A Major Project Report on

DIABETES DETECTION USING MACHINE LEARNING

BACHELOR OF TECHNOLOGY IN

INFORMATION TECHNOLOGY

During the year 2019-2023

By

G.JAYA PRAKASH	19017T1829
R.CHANDRA SHEKHAR	19017T1802
K.ROHITH	19017T1819
B.CHANDRA SHEKHAR	19017T1826
S.SAI DHRUVA TEJA	19017T1830

Under the guidance of

K.SRAVANTHI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY COLLEGE OF ENGINEERING,

KAKATIYA UNIVERSITY.

Kothagudem, Bhadradri kothagudem Dist,

Telangana-507101

UNIVERSITY COLLEGE OF ENGINEERING
BHADRADRI KOTHAGUDEM
A constituent college of KU,Warangal , Approved by AICTE, New Delhi
EAMCET Code:KUCE/KUCESF
ECET Code:KUCE/KUCESF

DEPARTMENT OF INFORMATION TECHNOLOGY

BATCH CERTIFICATE

This is to certify that the mini project work entitled **"DIABETES DETECTION USING MACHINE LEARNING"** is a bonafide work carried out by

1) G.JAYA PRAKASH	**19017T1829**
2) K.ROHITH	**19017T1819**
3) B.CHANDRA SHEKHAR	**19017T1826**
4) R.CHANDRA SHEKHAR	**19017T1802**
5) S.SAI DHRUVA TEJA	**19017T1830**

in partial fulfillment for the award of Bachelor of Technology in Department of Information Technology, University College of Engineering,Kakatiya University, kothagudem during the year 2019-2023 under my supervision and guidance. The result embodied in this project work done has not been submitted to any University or Institute for the award of any Degree or Diploma.

PROJECT GUIDE

K.SRAVANTHI

(Asst.Professor)

HEAD OF THE DEPARTMENT

Dr.T.ARCHANA

(Asst.Professor)

EXAMINER

ACKNOWLEDGMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success. They have been a guiding light and source of inspiration towards the completion of the project.

We would like to express our sincere gratitude and indebtedness toour project guide **Sravanthi Madam, Professor, Department of Computer Science**, who has supported us with her valuable suggestions and interest throughout our project with patience and knowledge.

We are also thankful to our **Head of the Department Dr. T.Archana** for providing excellent infrastructure and a conducive atmosphere for completing this project successfully.

We convey our heartfelt thanks to the lab staff for allowing us to use the required equipment whenever needed.

Finally, we would like to take this opportunity to thank our families for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

ABSTRACT

Diabetes Mellitus is among critical diseases and lots of people are suffering from this disease. Age, obesity, lack of exercise, hereditary diabetes, living style, bad diet, high blood pressure, etc. can cause Diabetes Mellitus. People having diabetes have high risk of diseases like heart disease, kidney disease, stroke, eye problem, nerve damage, etc. Current practice in hospital is to collect required information for diabetes diagnosis through various tests and appropriate treatment is provided based on diagnosis. Big Data Analytics plays a significant role in healthcare industries. Healthcare industries have large volume databases. Using big data analytics one can study huge datasets and find hidden information, hidden patterns to discover knowledge from the data and predict outcomes accordingly. In existing method, the classification and prediction accuracy is not so high. In this paper, we have proposed a diabetes prediction model for better classification of diabetes which includes few external factors responsible for diabetes along with regular factors like Glucose, BMI, Age, Insulin, etc. Classification accuracy is boosted with new dataset compared to existing dataset. Further with imposed a pipeline model for diabetes prediction intended towards improving the accuracy of classification.

Data analysis and machine learning libraries and algorithms are used for prediction on diabetes and information is shown in detail in the form of different types of graphs (histogram, density plots, box and whisker plots, and correlation matrixplots).

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

DM	Diabetes Mellitus
T2DM	Type 2 Diabetes Mellitus
IDDM	Insulin-dependent diabetes mellitus
RF	Random Forest
ANN	Artificial Neural Networks
WEKA (Tool)	Waikato Environment for Knowledge Analysis
ERH	Electronic Health Records
NIH	National Institute of Health

CHAPTER 1

INTRODUCTION

GENERAL

A report by the WHO (world health organization) as of Nov 2016, says that there are 422 million adults are with diabetes, 1.6 million deaths. In 2012 High Blood Glucose has been the cause of 2.2 million people deaths. Many diseases are caused due to Diabetes, and they affect our kidneys, eyes, heart and also other organs. To understand diabetes, we first need to learn how the body works without diabetes.

The food that we eat contains various kinds of components such as sugar, protein, fat, etc. The sugar we gain mainly comes from foods that contain carbohydrates which provides our body with energy. Foods such as bread, cereal, pasta, rice, fruit, dairy products, and vegetables contain carbohydrates. Such kinds of food, when consumed, are broken down into glucose by our body, and they are supplied throughout by the means of our bloodstream.

Mainly, glucose travels to the brain as it is required mainly for the body's thinking and functionality. The rest of the glucose is supplied to the rest of our bodies such as the cells, and the liver. Insulin is an important component that is required for the functionality of the human body. It is a hormone produced by beta cells in the pancreas. It permits the glucose to move from the bloodstream to the cells in our body. Since the pancreas is used to produce insulin, it needs enough glucose. If the pancreas cannot produce enough insulin, the glucose builds up and this is how diabetes is developed in an individual.

