TRIBHUVAN UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

COMPARATIVE ANALYSIS OF RANDOM FOREST AND LOGISTIC REGRESSION FOR DIAGNOSIS OF DIABETES MELLITUS

Dissertation

Submitted to

Central Department of Computer Science and Information Technology

Kirtipur, Kathmandu, Nepal

In Partial Fulfillment of the Requirements for

Master's Degree in Computer Science & Information Technology

by

Mr. Madhu Pandey

June, 2019

Under the Guidance of

Asst. Prof. Dhiraj Kedar Pandey

Central Department of Computer Science and Information Technology

Kirtipur, Kathmandu, Nepal

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Technology

Student's Declaration

I hereby declare that I am the only author of this work and that no sources other than listed here have been used in this work.

_____

Madhu Pandey
Date:

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Technology

Supervisor's Recommendation

I hereby recommend that this dissertation prepared under my supervision by **Mr. Madhu Pandey** titled "**Comparative Analysis of Random Forest and Logistic Regression for Diagnosis of Diabetes Mellitus**" in partial fulfillment of the requirements for the degree of M.Sc. in Computer Science and Information Technology be processed for the evaluation.

Asst. Prof. Dhiraj Kedar Pandey

Central Department of Computer Science and Information Technology,

Tribhuvan University,

Kathmandu, Nepal

(**Supervisor**)

Date:

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Technology

LETTER OF APPROVAL

We certify that, we have read this dissertation and, in our opinion, it is satisfactory in the scope and quality as a dissertation in partial fulfillment for the requirement of Master's Degree in Computer Science and Information Technology.

Evaluation Committee

Asst. Prof. Nawaraj Poudel
Central Department of CSIT,
Tribhuvan University,
Kathmandu, Nepal
(**Head of Department**)

Asst. Prof. Dhiraj Kedar Pandey
Central Department of CSIT,
Tribhuvan University,
Kathmandu, Nepal
(**Supervisor**)

(**External Examiner**)

(**Internal Examiner**)

ACKNOWLEDGEMENT

I am highly indebted to my thesis supervisor, Asst. Prof. Dhiraj Kedar Pandey, Central Department of Computer Science and Technology, Kirtipur, Kathmandu for his valuable and constructive suggestion during the planning and development of this research. Otherwise it would have never seen the light of the day. His willingness to give his time so generously has been very much appreciated.

Also my gratitude goes to Asst. Prof. Nawaraj Poudel Head of Central Department of Computer Science and Technology, Kirtipur, Kathmandu.

My special thanks to Mr. Indra Chaudhary for frequent discussion, Mr. Suresh Kumar Mukhiya, all my colleagues and best wishers who exhorted me for the initiation.

Mr. Madhu Pandey

…….....

…….....

ABSTRACT

In our daily life there is lots of data in different field. Whenever there is data we can have lots of information, patterns, meaning etc. and the process of Extracting or "mining" knowledge from large amount of data is called Data mining and is also known as "Knowledge discovery from data (KDD)". Data mining applications has got rich focus due to its significance of classification algorithms. Diabetes Mellitus (DM) is a result of bad metabolism. DM, if not controlled, causes several complications and even affects other parts of the body. This study aims to survey on the two different classifiers with dataset of patients regarding Diabetes Mellitus and to implement as well as assist by comparing Random Forest and Logistic Regression classification techniques to standardize the diagnosis and treatment of Diabetes Mellitus. From the context analysis it was seen that Logistic Regression was able to classify 81.17% of the data correctly which was better than Random Forest in comparison to results of evaluation metrics (Accuracy, Precision, Recall and F-Measure). In a nut shell, the experiment result showed that Logistic Regression had got 2% better accuracy than Random Forest for the diagnosis of diabetes mellitus.

Keywords: Data Mining, Decision Tree, Diabetes Mellitus, Logistic Regression, Random Forest

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

BMI	:	Body Mass Index
CART	:	Classification and Regression Tree
DM	:	Diabetes Mellitus
FN	:	False Negative
FP	:	False Positive
ID3	:	Iterative Dichotomiser
KDD	:	Knowledge Discovery from Data
PCU	:	Primary Care Unit
TN	:	True Negative
TP	:	True Positive

CHAPTER 1
INTRODUCTION

1.1. Background of the Study

In our daily life there are lots of data in different fields. Whenever there is data, we can have lots of information, patterns, meaning etc. and information is an important asset for an organization during this competitive global market. The information can be stored in computer in the form file, database or data warehouse. Moreover, this information helps us to extract knowledge for decision making. Good decision-making process helps us for identifying, selecting, and implementing alternatives. The right information, in the right form, at the right time is needed to make good decisions. The process of extracting or "mining" knowledge from large amount of data is called Data Mining [1]. Data mining also can be defined as exploration and analysis of large quantities of data to discover meaningful pattern from data and is also known as "Knowledge Discovery from Data (KDD)" [1].

Decision Tree is also the most widely applied supervised machine learning or classification technique. The learning and classification steps of decision tree induction are simple and fast and it can be applied to any domain [2]. Logistic regression is one of the simpler classification models[3]. It has been around for a long time but is still widely used. Because of its parametric nature it can to some extent be interpreted by looking at the parameters making it useful when experimenters want to look at relationships between variables.

Human body needs energy to do different daily activities. The source of this energy is the food one consumes. Pancreas is an organ in human body that lies near the stomach; it produces an important hormone called insulin, which helps glucose to flow all over the cells of a human body. Diabetes Mellitus (DM) is a result of bad metabolism, in which the body fails to make sufficient insulin or cannot utilize it the way it should be utilized. DM, if not controlled, causes several complications and even affects other parts of the body like heart, nerves, eyes, kidneys, and so on.

1.2. Statement of Problem

Data mining applications have got rich focus due to its significance of classification algorithms. The comparison of classification algorithm is a complex task and it is an open problem. First, the notion of the performance can be defined in many ways: accuracy, speed, cost, reliability etc. Second, an appropriate tool is necessary to quantify this performance. Third, a consistent method must be selected to compare with the measured values. The selection of the best classification algorithm for a given dataset is a very widespread problem. In this sense; it requires to make several methodological choices.

Early detection of diabetes is crucial for active management for people who have been newly diagnosed and have not developed complications yet. It is unlikely to expect everybody to be aware of the early symptoms of diabetes and visit a doctor. This study, hence, focuses on a potential system which can help a healthcare professional to early diagnosis of diabetes with the help of one the widely used classification algorithms.

1.3. Objectives of the Study

The objectives of this research are:

- To survey on the two different classifiers with dataset of patients regarding Diabetes Mellitus.
- To implement and assist by comparing Random Forest and Logistic Regression classification techniques to standardize the diagnosis and treatment of Diabetes Mellitus.

1.4. Limitations of the Study

Limitations of the research are:

- This study was done by comparison between two classification algorithms. (Random Forest and Logistic Regression).
- This research focused on comparison of Accuracy, Precision, Recall, and F-measure of the implemented algorithms.
- Dataset comprised of 8 attributes.
- All the algorithms were implemented in Python version 3.7.3.

1.5. Structure of the Report

This report is organized in five chapters and is enlisted below:

- Chapter 1 **"Introduction"** explains the background of the study, statement of problems, objectives of the study as well as limitations of the study.
- Chapter 2 **"Literature Review"** describes the various concepts of data mining, diabetes mellitus and related works in the domain.
- Chapter 3 **"Research Methodology"** explains the framework of the research and implemented algorithms.
- Chapter 4 **"Experiment and Result"** explains about experiments, results, evaluation and context analysis.
- Chapter 5 **"Conclusion and Future Works"** describes the conclusion and future works for the upcoming researcher.
- **References**
- **Appendix A**
- **Appendix B**

CHAPTER 2
LITERATURE REVIEW

2.1. Data Mining

In our daily life; there are lots of data in different fields. Whenever there is data we can have lots of information, patterns, meaning etc. and information is an important asset for an organization during this competitive global market. Moreover, this information helps us to extract knowledge for decision making. Good decision-making process helps us for identifying, selecting, and implementing alternatives. The right information, in the right form, at the right time is needed to make good decisions. The process of extracting or "mining" knowledge from large amount of data is called data mining [1]. Data mining also can be defined as Exploration and analysis of large quantities of data to discover meaningful pattern from data and is also known as "Knowledge Discovery from Data (KDD)" [1].

In data mining [1] there are lots of techniques to mine the knowledge from data which are recently used widely in different fields such as Business, Scientific Research, Computer Science, Machine Learning, Information Science, Statistics, and Database Technology etc. Most commonly used data mining techniques are **Classification**, **Dependencies and Associations, Regression and Clustering.** These above-mentioned techniques are effectively used in different fields separately.

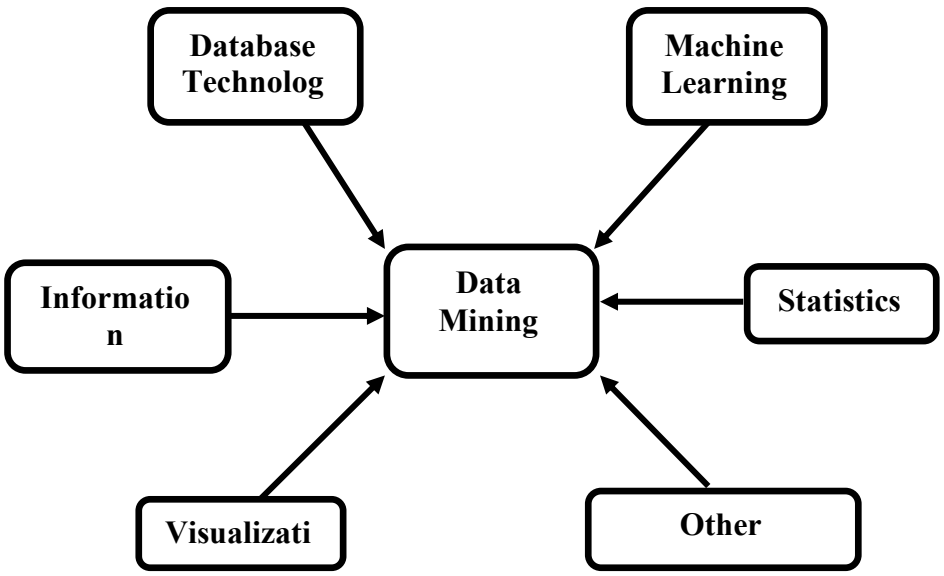


Figure 2.1: Data mining as confluence of multiple disciplines

2.1.1. Machine Learning

Machine learning investigates how computers can learn or improve their performance based on data. It is the main research area is for computer programs to automatically learn to recognize complex patterns and make intelligent decision based on the data. For example: a system that can automatically recognize hand written postal codes on mail after learning from a set of examples. Machine learning are sub divided into two parts i.e. supervised learning and unsupervised learning.

