer has the second highest mortality rate, where lung cancer is the first and this cancer affects mostly women. For its detection and classification, physicians used mammography to make prognosis and diagnosis on their patients. However, the accuracy of mammography is less impressive, so the need for a better prediction facilitator is ever fervent.

Many researchers have employed the techniques of machine learning and artificial intelligence for the prediction and classification of breast cancer. These techniques take data as input, learn from the data and next time will be able to make predictions on any new data that has the same dimension with that which they learn from. In this research, the machine learning technique employed is logistic regression. Logistic regression is a statistical probabilistic model, which uses sigmoid function as its activation function. The data used in this work is Wisconsin breast cancer dataset from UCI online machine learning repository. The data has 11 attributes with 699 data points and 16 missing values. The missing values were filled up with mean value calculated from the non-missing values in 'Bare_Nuclei' features of the data. For the classification task, the data is split into two sets, which are training set (80% of data) and test set (20% of data).

The training set is used to train the logistic regression model and subsequently, the test set is used to test the trained model. In this work, we checked for the behavior of our model in cases where the data used for training and testing has its features scaled and that when its features are not scaled.

The performances of our model on both cases are compared with other extant prediction models namely, SVM, NB, and MLP (an artificial neural network). From our result analysis, SVM performed slightly better than our LR model, however, the notable observation from our work is that the performance our LR model remained the same for both cases, and this is not true for the other models. The performance metrics used for our performance evaluation are the confusion matrix, precision score, recall score and f1-score.

5.2 Conclusion

Logistic regression model does not necessarily require data feature scaling of data, neither is it greatly affected by unbalanced data nor dependency among data set features. Hence, for medium size data, logistic regression is a good probabilistic prediction model to employ for a binary classification problem, because of its simplicity and less time complexity; therefore, logistic regression model can be used for the prediction of breast cancer, which greatly help physicians to make proper and early diagnosis, which will go a long way in increasing the survivability rate of breast cancer patients.

5.3 Future work

For future work, we propose the development of an ensemble learning model, comprising Logistic regression, Artificial Neural Network, and Support Vector Machine for cancer predictions.