The signs or symptoms of Diabetes can be listed as Blurred vision, Fatigue, Weight Loss, Increased Hunger and Thirst, Frequent Urination, Confusion, Poor Healing, Frequent Infections, Difficulty in Concentrating.

Type 1 Diabetes: Type 1 diabetes occurs when our immune system destroys cells in your pancreas called beta cells, the cells that remake the insulin in our body. The insulin builds up in our blood and as a result, our cells are in a state of starvation which causes diabetes. It occurs usually in people less than 30 years and about 5 - 10% of those with diabetes but

can occur at any age. On the contrary to ancient belief, it is not a childhood disease. It happens to occur to adults more than children, although it was known to be a juvenile disease.

Type 2 Diabetes: People with type 2 diabetes make insulin, but their cells don't consume it as much as they should. The pancreas cannot keep up, and the sugar builds up in our bloodstream.

Data analytics is the identification of the hidden patterns from huge amounts of data for the drawing of conclusions. In health care, machine learning algorithms are used for analyzing the medical data to build machine learning models to carry out medical diagnoses. In this paper, we are going to use techniques such as Naive Bayes, Random Forest Algorithm and Logistic Regression to predict diabetes with the help of PIMA dataset.

PURPOSE

The aim of the project is to determine the appropriate classification model or algorithm that gives the best accuracy results ever possible. So, that algorithm proven to be the best can then be used in the prediction of diabetes to figure out if a person is diabetic or non-diabetic so far. This is to avoid any kind of misconceptions due to the incompetent classification algorithm or model can cause if the best one is not chosen.

It is also one of the most chronic diseases in India or elsewhere, the prediction of this in the early stage or even before should be able to control and contain it more easily maybe with a proper diet or a less severe treatment. The Type 2 diabetes has a much stronger link to family history or lineage than the Type 1 diabetes. So, if a member of a family has Type 2 diabetes it is likely that any member of the family could possess the same, so it has to be eliminated before it gets too complicated.

SCOPE

Type 2 diabetes is very different from Type 1 diabetes, which was previously called as insulin dependent diabetes mellitus (IDDM). Before the 2000s Type 2 diabetes was considered a disease of elderly and middle-aged individuals (hence it was also called adult-onset diabetes). But once it started to show up on teenagers the name faced away as it no

longer was confined to middle-aged or adults.

According to U.S. NIH, Type 2 diabetes contributes to the respective conditions directly:

1. Stroke and Heart diseases. Adults who are subjected to diabetes die due to heart diseases 2x to 4x times than those of adults who are not subjected to diabetes. The risk of the stroke the take place is also 2x to 4x times than those adults who are not subjected to diabetes.

2. Nervous system disease. Half of the population with diabetes feel impaired sensation, pain in the hands or feet, carpal tunnel syndrome, slower digestion, and many other nervous problems.

3. Possess High blood pressure. Most of the adults have blood pressure that goes higher than 130/80 mmHg.

4. Blindness. It is one of the new causes of being diabetic for the ones who are between the ages of 20 to 74.

5. Amputation. 60% of this occurs among the people with diabetes, non-traumatic lower limb amputation.

6. Kidney disease. One of the leading cause of kidney failure. About 150,000 individuals having diabetes survive on chronic dialysis or due to a kidney transplant.

7. Immune system disorder. Individuals with diabetes have fewer abilities to fight or reject viral and bacterial infections. They have more chances of dying of influenza or pneumonia from the individuals who do have diabetes.

8. Pregnancy complication. Mothers having diabetes having a greater number of abortions, and their babies tend to have a greater risk of major born defects and of being susceptible to diabetes later in their life.

Diabetes is a disease that is considered to be as one of the leading causes of death In India, 72 Million diabetes cases were recorded in 2017 and this is expected to double by 2025. This poses a serious public Health Issue in a country where population keeps increasing

exponentially every year. Among the Indian states, Tamil Nadu has been having the highest Death Rates from Diabetes. Diabetes often leads to long term disabilities and complications. It leads to Heart Attacks, Kidney Failure, Blindness and Gestational Diabetes causes birth defects to the new born babies.

Around 1.95 Lakh Crores will be needed as the Annual Cost to treat Diabetes. Urban Poor in India spend 34% of their income on Diabetes Treatment.

These trends indicate that there is a rise in premature death and this is a major threat to global development. Technological advancements have been useful in reducing hyperglycemia. But irrespective of all of these Technological Advancements, Diabetes still poses a serious threat to life.

We aim to perform Prediction and analysis on a PIMA Dataset that can be used to find the efficiency and accuracy. This can be used to find the most suitable algorithm and the one that has the highest accuracy. We split the dataset into 4 different splits namely: 60 / 40, 70 / 30, 80 / 20, 65 / 35. The Input Training and Test Data is fitted to the model and we then classify the Training data into different arrays for the purpose of prediction. We then find the accuracy by comparing the predicted values with the original set of values that we have.

MOTIVATION

Diabetes is indeed one of the chronic health problems that are devastating and with preventable consequences. The driving agents would be high blood glucose levels due to low insulin production. Type 2 diabetes affects men and women proportionately, around 12 million men and 11.5 million women have diabetes. To improve the quality of life means to take one’s own diabetes into control, for which additional support and education need to be provided to the patients.