- **Supervised Learning**

Supervised learning is fairly common in classification problems because the goal is often to get the computer to learn a classification system that we have created. Digit recognition, once again, is a common example of classification learning. More generally, classification learning is appropriate for any problem where deducing a classification is useful and the classification is easy to determine. Supervised learning is the most common technique for training neural networks and decision trees. Both of these techniques are highly dependent on the information given by the pre-determined classifications [4].

- **Unsupervised Learning**

Unsupervised learning seems much harder: the goal is to have the computer learn how to do something that we don't tell it how to do! There are actually two approaches to unsupervised learning. The first approach is to teach the agent not by giving explicit categorizations, but by using some sort of reward system to indicate success. Note that this type of training will generally fit into the decision problem framework because the goal is not to produce a classification but to make decisions that maximize rewards. This approach nicely generalizes to the real world, where agents might be rewarded for doing certain actions and punished for doing others. A second approach is called clustering. In this type of learning, the goal is not to maximize a utility function, but simply to find similarities in the training data [4].

2.1.2. Classification

Classification or prediction is the most widely used data mining task. Classification algorithms are supervised methods that uncover the hidden relationship between the target class and the independent variables [5]. Supervised learning algorithms allow

labels to be assigned to the observations so that new data can be classified based on training data [1, 5]. Examples of classification tasks are image and pattern recognition, medical diagnosis, loan approval, detecting faults or financial trends [5].

It is a two-step process

1. Model Construction (Learning step or Training Phase)
    - Build a model to explain the target concept
    - Model is represented as classification rules, decision trees, or mathematical formulae
2. Model Usage (Testing Phase)
    - is used for classifying future or unknown cases
    - estimate the accuracy of the model

Decision Tree

A decision tree is a flowchart-like tree structure, where each internal node (non-leaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label. The topmost node in a tree is the root node. A typical decision tree is shown in *figure 2.2*[1].

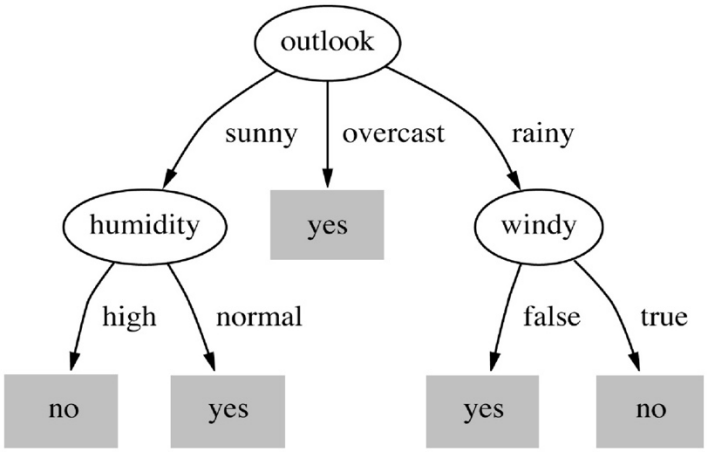


Figure 2.2: Decision tree example

During the late 1970s and early 1980 J. Ross Quinlan, a researcher in machine learning, developed a decision tree algorithm known as **ID3** (Iterative Dichotomiser). Quinlan later presented C4.5[6, 7] (a successor of ID3), which become a benchmark to which newer supervised learning algorithms are often compared. In 1984, a group of statisticians published the book classification and regression trees (CART)[7],

which described the generation of binary decision trees. ID3 and CART were invented independently of one another at around the same time, yet follow a similar approach for learning decision trees from training tuples. These two cornerstone algorithms spawned a flurry of work on decision tree induction. The basic decision tree algorithm is summarized as below:

- **Decision Tree Construction Algorithm**

Input: A data set, D

Output: A decision tree

- If all the instances have the same value for the target attribute then return a decision tree that is simply this value (not really a tree - more of a stump).
- Else
  1. Compute Gain values for all attributes and select an attribute with the highest value and create a node for that attribute.
  2. Make a branch from this node for every value of the attribute
  3. Assign all possible values of the attribute to branches.
  4. Follow each branch by partitioning the dataset to be only instances whereby the value of the branch is present and then go back to 1.

- **Attribute Selection Measures**

In a data set there are lots of attributes and we do have problem on selection of attribute as node and as leaf. There arise questions **which attribute first?**

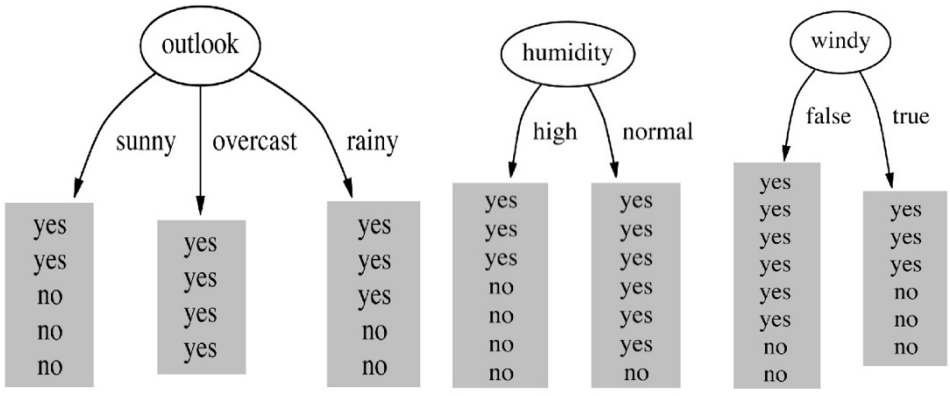


Figure 2.3: Possibility of attribute as node

Attribute selection measure [1] is a heuristic for selecting the splitting criterion that "best" separates given data partition, D, of class-labeled training tuples into individual

classes. Attribute selection measures are also known as splitting rules because they determine how the tuples at a given node are to be split. The attribute selection measure provides a ranking for each attribute describing the given training tuples. The attribute having the best score for the measure is chosen as the splitting attribute for the given tuples.

- **Information Gain**

ID3 uses information gain as its attribute selection measure. This measure is based on pioneering work by Claude Shannon on information theory, which studied the value or "information content" of messages [1].

Information gain = (information before split) – (information after split) bits

$$Gain(A) = Info(D) - Info_A(D)\ bits$$ ---------- Equation2.1

Where,

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)\ bits$$ ---------- Equation 2.2

- $P_i = |C_{i,D}| / |D|$
- A having v distinct value, $\{a_1, a_2, \ldots, a_v\}$
- $D_1, D_2, \ldots, D_v$then,

$$Info_A(D) = \sum_{j=1}^{v} \frac{|Dj|}{|D|} \times Info(Dj)\ bits$$ ---------- Equation 2.3

- **Gain Ratio:**

The information gain [1] measure is biased toward tests with many outcomes. That is, it prefers to select attributes having a large number of values. C4.5[6, 7], a successor of ID3, uses an extension to information gain known as **gain ratio**, which attempts a overcome this bias. It applies a kind of normalization to information gain using a "Split information" value define analogously with $Info(D)$as

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|Dj|}{|D|} \times \log_2\left(\frac{|Dj|}{|D|}\right) bits$$ ---------- Equation 2.4

The gain ratio is defined as

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$ ---------- Equation 2.5

The attribute with the maximum gain ratio is selected as the splitting attribute. Note, however, that are the split information approaches 0, the ratio becomes unstable. A constraint is added to avoid this, whereby the information gain of the test selected must be large-at least as great as the average gain over all tests examined.

- **Gini Index**

The Gini index [1] is used in CART [7]. Using the notation previously described, the Gini index measures the impurity of D, a data partition or set of training tuples, as

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2 \; bits$$ ---------- Equation 2.6

Where, P_i is the probability that a tuple in D belong to class C_i and is estimated by $|C_{i,D}| / |D|$. The sum is computed over m classes. The Gini index considers a binary split for each attribute. Let's first consider the case where A is a discrete-valued attribute having *v* distinct values, $\{a_1, a_2, ...,a_v\}$, occurring in D. If A has v possible values, then there are 2^v possible subsets but we exclude the power set, and the empty set from consideration since, conceptually, they do not represent a split. Therefore, there are 2^v-2 possible ways to form two partitions of the data, D, based on a binary split on A.

When considering split, we compute a weighted sum of the impurity of each resulting partition. For example, if a binary split on A partitions D into D_1 and D_2, the Gini index of D given that partitioning is

$$Gini_A(D) = \left\{ \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \right\} bits$$ ---------- Equation 2.7

For each attribute, each of these possible binary splits is considered. For discrete-valued attribute, the subset that gives the minimum Gini index for that attribute is selected as its splitting subset.

The reduction in impurity that would be incurred by a binary split on a discrete-or continuous-valued attribute A is

$$\Delta Gini(A) = \{Gini(D) - Gini_A(D)\} bits$$ ---------- Equation 2.8

The attribute that maximizes the reduction in impurity (or, equivalently, has the minimum Gini index) is selected as the splitting attribute.

Random Forest

Arbitrary Woods are an ensemble of decision woods, and derive from ensemble learning techniques for classification and regression. They're also looked at as form of a nearest friend predictor, that construct numerous decision woods at instruction time and result the method of the courses because the result class. Arbitrary Woods take to reduce the problems with high bias and difference by processing a typical, and managing the two extremes. Moreover, Arbitrary Woods have hardly any parameters to song and the majority of the time work well by simply using them with parameter settings collection to default values [8].

Bagging

Bagging is also known as Bootstrap Aggregating. It is a met algorithm which helps to improve the accuracy of algorithms used in statistical classification and regression. It also reduces variance and helps to avoid over-fitting problems. Bagging is a special case of the model averaging approach [1].

Logistic Regression

Logistic regression [3] is one of the simpler classification models. It has been around for a long time but is still widely used. Because of its parametric nature it can to some extent be interpreted by looking at the parameters making it useful when experimenters want to look at relationships between variables.