APPENDIX

```
#PYTHON IMPLEMENTATION CODE FOR BREAST CANCER CLASSIFICATION USING LOGISTIC REGRESSION

#SUPPORT VECTOR MACHINE, NAIVE BAYES, AND MULTILAYER PERCEPTRON

#@author----- Ude Anthony Anene

# coding: utf-8

#we import the relevant libraries needed for this work


import pandas as pd

import numpy as np

from matplotlib import pyplot as plt

get_ipython().run_line_magic('matplotlib', 'inline')

#using the pandas object we read in our csv file ( wisconsin breast cancer dataset)

df = pd.read_csv("WBCD_9_attributes.csv")


#Displays the first 3 rows of the dataframe

df.head(3)


# displays the data information

df.info()


#importing the imputer class to handle the missing the value

from sklearn.preprocessing import Imputer


# an imputer class object is created and given a strategy of mean

imputer = Imputer(missing_values='NaN',strategy='mean',axis=0)


df.iloc[:,[6]]= imputer.fit_transform(df.iloc[:,[6]])
```

```
df.Bare_Nuclei = df.Bare_Nuclei.astype(int)

df.info()
# x contains the independent variables and y contains the label
x = df.iloc[:,1:10]
y = df.iloc[:,10:11]

from sklearn.model_selection import train_test_split
# the data set is split into two sets: training set (80%) and test (20%)
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)

x_train.shape

# StandardScaler is used to normalize the dataset
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)

# LogisticRegression class imported from sklearn.learn_model
from sklearn.linear_model import LogisticRegression

# instance (our model) of LogisticRegression is created
lr = LogisticRegression()

#Using the fit method the model is trained
lr.fit(x_train,y_train)

#The prediction accuracy of the trained model is tested
```

```
lr.score(x_test,y_test)

# y_pred is variable to store the predicted values from the model
y_pred = lr.predict(x_test)

#Displays the predicted values
y_pred

from sklearn.metrics import precision_score,recall_score, f1_score,confusion_matrix

#Displays the confusion matrix for the Logistic regression model
confusion_matrix(y_test,y_pred)

#This displays the accuracy score, precision score, recall score and f1-score for the logistic regression model

from sklearn import metrics
from sklearn.metrics import classification_report
print('accuracy %s' %metrics.accuracy_score(y_pred, y_test))
print(classification_report(y_test,y_pred,target_names=["Benign","Malignant"]))

# pred_prob stores the probability for the labels
pred_prob = lr.predict_proba(x_test)[:,1]

pred_prob
```

```
#*****************************************************************************************
#SUPPORT VECTOR MACHINE IMPLEMENTATION MODEL STARTS HERE

# LinearSVC class imported from sklearn.svm
from sklearn.svm import LinearSVC

# instance (our model) of SVM is created
svm_clf = LinearSVC(C=1,loss='hinge')

#Using the fit method the model is trained
svm_clf.fit(x_train,y_train)

#The prediction accuracy of the trained model is tested
svm_clf.score(x_test,y_test)

# y_pred is variable to store the predicted values from the model
y_pred = svm_clf.predict(x_test)

#Displays the predicted values
y_pred

#Displays the confusion matrix for the Logistic regression model
confusion_matrix(y_test,y_pred)

#This displays the accuracy score, precision score, recall score and f1-score for the
#svm model
from sklearn import metrics
from sklearn.metrics import classification_report
print('accuracy %s' %metrics.accuracy_score(y_pred, y_test))
print(classification_report(y_test,y_pred,target_names=["Benign","Malignant"]))
```

```
#*******************************************************************************
#NAIVE BAYES MODEL IMPLEMENTATION CODE STARTS HERE

# GaussianNB class imported from sklearn.naive_bayes
#from sklearn.naive_bayes import GaussianNB

# instance (our model) of naive_bayes is created
gnb = GaussianNB()

#Using the fit method the model is trained
gnb.fit(x_train,y_train)

#The prediction accuracy of the trained model is tested
gnb.score(x_test,y_test)

# y_pred is variable to store the predicted values from the model
y_pred = gnb.predict(x_test)

#Displays the predicted values
y_pred
#Displays the confusion matrix for the Logistic regression model
confusion_matrix(y_test,y_pred)

#This displays the accuracy score, precision score, recall score and f1-score for the
#naive_bayes model

from sklearn import metrics
from sklearn.metrics import classification_report
print('accuracy %s' %metrics.accuracy_score(y_pred, y_test))
print(classification_report(y_test,y_pred,target_names=["Benign","Malignant"])
```

```
#*****************************************************************************************
#MULTILAYER PERCEPTRON IMPLEMENTATION CODE STARTS HERE

# MLPClassifier class imported from sklearn.neural_network
from sklearn.neural_network import MLPClassifier
# instance (our model) of MLPClassifier is created
mlp_clf=MLPClassifier(solver='adam',alpha=1e5,max_iter=10000,hidden_layer_sizes=(5,2), random_state=1)
#Using the fit method the model is trained
mlp_clf.fit(x_train,y_train)

#The prediction accuracy of the trained model is tested
mlp_clf.score(x_test,y_test)
# y_pred is variable to store the predicted values from the model
y_pred = mlp_clf.predict(x_test)

#Displays the predicted values
y_pred
from sklearn.metrics import precision_score,recall_score, f1_score,confusion_matrix
#Displays the confusion matrix for the Logistic regression model
confusion_matrix(y_test,y_pred)
#This displays the accuracy score, precision score, recall score and f1-score for the #MLPClassifier
from sklearn import metrics
from sklearn.metrics import classification_report
print('accuracy %s' %metrics.accuracy_score(y_pred, y_test))
print(classification_report(y_test,y_pred,target_names=["Benign","Malignant"]))

#The code ends here!
```

REFERENCE

Aaltonen, L. A., Salovaara, R., Kristo, P., Canzian, F., Hemminki, A., Peltomäki, P., … de la Chapelle, A. (1998). Incidence of hereditary nonpolyposis colorectal cancer and the feasibility of molecular screening for the disease. *The New England Journal of Medicine*. https://doi.org/10.1056/NEJM199805213382101

Abedin, T., Chowdhury, M. Z. I., & Afzal, A. (2016). Review Article Application of Binary Logistic Regression in Clinical Research. *Journal of National Heart Foundation of Bangladesh*, *5*(1), 8–11.

Agarap, A. F. (2017). *On Breast Cancer Detection: An Application of Machine Learning Algorithms on the Wisconsin Diagnostic Dataset*. (1), 5–9. https://doi.org/10.1145/3184066.3184080

Aggarwal, C. C. (2015). Data Mining. In *Journal of Visual Languages & Computing* (Vol. 11).