Though the technology has evolved and new treatments are found in controlling diabetes, the challenges of self-comprehension are the most overwhelming for most of the individuals. It demands individual patient self-management that includes monitoring the blood glucose levels, maintaining a healthy diet, taking medication and regularly exercising. There are high non-compliance patterns in self-management behaviors, this could be usually due to the changes that are encountered in the patient's routine life. To

adapt and inherit to such changes the patients are usually motivated to achieve their goals and them their new way of living to create a long living life which allows them to manage diabetes. The support, assistance, and feedback play a very important role in achieving self-management goals. Organizations, where there are peer diabetes people support groups, are a valuable source for the patients as they can cling on to the mutual changes and awareness of themselves.

PROBLEM STATEMENT

Around 50.9 Million People in India suffer from diabetes and Tamil Nadu stands second in the list of Indian states that has the largest number of diabetes patients. The main objective of this project is to develop prediction modelling of the given medical data of patients with and without diabetes. We aim to predict and analyze the best algorithm that is suitable for Diabetes prediction and also find the efficiency. The Pima Indian dataset was used for this study and then analysis was done with data mining techniques.

It is the data obtained from the National Institute for Diabetes. Contains of several medical predictor variables and one target variable. The various medical variables are BMI, Glucose levels, Blood Pressure .etc. It contains 768 rows and 9 columns. The dataset file is in a .csv(Comma Separated Values) format. Using the help of Python's inbuilt library Pandas, which is a dataframe library, we import the file into our Python environment.

We are splitting the Dataset into:

1) Training Set and 2) Testing Set

and the performing analysis on them. We use algorithms such as Naïve Bayes, Random forest in order find the accurate method of prediction and the efficiency of these algorithms. Thereby, this early prediction can result to an early diagnosis which helps in curing a patient.

INTRODUCTION TO NAÏVE BAYES

The Bayesian Network plays an important part of machine learning in classification or prediction of diabetes. The most commonly used type of Bayesian Network for classification is the Naïve Bayesian's, which has the highest accuracy value of up to 99.51% respectively. The Bayesian Network applies the Naïve Bayes theorem which firmlyassumes that the presence of any particular attribute in a class is not related to the presence

of any other attribute, making it much more advantageous, efficient and independent.

The Bayesian Network is one of the most used techniques in the classification of diabetes, which has an accuracy in the range of 71% to 99.51%. The Naïve Bayesian is based on the conditional probability (given a set of features, the probability of a certain results occurrence):

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}, \text{ Where } X = (x_1, x_2, x_3, x_4, \ldots\ldots\ldots\ldots x_n)$$

$$P(y|x_1, \ldots, x_n) = \frac{P(x_1|y)P(x_2|y)\ldots P(x_n|y)P(y)}{P(x_1)P(x_2)\ldots P(x_n)}$$

$$P(y|x_1, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1)P(x_2)\ldots P(x_n)}$$

$$P(y|x_1, \ldots, x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y)$$

$$y = argmax_y P(y)\prod_{i=1}^{n} P(x_i|y)$$

Fig. 1.1 : Naive Bayes Formula

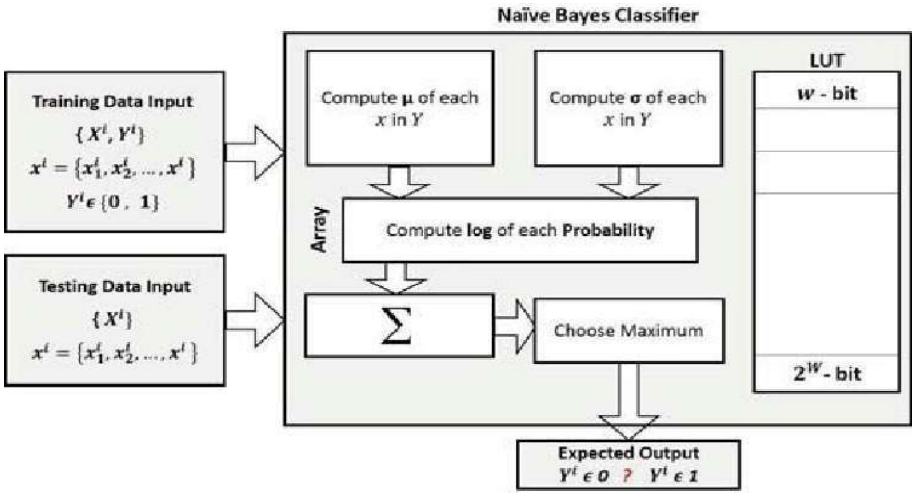

Fig 1.2: Naïve Bayes Architecture

Naive Bayes classifiers can handle a subjective number of autonomous factors whether nonstop or all out. Given a lot of factors, X = {x1,x2,x...,xd}, we need to build the posterior probability for the occasion Cj among a lot of conceivable results C = {c1,c2,c...,cd}. In an

increasingly well-known language, X is the indicators and C is the arrangement of absolute dimensions present in the needy variable. Utilizing Bayes' standard:

$$p(C_j \mid x_1, x_2, \ldots, x_d) \propto p(x_1, x_2, \ldots, x_d \mid C_j) p(C_j)$$

where p(Cj | x1,x2,x...,xd) is the posterior probability of class enrollment, i.e., the probability that X has a place with Cj. Since Naive Bayes expect that the restrictive probabilities of the autonomous factors are factually free we can decay the probability to a result of terms:

$$p(X \mid C_j) \propto \prod_{k=1}^{d} p(x_k \mid C_j)$$

and revise the posterior as:

$$p(C_j \mid X) \propto p(C_j) \prod_{k=1}^{d} p(x_k \mid C_j)$$

Utilizing Bayes' standard above, we name another case X with a class level Cj that accomplishes the most astounding posterior probability.