2.2. Diabetes Mellitus

Diabetes Mellitus is not a single hereditary disease but a heterogeneous group of diseases, all of which ultimately lead to an elevation of glucose in the blood (hyperglycaemia) and loss of glucose in the urine as hyperglycaemia increases. It is also characterized by the three "polys" and inability to reabsorb water, resulting in increased urine production (polyurea) excessive thirst (polydipsia) and excessive eating (polyphagia) [9]. The types of Diabetes Mellitus are Type 1 DM and Type 2 DM.

2.2.1. Type 1 Diabetes Mellitus

Occurs abruptly, characterized by an absolute deficiency of insulin due to a marked decline in the number of insulin producing beta cells (perhaps caused by the auto immune destruction of beta cells) even though target cells contain insulin receptors.

Type 1 DM is also known as insulin dependent diabetes and juvenile onset diabetes, as it most commonly develops in people under 20 years old though it persists through life, and requires periodic insulin injections to treat it. Although type 1 DM appears to have certain genes which make them more susceptible, some triggering factor is required e.g. viral infection, shock etc.

2.2.2. Type 2 Diabetes Mellitus

It most often occurs in people who are over forty and overweight hence another name is "maturity onset diabetes". Clinical symptoms are mild, and high glucose levels in the blood can usually be controlled by diet, exercise, and/or with anti-diabetic drugs.
Some type 2 DM have sufficient amounts of insulin in the blood, but they have defects in the molecular machinery that mediates the action of insulin on its target cells, cells can become less sensitive to insulin because they have fewer insulin receptors. Type 2 DM is therefore called non-insulin dependent diabetes. 90% of all cases are type 2 DM.

2.3. Related Works

According to research [10] machine learning as a paradigm that may refer to learning from past experience to improve future performance. The sole focus of this field is automatic learning methods. Learning refers to modification or improvement of algorithm based on past “experiences” automatically without any external assistance from human.

The study on the classifiers for the risk of diabetes prediction carried out in [11] a web application is developed and prior to this development, thirteen classification models (Decision tree, Neural Network, Logistic Regression, Naïve Bayes and Random Forest algorithms including combination of Bagging and Boosting techniques except Random Forest) were evaluated to build a predictive model. Furthermore, accuracy

and ROC curve were calculated and compared with each other to investigate the robustness of each model. This method concluded that if important variables are considered, then Random Forest stands tall before all the other classification models.

Analysis and prediction of diabetes diseases using machine learning algorithm was introduced by authors of [12]. These methods used various data mining techniques of machine learning algorithm and were applied in different medical data set. The study revealed that single algorithm provided less accuracy than ensemble one. The research work [13] have compared four prediction models namely, J48, KNN, SVM and Random Forest for predicting diabetes mellitus using 8 important attributes under two different scenarios: one is before pre-processing the dataset where the decision tree J48 classifier gave the best result with accuracy 73.82% and on the other hand, both KNN and Random Forest performed quite well than the rest classifiers and provided 100% accuracy which concluded that dataset containing no noisy data provides better result for prediction.

According to [14] the prediction of diabetes diagnosis using classification-based data mining techniques had used Binary Logistic Regression, Multilayer Perceptron, and k-nearest Neighbor classifiers. In this method, Binary Logistic Regression has yielded accuracy of 69%, Multilayer Perceptron 71% and k-nearest Neighbor 80%. This research was carried out in a multidimensional diabetes dataset containing 100 observations with 7 features. The comparative study showed that out of 100 instances, Binary Logistic Regression correctly classified 72 instances and 28 instances were incorrectly classified, whereas in case of Multilayer Perceptron Technique, 74 instances were correctly classified, 26 incorrectly classified, and K-Nearest Neighbor correctly classified 81 instances and 19 were incorrectly classified. The evaluation measures of the algorithms were done by sensitivity, specificity and accuracy.

Another approach suggested in [15] uses a Random Forest algorithm to analyze on diabetes complication data where classification results of Decision tree, bagging with decision tree-based classifier, Random Forest with all input attributes, and Random Forest with feature selection are compared. This method concludes that Random Forest with feature selection gives the best result which overcame the overfitting problem generated due to missing values in the datasets. However, it has used a small

dataset, and suggested to use large dataset and study different types of learning settings. The data were collected from Sawanpracharak Regional Hospital, which consisted of 27 Primary Care Units (PCU). There were altogether 7,498 instances consisting of patients related to eye disease, kidney disease, heart disease and stoke diabetes and 18 attributes. The classification results showed that with small number of attributes (14 attributes), Random Forest gave better result up to the accuracy of 94.743%.

The research carried out in [16] had compared two traditional classification methods (Logistic regression and Fisher linear discriminant analysis) and four other machine learning classifiers namely neural networks, support vector machines, fuzzy c-mean, and random forest. The dataset including 6500 instances was collected from the Iranian national non-communicable diseases risk factors surveillance. The performances of those six classifiers were compared in terms of sensitivity, specificity, area under the curve, and total accuracy. When logistic regression and random forest are considered, logistic regression showed sensitivity, specificity, area under the curve, and total accuracy as 0.133, 0.999, 0.763, and 0.935 respectively. On the other hand, random forest showed 0.081, 0.998, 0.717, and 0.930 respectively. It can be said that, logistic regression performed somewhat better than random forest in this study.

CHAPTER 3
RESEARCH METHODOLOGY

3.1.Background

This chapter deals with the framework of research and used algorithms.

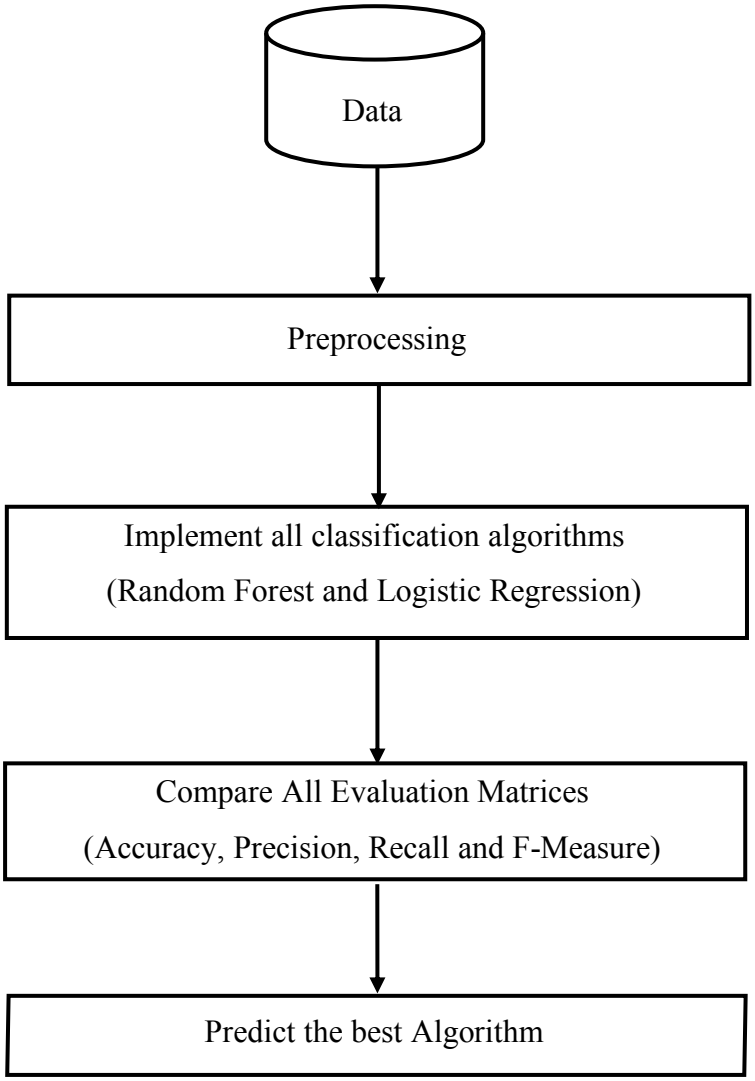


Figure 3.1: Implementation Model

3.2. Algorithms

In this research, four classification algorithms were implemented and they are

a) Random Forest b) Logistic Regression

3.2.1. Random Forest

Random Forest [17, 18] constructs random forests by bagging ensembles of random trees. It combines learning method for classification and regression. It is operated by using a collection of multiple decision trees at training time and outputting the class by individual trees. This algorithm is combination of two ideas i.e. "bagging" and "random decision forest". In this algorithm, the individual decision trees are generated using a random selection of attributes at each node to determine the split. Each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. Each tree votes and the most popular class are returned. Bagging is also known as Bootstrap Aggregating. It is a met algorithm which helps to improve the accuracy of algorithms used in statistical classification and regression. It also reduces variance and helps to avoid over-fitting problems [1].

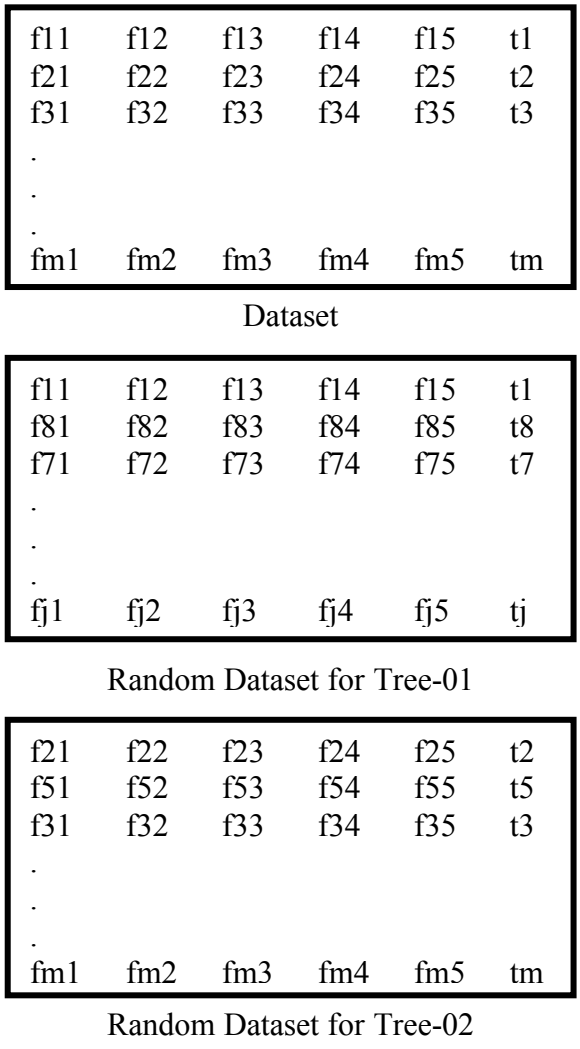


Figure 3.2: Random feature selection by bagging of Random Forest

- **Algorithm**

1. Let *N* be the number of training cases and let *M* be the number of variables in the classifier.
2. Let *F* be the input variables to be used to determine the decision at a node of the tree; *F* should be much less than *M*.
3. Choose a training set for this tree by choosing k times with replacement from all *N* available training cases
4. For each node of the tree, randomly choose *F* variables on which to base the decision at that node. Calculate the best split based on these *F* variables in the training set.
5. Each tree is fully grown and not pruned.