Aggarwal, C. C., & Xhai, C. (2012). A survery of text clustring algorithms. *Mining Text Data*, *8*, 77–128. https://doi.org/10.1007/978-1-4614-3223-4

Agrawal, R., Gunopulos, D., & Leymann, F. (n.d.). *Workflow and Scientific Databases Mining Process Models from Workflow Logs*. Retrieved from https://link.springer.com/content/pdf/10.1007%2FBFb0101003.pdf

Alasadi, S. A., & Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*. https://doi.org/10.3923/jeasci.2017.4102.4107

Amrane, M., Oukid, S., Gagaoua, I., & Ensari, T. (2018). Breast cancer classification using machine learning. *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting, EBBT 2018*, 1–4. https://doi.org/10.1109/EBBT.2018.8391453

Bazazeh, D., & Shubair, R. (2016). *Comparative Study of Machine Learning Algorithms for Breast Cancer Detection and Diagnosis*. 2–5.

Cabrera, A. F. (2007). *Logistic Regression Analysis in Higher Education : An Applied Perspective* *. (814).

Cleophas, T. J., & Zwinderman, A. H. (2013). Machine learning in medicine. In *Machine Learning in Medicine*. https://doi.org/10.1007/978-94-007-5824-7

Frank, E., Trigg, L., Holmes, G., & Witten, I. H. (2000). Technical note: Naive Bayes for regression. *Machine Learning*, *41*(1), 5–25. https://doi.org/10.1023/A:1007670802811

Garg, B., Beg, S. M. M., & Ansari, A. Q. (2009). Optimizing Number of Inputs to Classify Breast Cancer Using Artificial Neural Network. *Journal of Computer Science & Systems Biology*, *02*(04), 247–254. https://doi.org/10.4172/jcsb.1000037

Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn* (First Edit). CA: O'REILLY.

Goyal, A., & Mehta, R. (2012). Performance comparison of Naïve Bayes and J48 classification algorithms. *International Journal of Applied Engineering Research*.

Grus, J. (2015). Data Science from Scratch. In *Climate Change 2013 - The Physical Science Basis*. https://doi.org/10.1017/CBO9781107415324.004

Guo, X., Huang, X., & Zhang, L. (2014). Three-dimensional wavelet texture feature extraction and classification for multi/hyperspectral imagery. *IEEE Geoscience and Remote Sensing Letters*, *11*(12), 2183–2187. https://doi.org/10.1109/LGRS.2014.2323963

Gupta, M., & Gupta, B. (2018). A Comparative Study of Breast Cancer Diagnosis Using Supervised Machine Learning Techniques. *2018 Eleventh International Conference on Contemporary Computing (IC3)*, 1–6. IEEE.

Haigh, J., Cox, D. R., & Snell, E. J. (2007). Analysis of Binary Data. *The Mathematical Gazette*. https://doi.org/10.2307/3618895

Huang, M. W., Chen, C. W., Lin, W. C., Ke, S. W., & Tsai, C. F. (2017). SVM and SVM ensembles in breast cancer prediction. *PLoS ONE*, *12*(1), 1–14. https://doi.org/10.1371/journal.pone.0161501

Ivančaková, J., Babič, F., & Butka, P. (2018). Comparison of different machine learning methods on Wisconsin dataset. *SAMI 2018 - IEEE 16th World Symposium on Applied Machine Intelligence and Informatics Dedicated to the Memory of Pioneer of Robotics Antal (Tony) K. Bejczy, Proceedings*, *2018-Febru*, 173–178. https://doi.org/10.1109/SAMI.2018.8324834

Jhajharia, S., Varshney, H. K., Verma, S., & Kumar, R. (2016). A neural network based breast cancer prognosis model with PCA processed features. *2016 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2016*, 1896–1901. https://doi.org/10.1109/ICACCI.2016.7732327

Jothi, N., Rashid, N. A., & Husain, W. (2015). Data Mining in Healthcare - A Review. *Procedia Computer Science*. https://doi.org/10.1016/j.procs.2015.12.145

Jouni, H., Issa, M., Harb, A., Jacquemod, G., & Leduc, Y. (2016). Neural Network architecture for breast cancer detection and classification. *2016 IEEE International Multidisciplinary Conference on Engineering Technology, IMCET 2016*. https://doi.org/10.1109/IMCET.2016.7777423

K.Srinivas, Rani, B. K., & A.Govrdhan, D. (2010). Applications of Data Mining Techniques in Healthcare and Prediction of Heart Attacks. *(IJCSE) International Journal on Computer Science and Engineering*. https://doi.org/10.1.1.163.4924

Kaymak, S., Helwan, A., & Uzun, D. (2017). Breast cancer image classification using artificial neural networks. *Procedia Computer Science*. https://doi.org/10.1016/j.procs.2017.11.219

Kharya, S, Dubey, D., & Soni, S. (2013). *Predictive Machine Learning Techniques for Breast Cancer Detection*. *4*(6), 1023–1028.