In spite of the fact that the presumption that the indicator (autonomous) factors are free isn't constantly exact, it simplifies the grouping task significantly, since it permits the class restrictive densities p(xk | Cj) to be determined independently for every factor, i.e., it lessens a multidimensional undertaking to various one-dimensional ones. Essentially, Naive Bayes lessens a high-dimensional density estimation undertaking to a one-dimensional part density estimation. Besides, the suspicion does not appear to extraordinarily influence the posterior probabilities, particularly in areas close choice limits, hence, leaving the grouping task unaffected.

Naive Bayes can be displayed in a few diverse ways including ordinary, lognormal, gamma and Poisson density functions:

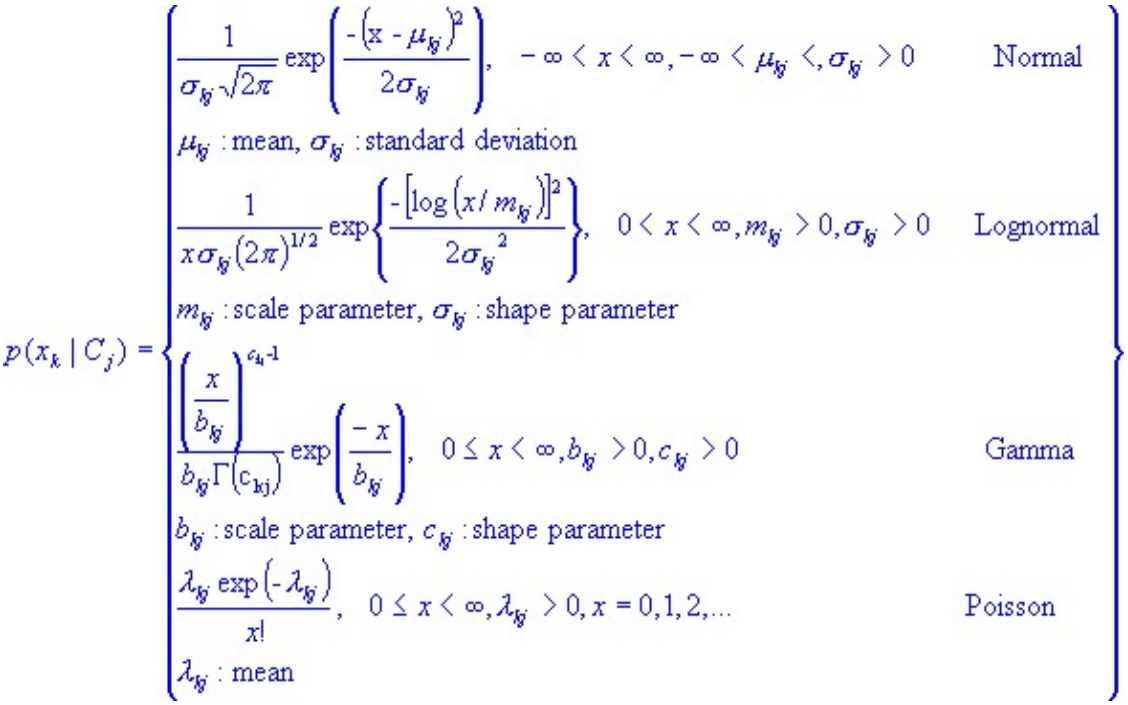

Fig 1.3: Poisson density functions

APPLICATIONS OF NAÏVE BAYES ALGORITHM

Naïve Bayes algorithm is a method that is commonly used for various analysis purposes. The applications of Naïve Bayes are as follows:

1. Since this method is mainly used in text classification, this algorithm is found to have a huge success rate when compared to other algorithm. This success rate makes it a very efficient algorithm for the prediction of diabetes and gives good percentages of accuracy while performing analysis.

2. It can be used for the purpose of spam filtering where all the spam emails in our inbox are stored in the spam folder. This helps in separating the important messages from messages that are sent for the intent of phishing, virus, etc.

3. Naïve Bayes is a very fast algorithm, so it can be used for the purpose of real time predictions. When it comes to diabetes prediction, the dataset is used to analyze and also predict if a person has diabetes or not.

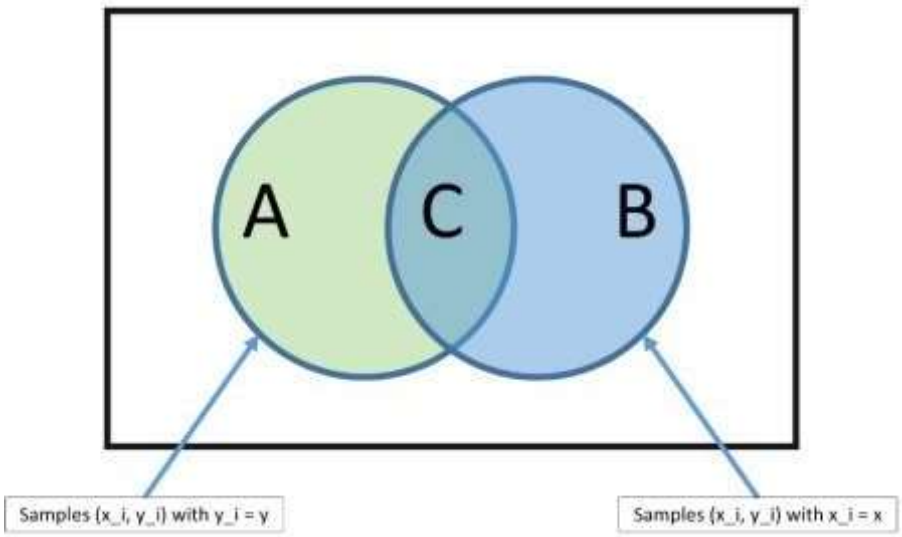

Fig 1.4: Naïve Bayes Classification

4. The posterior probability of many number of classes of a target variable in a dataset can be found using the help of Naïve Bayes.