3.2.2. Logistic Regression

Logistic regression is a well-known technique borrowed by machine learning from the field of statistics [19]. It takes real valued inputs and makes a prediction as to the probability of the input belonging to the default class (say, class 0). If the probability is greater than 0.5, the prediction goes for the class 0, otherwise, class 1.

For the dataset given in Table 4.2, the logistic regression has 9 coefficients, as:

$$Output = b_0 + b_1.x_1 + b_3.x_2 + \ldots\ldots\ldots + b_8.x_8 \quad \text{..... Equation 3.1}$$

If we suppose Pregnancy be denoted by x_1, Glucose by x_2,……..., Age by x_8 respectively.

The job of the learning algorithm will be to discover the best values for the regression coefficients viz. b_0, b_1, ……, b_8 based on the training data. The output is then transformed into a probability using the logistic function:

$$p(class = 0) = \frac{1}{1 + e^{-output}} \quad \text{.....Equation 3.2}$$

Instead of fitting a straight line or hyper plane, the logistic regression model uses the logistic function as defined in equation 3.2 to squeeze the output of a linear equation between 0 and 1 as shown in the figure 3.3.

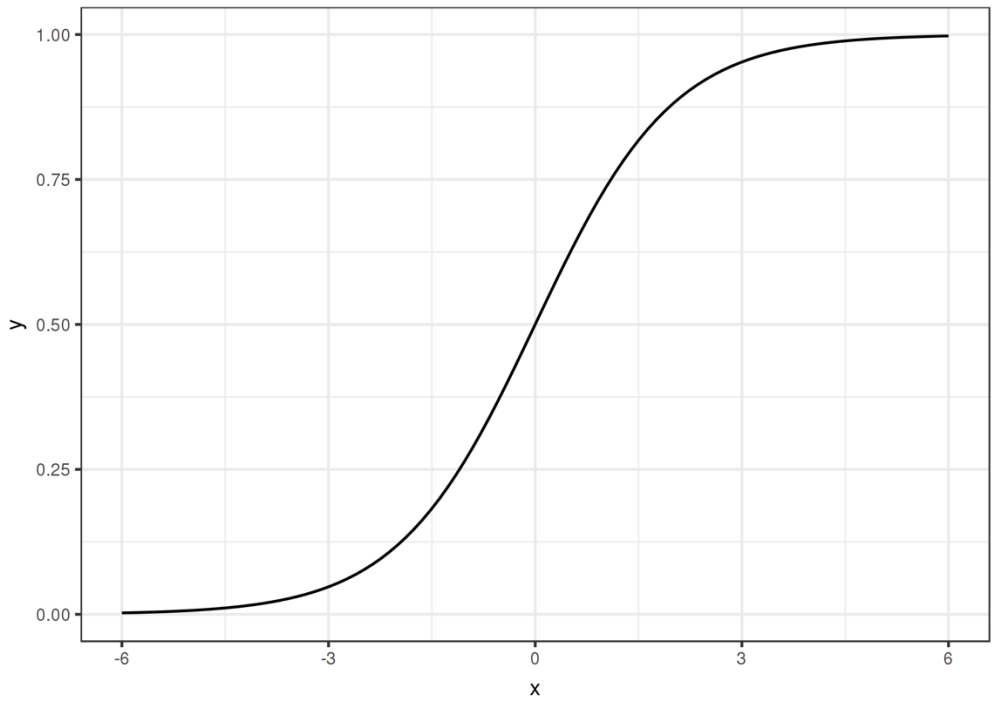


Figure 3.3: The logistic function. It outputs numbers between 0 and 1. At input 0, it outputs 0.5.

Logistic Regression by Stochastic Gradient Descent

The values of the coefficients are estimated using the Stochastic Gradient Descent by initially assuming the values of all the coefficients as 0.0

The new updated values of coefficients can be obtained by using a simple update equation:

$$b = b + \alpha \times (y - prediction) \times prediction \times (1 - prediction) \times x \quad \text{Equation 3.3}$$

Where *b* is the coefficient that is being updated and prediction is the output of making a prediction using the model. α is a parameter that must be specified at the beginning of the training run which is the learning rate and controls how much the coefficients change or learns each time it is updated. Good values for α might be in the range 0.1 to 0.3. The last term in equation 3.3 is *x* which is the input value for the coefficient. Since, b_0 does not have an input; its value is assumed to be 1.0.

The process of updating the values of regression coefficients is repeated for a fixed number of times, until a desired accuracy is obtained.

3.3. Source of Data

Source of data was secondary source and the data set was downloaded from kaggle.com and dataset was Pima Indians Diabetes Database (PIDD) [20] which contains 768 records of female patients.

3.4. Experimental Setup and Evaluation

Experiment had been done in python version 3.7.3 using sublime text.

Table 3.1: Experimental Parameters

Scheme1: Random Forest *Scheme2: Logistic Regression*
Relation: diabetes.csv
Test mode: split ratio 8:2

CHAPTER 4
EXPERIMENT AND RESULT ANALYSIS

4.1. Background

This section deals with the successful implementation and comparative analysis of random forest and logistic regression for diagnosis of diabetes mellitus. The experiments were performed in python version 3.7.3 using sublime text installed in system consist of 1.8 GHz Intel Core i5 processor with 8 GB RAM in macOS Mojave Operating System.

4.2. Tools

Algorithms can be compared using many data mining tools. However, in this research python version 3.7.3 had been used for simulation and following libraries are used:

- **NumPy (Numerical Python):** NumPy had been used to handle numerical data and arrays.
- **Pandas:** Pandas had been used for data manipulation like reshaping, splitting, aggregating and selecting data.
- **Matplotlib:** It had been used to visualize data.
- **Scikit-learn:** It had been used for data analysis features and selecting the classification models (Random Forest and Logistic Regression).

4.3. Data Samples

The data sample used in this study is shown below:

Table 4.1: Portion of dataset diabetes.csv

Pregnancy	Glucose	Blood Pressure	Skin Thickness	Insulin	BMI	Diabetes Pedigree Function	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1

4.4. Data Structure

The main data structures used in this study are enlisted below:

Table 4.2: Dataset Description

S. No.	Attributes	Attribute Type
1	Pregnancies	Numerical
2	Glucose	Numerical
3	Blood Pressure	Numerical
4	Skin Thickness	Numerical
5	Insulin	Numerical
6	BMI	Numerical
7	Diabetes Pedigree Function	Numerical
8	Age	Numerical
9	Outcome	Binary (0/1)

4.5. Experiments and Results

In this section, each steps of the methodology were implemented for simulation and results were described.

4.5.1. Experiments

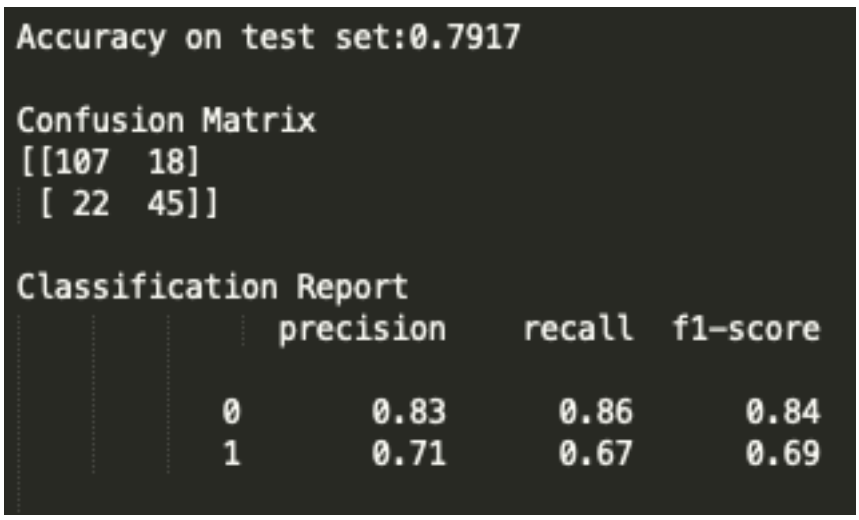


Figure 4.1: Result of Random Forest algorithm

Classification of large datasets is an important data mining methodology. For the purpose the most important figures here are the Accuracy. The output from the simulation in python is shown in the Figure 4.1 and Figure 4.2. In the output, Random

Forest was able to classify 79.17 % of the data correctly whereas Logistic Regression was able to classify 80.52% of the data correctly.

```
[15 rows x 9 columns]
Accuracy: 80.52%
confusion_matrix
[[97 10]
 [20 27]]

Classification report:
              precision    recall  f1-score

           0       0.83      0.91      0.87
           1       0.73      0.57      0.64
```

Figure 4.2: Result of Logistic Regression algorithm

4.5.2. Evaluation

For the comparison, two different classification algorithms are assessed using the following evaluation metrics.

- **Confusion Matrix**

A confusion matrix is a table for analyzing the result of the classifiers. It deals with how classifier can recognize tuples of different classes. In order to develop the confusion matrix, the following terms should be considered:

- **True Positive (TP):** Positive tuples that are correctively labelled by the classifier.
- **True Negative (TN):** Negative tuples that are correctly labelled by the classifier.
- **False Positive (FP):** Negative tuples that are incorrectly labelled as positive.
- **False Negative (FN):** Positive tuples that are mislabeled as negative.

		Predicted Class		
		Yes	No	Total
Actual Class	Yes	**TP**	**FN**	**P**
	No	**FP**	**TN**	**N**
	Total	**P′**	**N ′**	**P+N**

Figure 4.3: Confusion Matrix

- **Accuracy**

Accuracy of a classifiers on a given test set is the percentage of test set tuples that are correctly classified by the classifiers. It also refers to the recognition rate of the classifier that means how the classifier recognizes tuples of the various classes.

$$\text{Accuracy} = \frac{TP + TN}{P + N} \text{----------Equation 4.1}$$

- **Precision**

Precision refers to the measure of exactness that means what percentage of tuples labeled as positive are actually such.