Kharya, Shweta. (2012). Using Data Mining Techniques for Diagnosis and Prognosis of Cancer Disease. *International Journal of Computer Science, Engineering and Information Technology*. https://doi.org/10.5121/ijcseit.2012.2206

Khourdifi, Y., & Bahaj, M. (2019). Applying best machine learning algorithms for breast cancer prediction and classification. *2018 International Conference on Electronics, Control, Optimization and Computer Science, ICECOCS 2018*, 1–5. https://doi.org/10.1109/ICECOCS.2018.8610632

Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2006). Paper 1 (20). *Artificial Intelligence Review*, *26*, 159–190. https://doi.org/10.1007/s10462-007-9052-3

Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction. *Computational and Structural Biotechnology Journal*. https://doi.org/10.1016/j.csbj.2014.11.005

Kriegel, H.-P., Borgwardt, K. M., Kröger, P., Pryakhin, A., Schubert, M., & Zimek, A. (2007). Future trends in data mining. *Data Mining and Knowledge Discovery*, *15*(1), 87–97. https://doi.org/10.1007/s10618-007-0067-9

Learned-miller, E. G. (2014). *Introduction to Supervised Learning*. 1–5.

Li, L., Wang, X., Du, X., Liu, Y., Liu, C., Qin, C., & Li, Y. (2017). Classification of heart sound signals with BP neural network and logistic regression. *Proceedings - 2017 Chinese Automation Congress, CAC 2017*, *2017-Janua*, 7380–7383. https://doi.org/10.1109/CAC.2017.8244111

Li, T., Ma, S., & Ogihara, M. (2005). WAVELET METHODS IN DATA MINING. *Data Mining and Knowledge Discovery Handbook*.

Liu, L. (2018). Research on logistic regression algorithm of breast cancer diagnose data by machine learning. *Proceedings - 2018 International Conference on Robots and Intelligent System, ICRIS 2018*, 157–160. https://doi.org/10.1109/ICRIS.2018.00049

Mangasarian, O. L., Street, W. N., & Wolberg, W. H. (2008). Breast Cancer Diagnosis and Prognosis Via Linear Programming. *Operations Research*, *43*(4), 570–577. https://doi.org/10.1287/opre.43.4.570

Manjusha, K. ., Sankaranarayanan, K., & Seena, P. (2015). Data Mining in Dermatological Diagnosis: A Method for Severity Prediction. *International Journal of Computer Applications*. https://doi.org/10.5120/20597-3102

Marwala, T. (2018). Multi-layer Perceptron. *Handbook of Machine Learning*, (2001), 23–42. https://doi.org/10.1142/9789813271234_0002

McGee, M. (2013). Logistic Regression. In *Key Topics in Clinical Research*. https://doi.org/10.3109/9780203450307-30

Mitra, S., & Acharya, T. (2004). Data Mining in Multimedia, Soft Computing and Bioinformatics. In *A John Wiley & Sons, INC., pulication* (Vol. 106).

Peng, C. Y. J., Lee, K. L., & Ingersoll, G. M. (2002). An introduction to logistic

regression analysis and reporting. *Journal of Educational Research*. https://doi.org/10.1080/00220670209598786

Rashmi, G. D., Lekha, A., & Bawane, N. (2016). Analysis of efficiency of classification and prediction algorithms (Naïve Bayes) for Breast Cancer dataset. *2015 International Conference on Emerging Research in Electronics, Computer Science and Technology, ICERECT 2015*, 108–113. https://doi.org/10.1109/ERECT.2015.7498997

Rodrigues, L. (2016). Analysis of the Wisconsin Breast Cancer Dataset and Machine Learning for Breast Cancer Detection Analysis of the Wisconsin Breast Cancer Dataset and Machine Learning for Breast Cancer Detection. *XI Workshop de Visão Computacional*, (December), 415–423.

Tao, Y., Faloutsos, C., Papadias, D., & Liu, B. (2004). *Prediction and indexing of moving objects with unknown motion patterns*. https://doi.org/10.1145/1007568.1007637

Uppal, M. T. N. (2016). Classification of mammograms for breast cancer detection using fusion of discrete cosine transform and discrete wavelet transform features. *Biomedical Research (India)*, *27*(2), 322–327.

Williams, G., & Williams, G. (2011). Descriptive and Predictive Analytics. In *Data Mining with Rattle and R*. https://doi.org/10.1007/978-1-4419-9890-3_8

Yusuff, H., Mohamad, N., Ngah, U. K., & Yahaya, A. S. (2012). Breast cancer analysis using logistic regression. *International Journal of Recent Research and Applied Studies*.