$$\text{Posterior} = \frac{\text{Likelihood} * \text{Prior}}{\text{Evidence}}$$

$$P(C_j \mid A_1, A_2, \ldots, A_n) = \frac{\left(\prod_{i=1}^{n} P(A_i \mid C_j)\right) P(C_j)}{P(A_1, A_2, \ldots, A_n)}$$

Fig: 1.5: Posterior Probability

5. It is used in the credit analysis to analyze the payment probability of a loan in the financial world.

6. It is used for the treatment detection in the medical field where this can be used to find the probability of the treatment's effect on a person and which treatment is suitable for a particular disease.

7. It is also used for the categorization of news ie. Organizing the news according to its category which helps us to view news related to a certain topic.

ADVANTAGES OF NAÏVE BAYES ALGORITHM

Some of the advantages of Naïve Bayes algorithm are:

1. It is a very simple algorithm which is easy to implement and use for the purpose of analysis.

2. It works great for the purpose of practice even if the assumption does not hold.

3. Since we are splitting the data into training and testing data, we do not need much of training data for the purpose of analysis since less data is sufficient for analyzing the dataset. There will be no change in the pattern of prediction.

4. It can be used for the purpose of binary as well as multi-iclass classification.

5. Probabilistic predictions can also be performed with help of Naïve Bayes. It is the process of finding all possible probabilities of future outcomes in a given situation we perform analysis on. Naïve Bayes is apt for the analysis of future predictions.

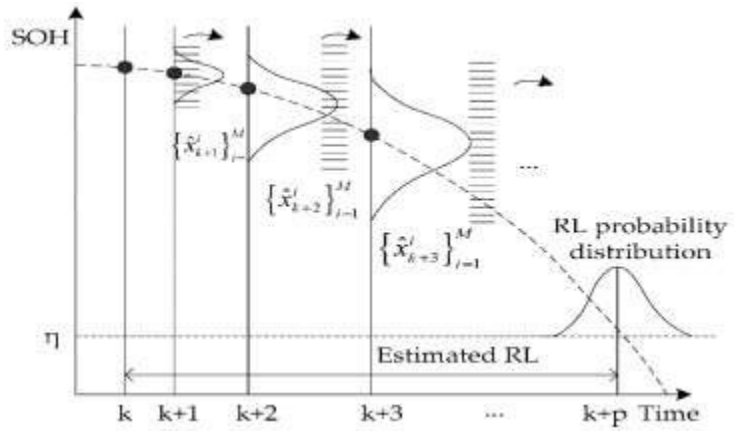

Fig 1.6: Probabilistic Prediction

6. It can be used to handle discrete as well as continuous data.

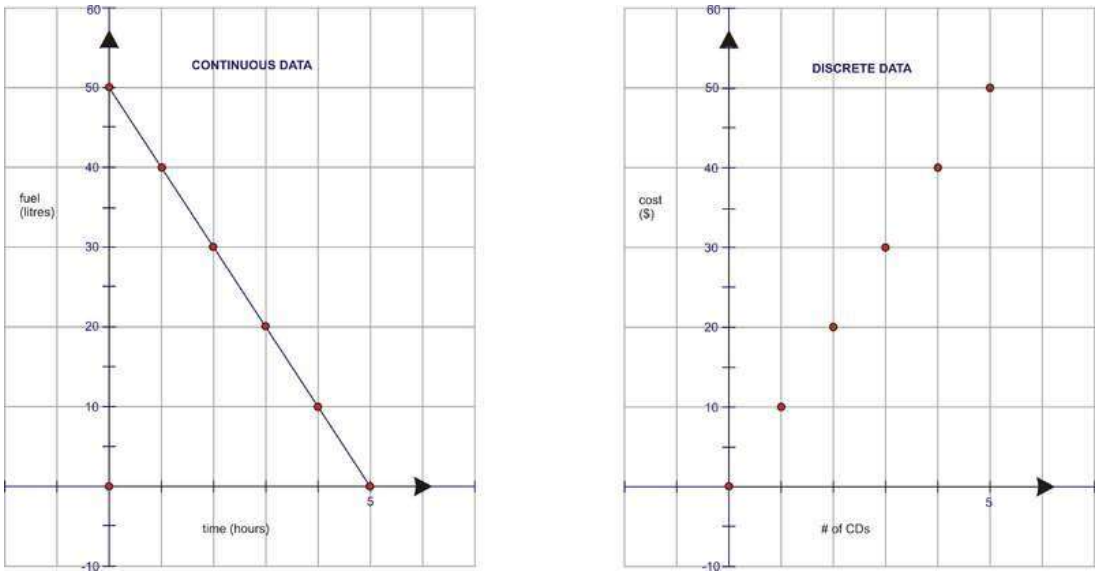

Fig 1.7: Discrete vs. Continuous Data

CHALLENGES OF NAÏVE BAYES

Even if the Naïve Bayes classifier can be an efficient method in prediction, there are also some challenges that we face while using this method.