$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{---------- Equation 4.2}$$

- **Recall**

Recall refers to the true positive rate that means the proportion of positive tuples that are correctly identified. It is also known as sensitivity of the classifier.

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{TP}{P} \text{---------- Equation 4.3}$$

- **F-Measure**

The F-Measure also refers to F_1-scorewhich combines both the measures i.e. Precision and Recall as the harmonic mean

$$\text{F} - \text{Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \text{---------- Equation 4.4}$$

4.5.3. Results

Table 4.3: Results of all algorithms

S.NO	Algorithms	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
1	Random Forest	79.17	77.0	76.5	76.5
2	Logistic Regression	80.52	78.0	74.0	75.5

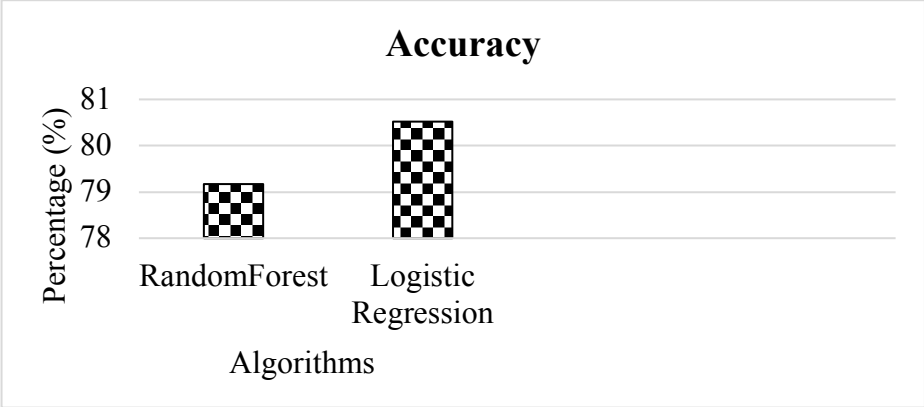


Figure 4.4: Graph of table 4.3 taking Accuracy

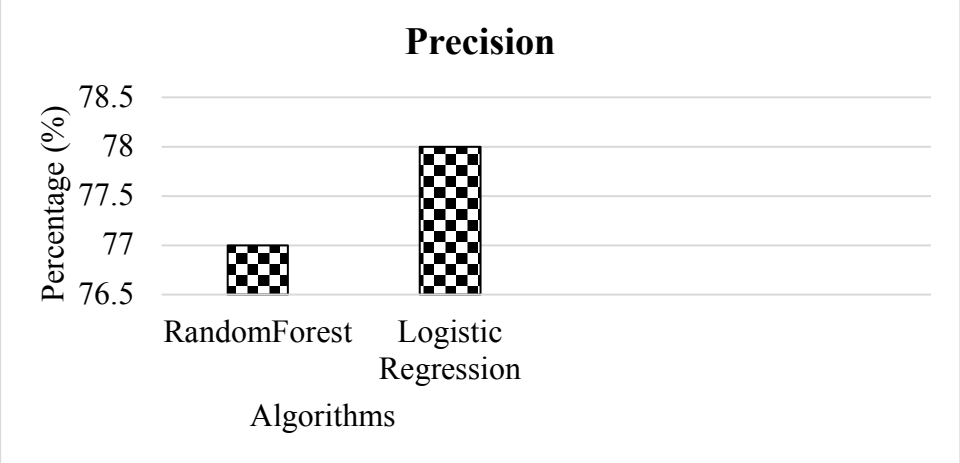


Figure 4.5: Graph of table 4.3 taking Precision

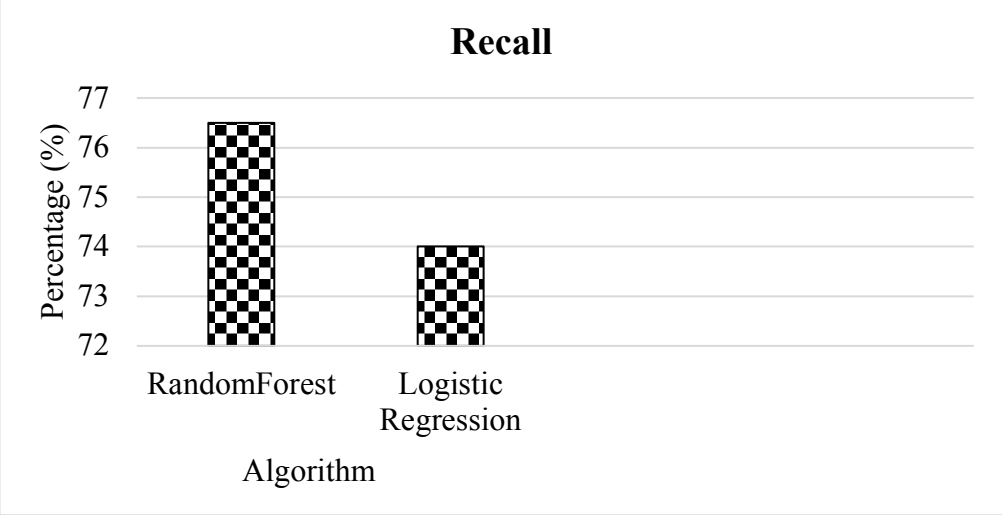


Figure 4.6: Graph of table 4.3 taking Recall

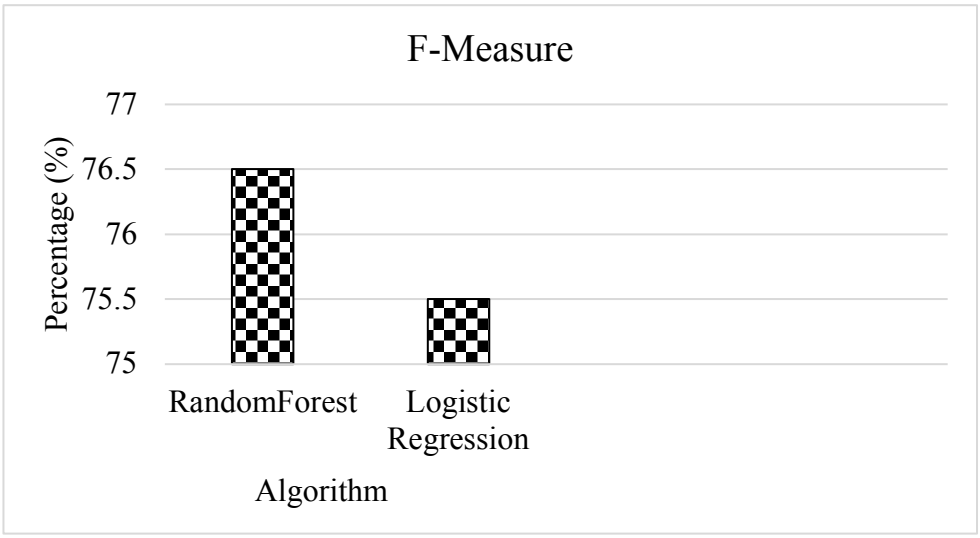


Figure 4.7: Graph of table 4.3 taking F-Measure

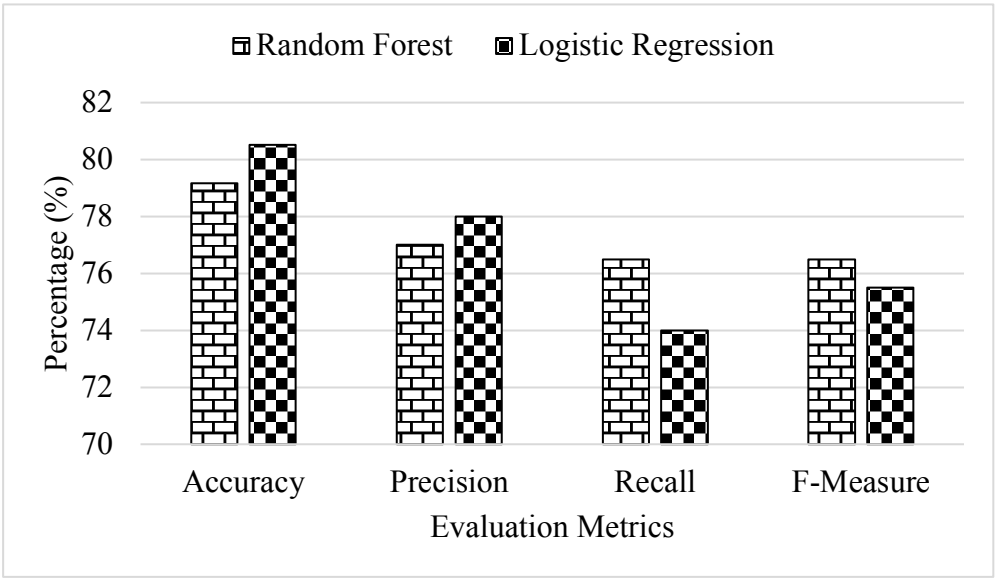


Figure 4.8: Graph of table 4.3 taking all evaluation metrics

4.6. Result Analysis

The table 4.3 and figures 4.4, 4.5, 4.6, 4.7 and 4.8 were results of the simulations, which demonstrated the performance of classification algorithm for the comparative analysis of random forest and logistic regression for diagnosis of diabetes mellitus.

Figure 4.4 showed that accuracy observed by implemented classification algorithms where it ranged from 79.17% to 80.52%. Among the algorithms Logistic Regression had got rich as well as motivating and encouraging result with 80.52% and Random Forest was less capable to classify with accuracy of 79.17%.

Figure 4.5 showed that precision observed by implemented classification algorithms where it ranged from 77.0% to 78.0%. Logistic Regression had got better precision level of 78.0% whereas Random Forest got less precision level of 77.0%.

Figure 4.6 showed that recall observed by implemented classification algorithms where it ranged from 74.0% to 76.5%. Random Forest had got encouraging recall of 76.5% whereas Logistic Regression got minimum recall of 74.0%.

Figure 4.7 showed that F-measure observed by implemented classification algorithms where it ranged from 75.5% to 76.5%. Again, Random Forest had got victory over Logistic Regression with the value 76.5%.