1. According to the shape of our data distribution, it can make a very strong assumption. Because of the strong assumption made, the results that are generated can be bad.

2. Data scarcity is possible when it comes to Naïve Bayes. Since data scarcity happens, the results reflected may be inclined to either 0 or 1, which in turn gives bad results while performing analysis. With the use of sklearn, we can smooth the probabilities that are being predicted.

3. If there is a continuous variable present in the dataset that we are performing the algorithm on (For eg: time), it will be tedious to apply Naïve Bayes directly on the dataset. The predictions that we get after performing analysis will not be 100% accurate in this case.

4. If the number of classes are greater than 100K, then the scaling will not be possible.

INTRODUCTION TO RANDOM FOREST

The Random Forests algorithm is a powerful classification algorithm that can classify large amounts of data with high accuracy. Random Forest is a group learning method (it's a form of the nearest neighbor predictor) for classification and regression that build a number of decision trees at training time and display the class that is the mode of the classes output by individual trees.

They are a combination of tree predictors where each tree depends on the values of a random vector sampled independently with the same distribution for all trees in the forest. Random Forests solves this problem of high variance and high bias by finding a natural balance between the two extremes. They also have a mechanism to estimate the error rates (Out of the Bag error).

Many machine learning models, like linear and logistic regression are easily impacted by the outliers in the training data. Outliers are changes in the system behavior and can also be caused by human error, instrument error. There are chances for a given sample to be contaminated. These outliers or extreme values do not impact the model performance/accuracy. RF Algorithm overcomes and solves this problem.

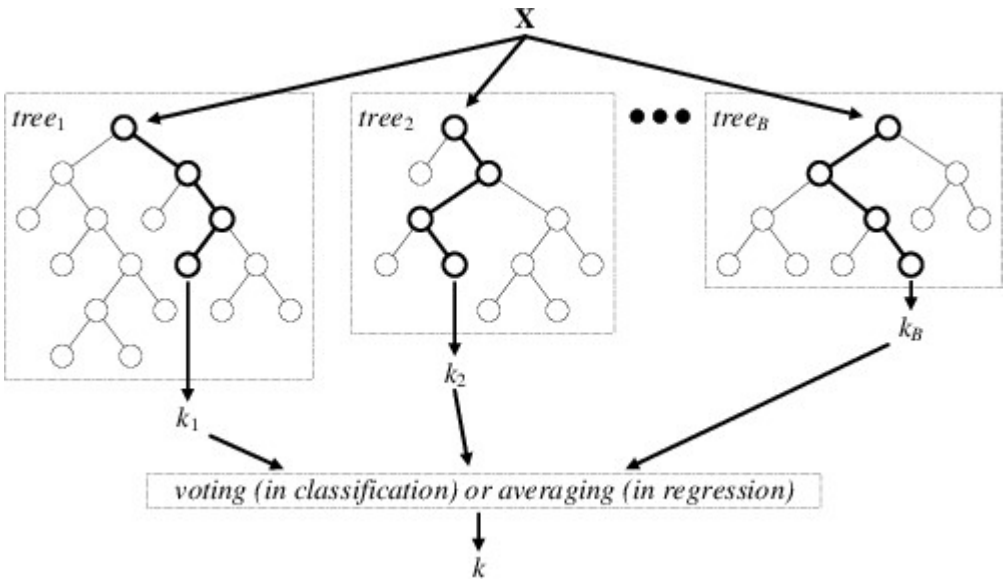

Fig 1.8: Random Forest Architecture

APPLICATIONS OF RANDOM FOREST

There are various applications in which Random Forest is used for the purpose of predictions and analysis.

1. Random forest can be used for the purpose on internet traffic interception where governments of certain countries want to block certain websites from the internet. Random Forest classifies the websites based on the features that are inputted and then the classification process finds the websites in order to block them. This should be done only when the data is unstructured so that the websites can be found easily.

2. When a video is uploaded on an online website such as Youtube, the video is directly out under a category which enables us to get the video when searched for using the category as a keyword. Random Forest helps in the direct categorization of videos while uploaded to the internet.

3. Random Forest can be used for the purpose of face recognition. The algorithm is trained by inputting large number of pictures and making the algorithm understand what the image depicts. For example, if the machine has to identify if it is a cat or a dog, the machine will learn for the large amount of images that is used for training and will identify the entity in the image.

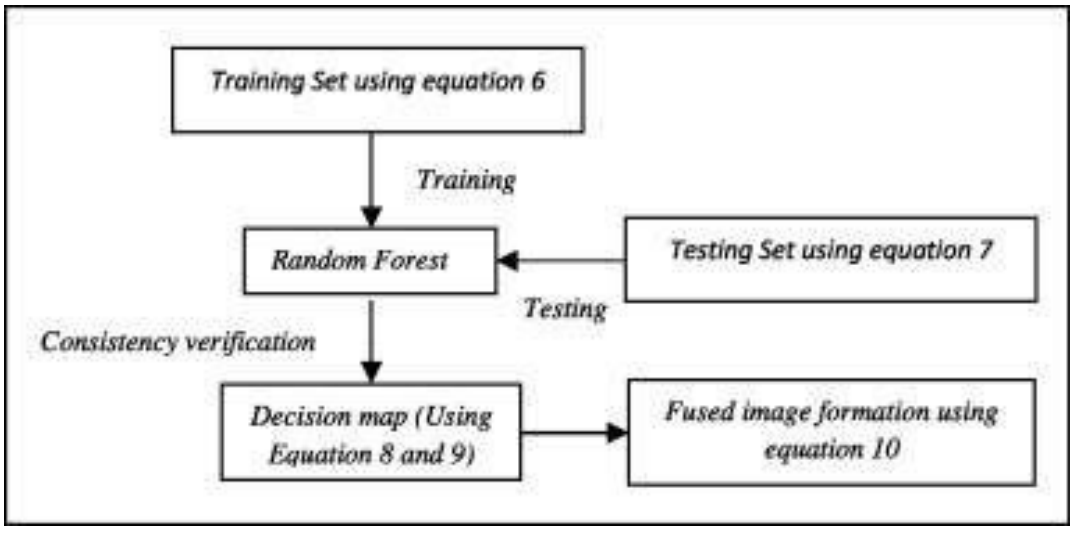

Fig 1.9: Face Recognition using Random Forest

4. It can also be useful in the process of voice recognition. Random Forest algorithm helps in the identification of voices where it is trained using various voice clips and

it's able to identify the owner of the voice that is being played. This is useful in the application of Siri in iPhones.

5. Random Forest can be used for the purpose of object detection. We train the algorithm to identify an object by training it with the help of images to identify the object in terms of its every angle.

6. Random Forest can be used for the purpose of human detection. The human detection is performed by the algorithm by identifying if a human is present or not. This is done with the help of a camera application where the algorithm detects the presence of a person in the frame.

7. Random Forest can be used for the prediction of diabetes and also in other fields of medicine.

8. Random Forest can be used for the purpose of stock market predictions.

9. In the field of e-commerce, Random Forest plays an important role of performing analysis and predictions on the applications that are under e-commerce.

ADVANTAGES OF RANDOM FOREST

Random Forest is advantageous in many ways in many fields.

1. Random forest can balance the errors in unbalanced datasets. This helps in the reduction of errors while performing analysis.

2. It computes prototypes that provides information between the classification of the variables and the variables itself. It classifies the variables that are important for the classification.

3. Larger datasets can be analyzed with the help of Random Forest as it tends to give better results.

4. Missing data in a dataset can be estimated with the help of Random Forest and also helps in maintaining the accuracy of the result while large sets of data are missing.

5. Variable interactions can be detected using experimental methods of Random Forest.

6. The learning of the algorithm is very fast and it trains very fast. If we split the data into training and testing data, then the data is learned by the algorithm very quickly to the point where the accuracy of the results are efficient compared to any other machine learning algorithm.

7. It does not delete any variables while handling thousands of variable inputs. This helps in better accuracy of the prediction results.

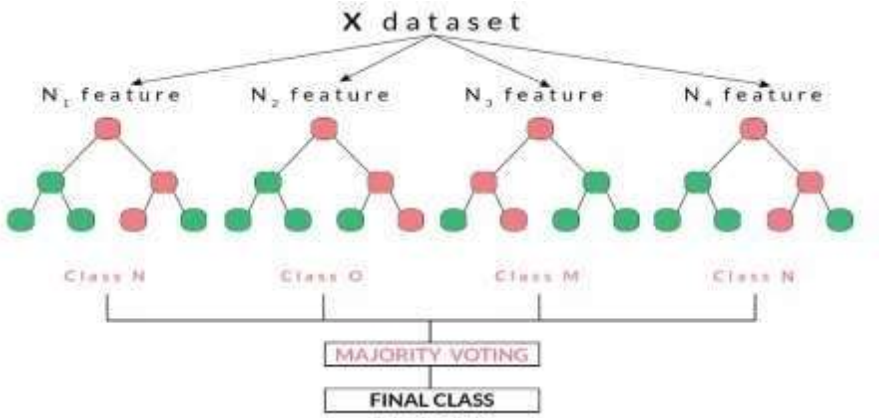

Fig 1.10: Random Forest Workflow

CHAPTER 2

LITERATURE SURVEY

1. A survey for detecting and predicting of diabetes using machine learning techniques [1]. This paper focuses on machine learning promising the improving accuracy of perception and diagnosis of the diseases. The various machine learning techniques that are used to classify the data sets include supervised, unsupervised, reinforcement, semi-supervised, and deep learning, evolutionary learning algorithms. It also shows the comparison of the two methods namely, Naïve Bayes and Artificial Neural Networks (ANN). The Bayesian Network applies the Naïve Bayes theorem which firmly assumes that the presence of any attribute in a class is not related to the presence of any other attribute, making it much more advantageous, efficient and independent.

2. They conducted a survey of data mining techniques for finding locally frequent diseases [2]. This paper focuses on mining the required medical data to find frequently occurring diseases, such as breast cancer, heart illness, lung cancer and so on. Data mining is defined as the process of digging in data for discovering latent patterns which can be converted into valuable information. Data mining techniques have been applied to many including Apriori and FPGrowth, linear genetic programming, decision tree algorithms, unsupervised neural networks, outlier prediction technique, classification algorithm, Naïve Bayesian and so on.