Figure 4.8 showed that the comparison between all the evaluation metrics of the implemented algorithms and from that comparison; Logistic Regression produced better classification result regarding accuracy and precision i.e. 80.52% and 78.0% respectively whereas Random Forest had got a better classification regrading recall and F-measure as 76.5% and 76.5% respectively.

CHAPTER 5
CONCLUSION AND FUTURE WORKS

5.1. Conclusion

The comparison of classification algorithm is a complex task and it is an open problem. For the best classification algorithm, it requires to make several methodological choices. So, this research focused in the comparative analysis of random forest and logistic regression for diagnosis of diabetes mellitus.

From the result analysis, it was seen that Logistic Regression was able to classify 80.52% of the data correctly which was better than Random Forest in comparison to results of evaluation metrics. In a nut shell, the experiment result showed that Logistic Regression had got 1.35% better accuracy than Random Forest for the diagnosis of diabetes mellitus.

5.2. Future Works

Directions for future works are:

- One important area for improvement is performance (Accuracy).
- Another is enhancing the performance (Accuracy) more by implementing other classification algorithms.

REFERENCES

[1] J. Han, M. Kamber, and J. Pei. "Data mining: concepts and techniques, Waltham, MA." *Morgan Kaufman Publishers* 10 (2012): 978-1.

[2] T. M. Lakshmi, A. Martin, R. M. Begum, and V. P. Venkatesan. "An analysis on performance of decision tree algorithms using student's qualitative data." *International Journal of Modern Education and Computer Science* 5, no. 5 (2013): 18.

[3] A. Wålinder. "Evaluation of logistic regression and random forest classification based on prediction accuracy and metadata analysis." (2014).

[4] "AI Horizon: Introduction to Machine Learning", *Aihorizon.com*, 2019. [Online]. Available: http://www.aihorizon.com/essays/generalai/supervised_unsupervised_machine_learning.htm[Accessed: 02- Jun- 2019].

[5] A. Papagelis, D. Kalles, "Breeding Decision Trees Using Evolutionary Techniques."In *ICML*,vol 1, pp. 393-400, 2001.

[6] J. R. Quinlan, "C4. 5: programs for machine learning." *Mach. Learn* 16, no. 3 (1993): 235-240.

[7] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan et. al., "Top 10 algorithms in data mining." *Knowledge and Information Systems* 14, no. 1 (2008): 1-37.

[8] R. Singh and D. Garg. "Hybrid Machine Learning Algorithm for Human Activity Recognition Using Decision Tree and Particle Swarm Optimization." *International Journal of Engineering Science* 8378 (2016).

[9] "Introduction to Diabetes", Ashfordstpeters.nhs.uk, 2019. [Online]. Available: http://www.ashfordstpeters.nhs.uk/introduction-to-diabetes. [Accessed: 09- May- 2019].

[10] K. Das and R. Behera, "A Survey on Machine Learning: Concept, Algorithms and Applications", *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 2, pp. 1301-1309, 2017.

[11] N. arun and R. Moungmai, "Comparison of Classifiers for the Risk of Diabetes Prediction", *7th International Conference on Advances in Information Technology*, pp. 132-142, 2015.

[12] R. Joshi and M. Alehegn, "Analysis and prediction of diabetes diseases using machine learning algorithm: Ensemble approach", *International Research Journal of Engineering and Technology*, vol. 4, no. 10, pp. 426-435, 2017.

[13] J. Kandhasamy and S. Balamurali, "Performance Analysis of Classifier Models to PredictDiabetes Mellitus", in *Procedia Computer Science*, 2015, pp. 45-51.

[14] S. Selvakumar, K. Kannan and S. GothaiNachiyar, "Prediction of Diabetes Diagnosis Using Classification Based Data Mining Techniques", *International Journal of Statistics and Systems*, vol. 12, no. 2, pp. 183-188, 2017.

[15] P. Sittidech and N. Nani-arun, "Random Forest Analysison Diabetes Complication Data", in *Proceedings of the IASTED International Conference*, Zurich, 2014, pp. 315-320.

[16] L. Tapak, H. Mahjub, O. Hamidi and J. Poorolajal, "Real-Data Comparison of Data Mining Methods in Prediction of Diabetes in Iran", Healthcare Informatics Research, 2013.

[17] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.

[18] L. Breiman, "Random forests." Machine learning 45, no. 1 (2001): 5-32.

[19] J. Brownlee, *Master Machine Learning Algorithms*. 2016.

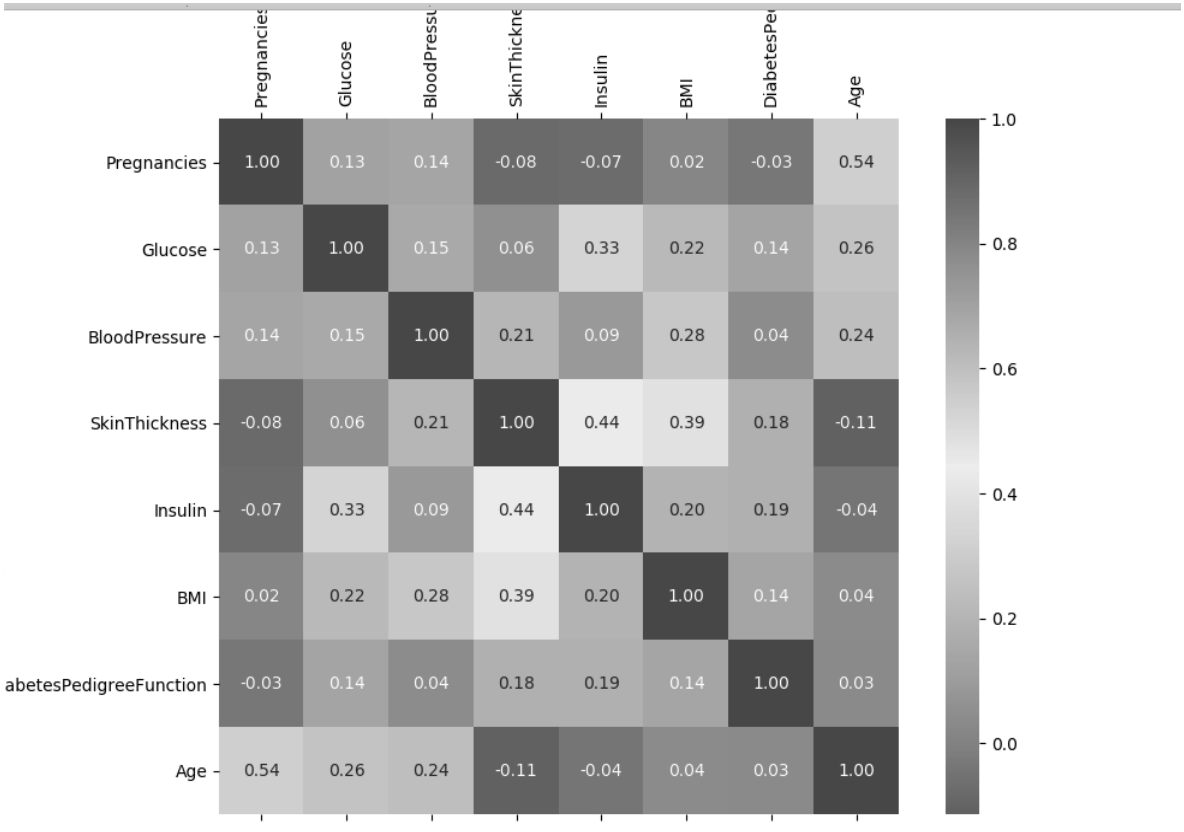
[20] "Pima Indians Diabetes Database", *Kaggle.com*, 2019. [Online]. Available: https://www.kaggle.com/uciml/pima-indians-diabetes-database. [Accessed: 16-Jan- 2019].

APPENDIX A
DATA VISUALIZATION

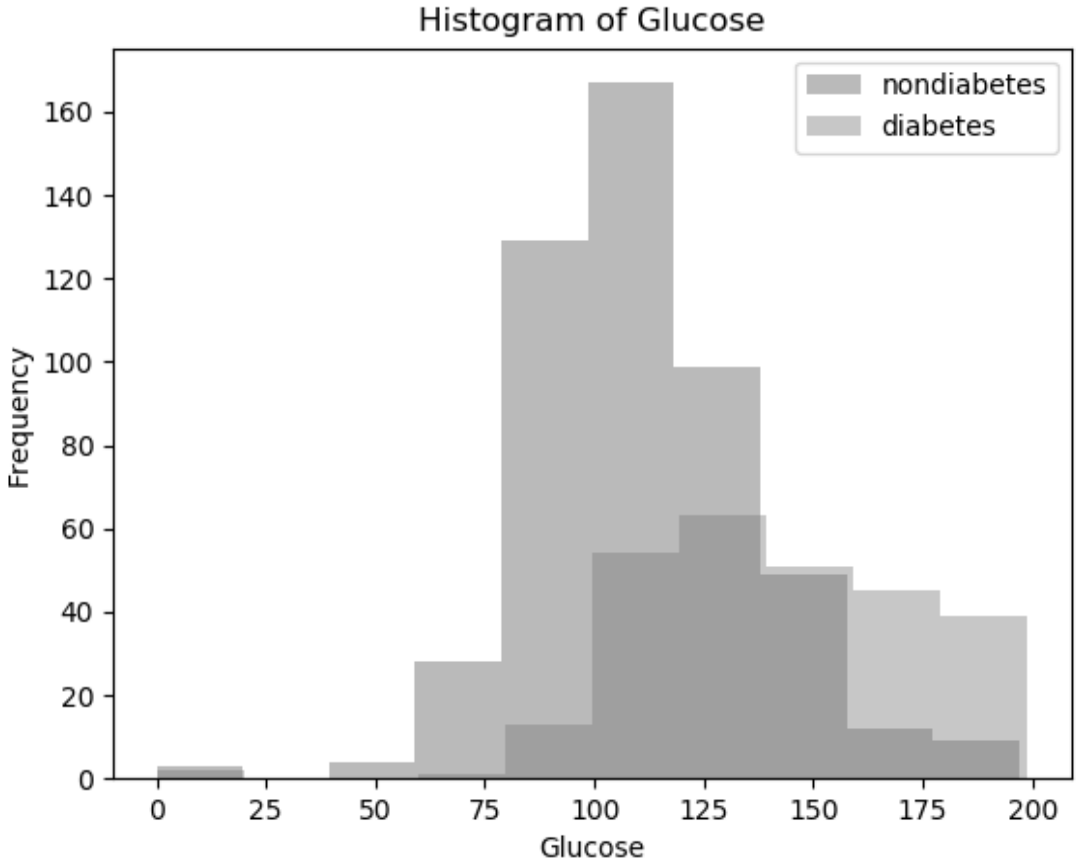
A.1 Instance of Data Sample

	Pregnancies	Glucose	BloodPressure	...	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.000000	...	0.627	50	1
1	1	85.0	66.000000	...	0.351	31	0
2	8	183.0	64.000000	...	0.672	32	1
3	1	89.0	66.000000	...	0.167	21	0
4	0	137.0	40.000000	...	2.288	33	1
5	5	116.0	74.000000	...	0.201	30	0
6	3	78.0	50.000000	...	0.248	26	1
7	10	115.0	72.405184	...	0.134	29	0
8	2	197.0	70.000000	...	0.158	53	1
9	8	125.0	96.000000	...	0.232	54	1
10	4	110.0	92.000000	...	0.191	30	0
11	10	168.0	74.000000	...	0.537	34	1
12	10	139.0	80.000000	...	1.441	57	0

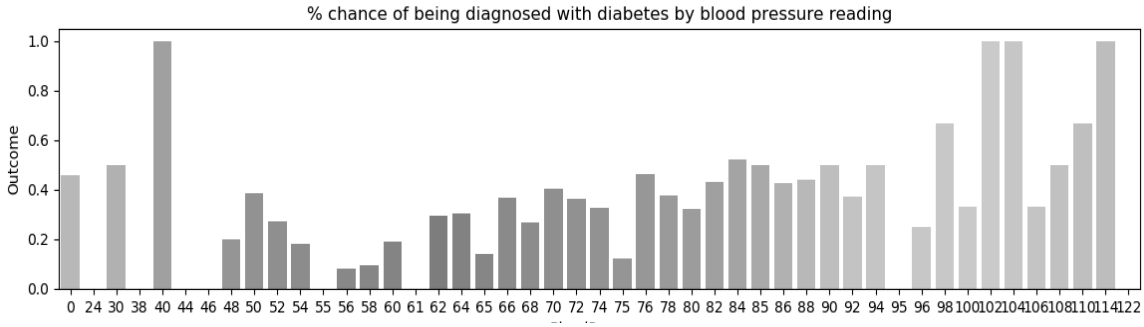
A.2 Attribute correlation



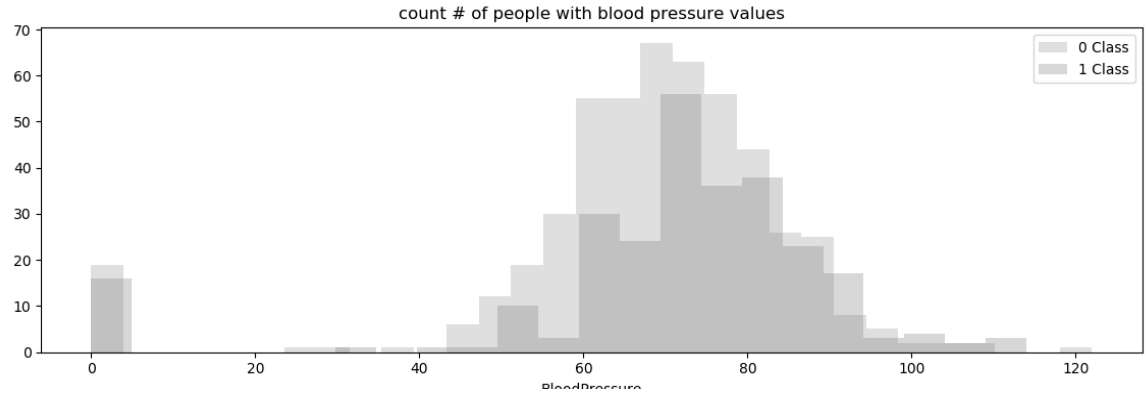
A.3 Histogram of Glucose



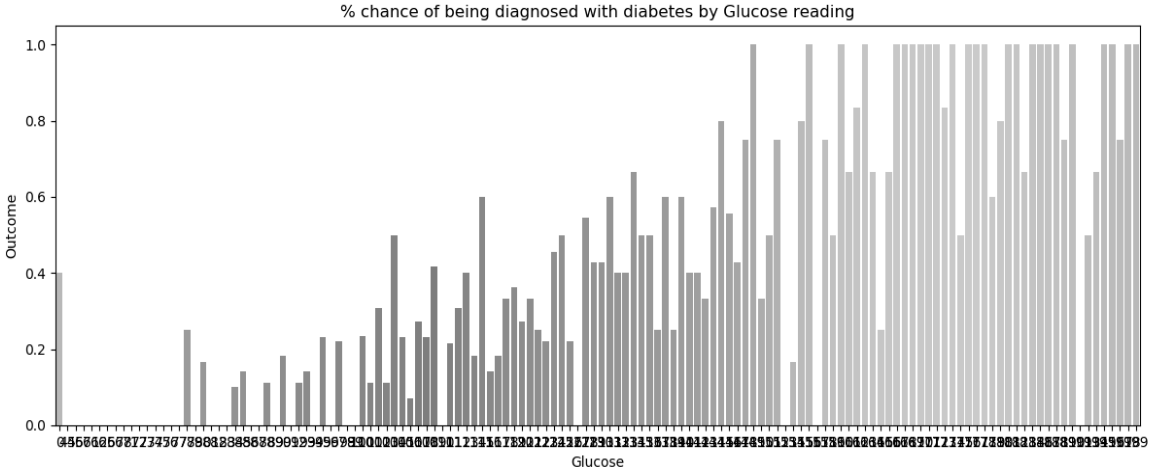
A.4 Percentage chance of being diagnosed with diabetes by Blood Pressure



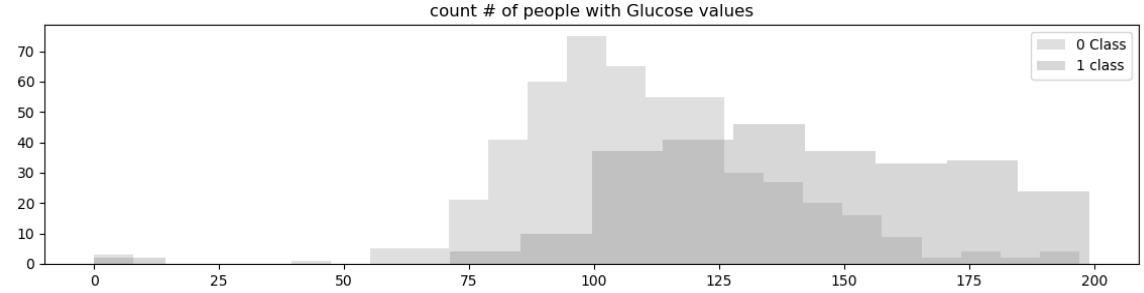
A.5 Count number of people with Blood Pressure values



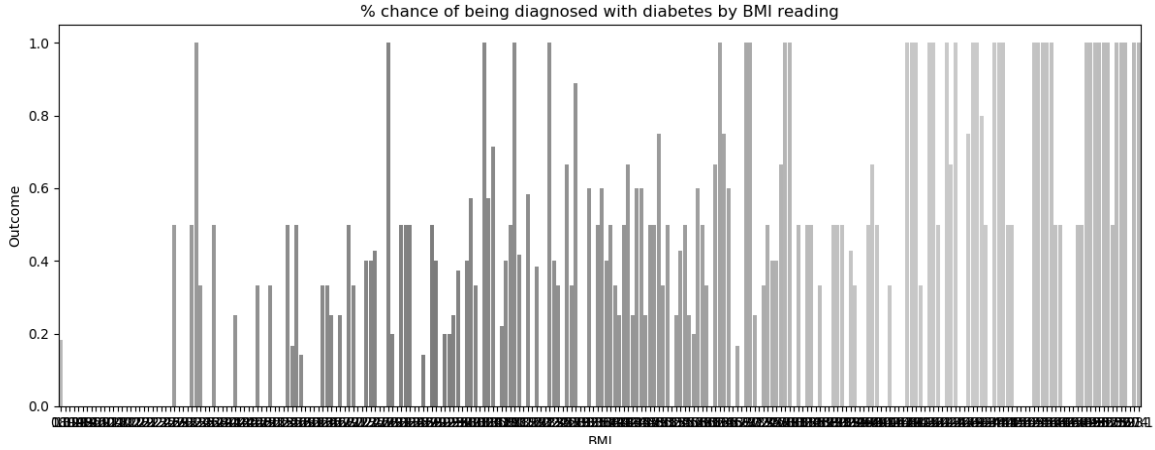
A.6 Percentage chance of being diagnosed with diabetes by Glucose reading



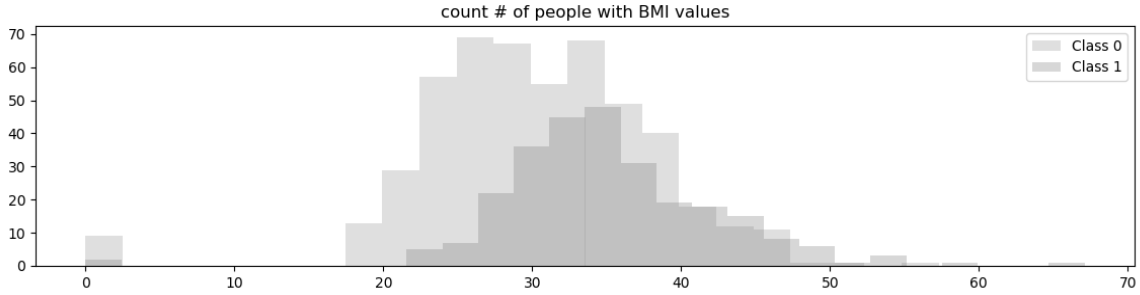
A.7 Count number of people with Glucose values



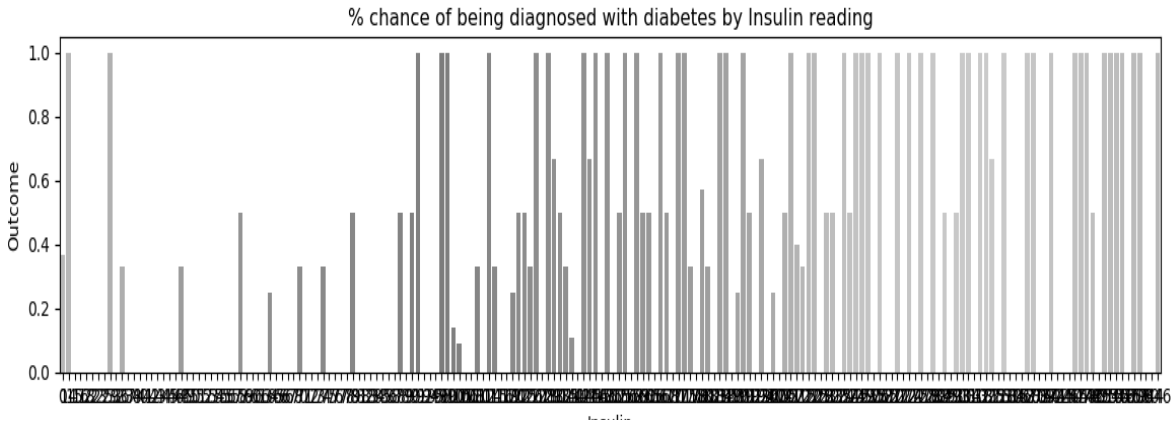
A.8 Percentage chance of being diagnosed with diabetes by BMI reading



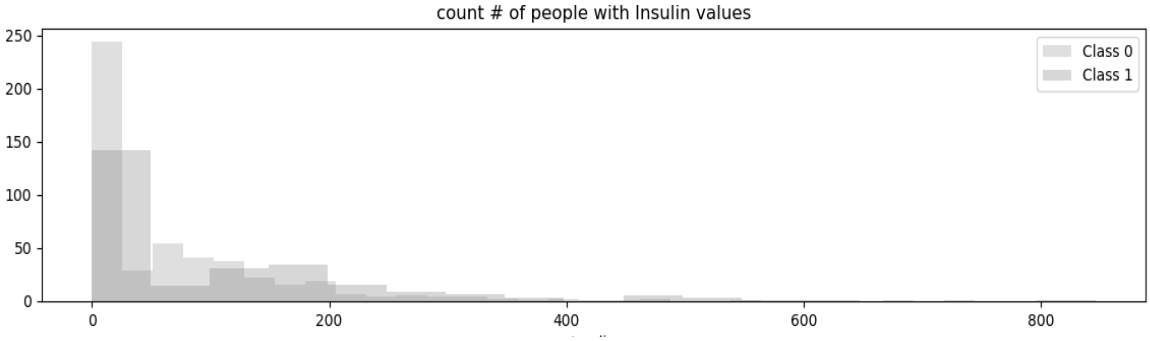
A.9 Count number of people with BMI values



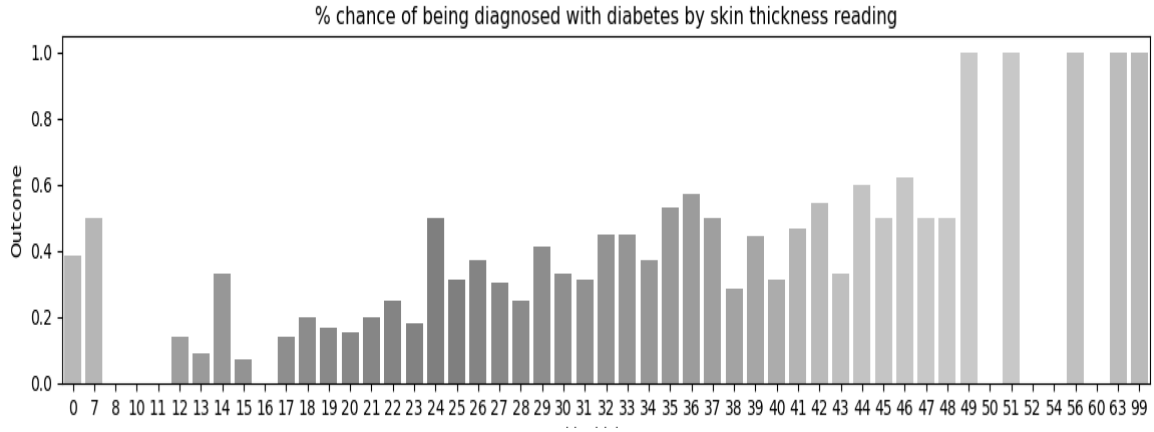
A.10 Percentage chance of being diagnosed with diabetes by insulin reading



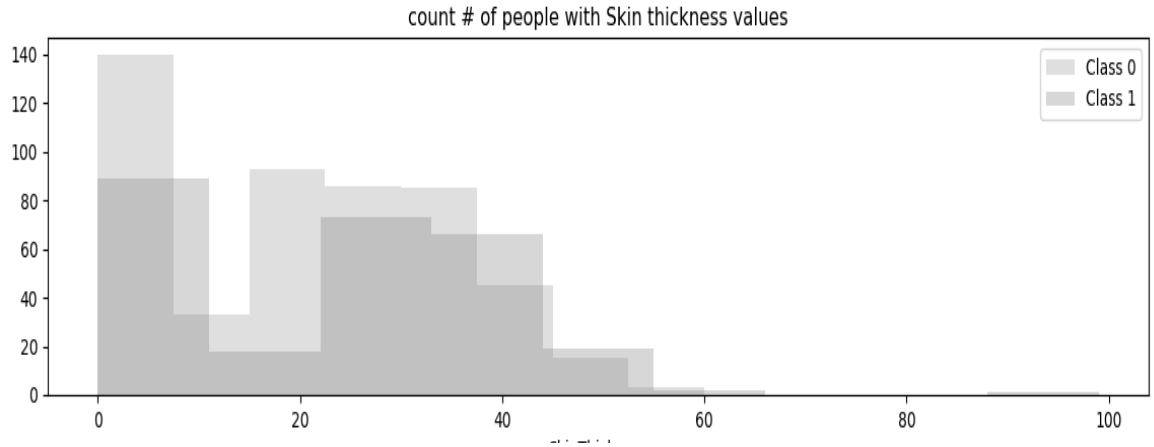
A.11 Count number of people with insulin values



A.12 Percentage chance of being diagnosed with diabetes by skin thickness reading



A.13 Count number of people with Skin thickness values



APPENDIX B
SOURCE CODE

B.1 Importing Libraries

```
import pandas as pd                                    #pandas is a dataframe library
import matplotlib.pyplot as plt                        #matplotlib.pyplot plots data
import numpy as np                                     #numpy provides N-dim object
support
from sklearn.metrics import accuracy_score, confusion_matrix,  classification_report
from warnings import simplefilter                      # ignore all future warnings
```

B.2 Dataset Splitting

```
from sklearn.model_selection import train_test_split
features_cols = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
'BMI', 'DiabetesPedigreeFunction', 'Age']
predicted_class = ['Outcome']

X = df[features_cols].values                           # Predictor feature columns (8 X
m)
Y = df[predicted_class]. values                # Predicted class (1=True, 0=False) (1 X
m)
split_test_size = 0.20
```

B.3 Dataset imputing with mean

```
from pandas import read_csv
import numpy
#dataset = read_csv('pima-indians-diabetes.csv', header=None)
# mark zero values as missing or NaN
df[['Glucose','BloodPressure','SkinThickness','Insulin','BMI']] =
df[['Glucose','BloodPressure','SkinThickness','Insulin','BMI']].replace(0, numpy.NaN)
# fill missing values with mean column values
```

```
df.fillna(df.mean(), inplace=True)
# count the number of NaN values in each column
        print(df.isnull().sum())
        print(df.head(15))
```

B.4 Data Visualization Code

```
plt.figure(figsize=(20,5))
glucose_pivot = df.groupby('Glucose').Outcome.mean().reset_index()
sns.barplot(glucose_pivot.Glucose, glucose_pivot.Outcome)
plt.title('% chance of being diagnosed with diabetes by Glucose reading')
plt.show()

plt.figure(figsize=(14,3))
glucose_pivot = df.groupby('Glucose').Outcome.count().reset_index()
sns.distplot(df[df.Outcome == 0]['Glucose'], color='turquoise', kde=False, label='0
Class')
sns.distplot(df[df.Outcome == 1]['Glucose'], color='coral', kde=False, label='1 class')
plt.legend()
plt.title('count # of people with Glucose values')
plt.show()

plt.figure(figsize=(20,5))
BMI_pivot = df.groupby('BMI').Outcome.mean().reset_index()
sns.barplot(BMI_pivot.BMI, BMI_pivot.Outcome)
plt.title('% chance of being diagnosed with diabetes by BMI reading')
plt.show()
```

B.5 Creating Random Forest

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

# do ploting inline instead of in a seperate window
```

```
#%matplotlib inline
#diabetes = pd.read_csv("/Users/madhupandey/Documents/dissertation/diabetes.csv")
                                                    # load Pima data. Adjust path as
necessary
xxx = df.isnull().sum()
        print('None of your data values is NULL', xxx)

X_train, X_test, y_train, y_test = train_test_split(df.loc[:, df.columns != 'Outcome'],
df['Outcome'], stratify=df['Outcome'], random_state=66)

diabetes_features = [x for i,x in enumerate(df.columns) if i!=8]

# Initial Trial
rf = RandomForestClassifier(n_estimators=100, random_state=0)
rf.fit(X_train, y_train)
        print("Accuracy on training set: {:.3f}".format(rf.score(X_train, y_train)))
        print("Accuracy on test set: {:.3f}".format(rf.score(X_test, y_test)))
```

B.6 Creating Logistic Regression

```
from sklearn.linear_model import LogisticRegression

# instantiate the model (using the default parameters)
logreg = LogisticRegression()

# fit the model with data
logreg.fit(X_train,y_train)
y_pred=logreg.predict(X_test)
```

B.7 Evaluating performance metrics

```
from sklearn import metrics
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train,y_train.ravel())
```

```
rf_predict_train = rf_model.predict(X_train)
        print("Accuracy on training set:
        {0:.4f}".format(metrics.accuracy_score(y_train,rf_predict_train)))
                print()
rf_predict_test = rf_model.predict(X_test)
        print("Accuracy on test
        set:{0:.4f}".format(metrics.accuracy_score(y_test,rf_predict_test)))
        print()

        print("Confusion Matrix")
        print(metrics.confusion_matrix(y_test, rf_predict_test) )
        print("")
#Classification Report
        print("Classification Report")
        print(metrics.classification_report(y_test, rf_predict_test))
```