3. Aims on analyzing heart diseases using naïve bayesian algorithm [3]. The algorithm used here is naïve bayes, which firmly assumes that the presence of any attribute in a class is not related to the presence of any other attribute, making it much more advantageous, efficient and independent. The tools used is WEKA and classification is done by splitting data in to 70% of percentage split. The naïve bayes specified was able to produce 86.41% of the input data correctly and 13.58% of incorrect instances. He uses a dataset collected from a leading diabetic research institute in Chennai which has about 500 instances or patients.

4. One of the research presented an expert system for diabetes diagnosis [4]. Their

proposed system was rule based that uses IF- THEN. The divided the module into 3 stages: Block Diagram, Mockler Charts, Decision Tables. After considering many factors, this system provides4. Tawfik Saeed Zeki et al. [3] in their research presented an expert system for diabetes diagnosis. Their proposed system was rule based that uses IF- THEN. The divided the module into 3 stages: Block Diagram, Mockler Charts, Decision Tables. After considering many factors, this system provides diagnosis for diabetes. It was developed in VP-Expert.

5. There was a paper comparing different fuzzy expert systems by using multiple parameters for diabetes diagnosis [5]. MATLAB fuzzy logic toolbox was used for comparative study for both types of these expert systems. 5 parameters were used and results were generated.

6. In this paper they made an independent assessment, based on general characteristics of data [6]. The Methods which are a part of this approach are called filter methods and the feature set is filtered out before the model construction. They this were able to make the diabetes prediction algorithm using this. data mining and machine learning methods in Diabetes. They dwelled with biomarker identification and prediction of DM. Obesity is one of the major risk factors, specifically in T2D. Diabetes Mellitus (DM) diagnosis is taken forward by several tests such as random blood sugar test, hemoglobin(A1C), fasting sugar test. Calisir and Dogantekin for the prediction of DM proposed LDA-MWSVM, a system for the diagnosis of diabetes. The system performs attribute extraction and reduction via Linear Discriminant analysis (LDA) technique after which the classification using the Morlet Wavelet SVM (MWSVM) classifier. In the case of neuropathy similarly, DuBrava used Random Forest in order to choose specific attributes for prediction of diabetic peripheral neuropathy (DPN).

7. A paper focuses on detecting cardiovascular disease risk levels using Naïve Bayes classifier [7]. The characteristics of cardiovascular illness are identified through some primary risk factors such as diabetes mellitus, coronary artery function, kidney function and the level of lipids in the blood. Class labels were to be assigned according to the values of these risk factors: risk level 1, risk level 2 and so on. The evaluation of this method was done in three parameters namely, accuracy,

sensitivity and specificity. The proposed model delivered the class label of tuples above 80% correct. The experiment was conducted by variety of machine learning methods like naïve bayes, decision trees, classification or clustering and neural networks. The result showed that among all the tried methods naïve bayes has the highest accuracy rate amongst other algorithms.

8. The aim of one of the papers is to identify type 2 diabetes(T2DM) through electronic health records(ERH) [8]. To identify diverse genotype-phenotype associations affiliated with diabetes type 2 through phenome-wide association (PheWAS) study and genome-wide association (GWAS) study more cases and controls are to be identified (for example, via an Electronic Health Records). They urge to develop a semi-automated framework based using machine learning to liberalize the filtering criteria on improving the recall rate and keeping low false positive rate. They proposed a framework that identifies subjects with or without T2DM from ERH through engineering and machine learning. They evaluated and contrasted the identified performance of widely used machine learning models including Random Forest, Naïve Bayes, Logistic Regression, K-Nearest- Neighbor and Support Vector Machine. They had a sample of 300 patients randomly selected from 23,281 patients having diabetes retrieved from the ERH repository between 2014 to 2016.

9. Diabetes affected patients' classification via machine learning techniques [9]. The aim of this paper is to differentiate between the people affected with diabetes versus the ones that aren't affected by diabetes. They conduct hypotheses testing to verify whether the attributes are different people with diabetes and the ones who don't have diabetes. They used two methods to test for null hypotheses namely, Mann-Whitney (With p- level adhered to 0.05) and Kolmogorov-Smirnov (With p-level adhered to 0.05). They chose a 0.05 level of significance that is, they accept to make 5 mistakes out of 100. They used six classification algorithms namely, Hoeffding Tree, Random Forest, JRip, Multilayer Perceptron (deep learning algorithm), Bayes Network. The metrics they used to evaluate the result are Precision, Recall, F-measure and ROC area.

10. Prediction and analysis of diabetes using machine learning algorithms : an ensemble approach [10]. The use the PIMA dataset for their experiment that has about 768

rows of data or patients with about eight columns that are useful for the analysis and prediction if diabetes. The variety of methods or algorithms and statistical techniques used by them are Random Forest (RF), classification algorithm, K-Nearest Neighbour (KNN), Naïve Bayes. Classification algorithm like J48 which is an improvement to ID3 one of the classification algorithm. This can assist continuous as well as categorical instances to the process of tree construction. The aim was to predict the diabetes and also compare the models on which one provides a high accuracy. Than finally, select the best appropriate algorithm for the prediction of the diabetes disease at an early age.

11. Identifying diabetic Nephropathy using a Neuro-Fuzzy system [11]. Nephropathy can be easily controlled at if early recognition and treatment of renal changes. The fuzzy expert system was generated through seven inputs and one result (output). The inputs 1 to 3 are having three gaussian membership functions and input 4 to 7 are having two gaussian membership functions. The result has four membership functions of constant nature. In the table as the outcome, the result of 80 diagnosed patients, in the first column, 20 diagnosed patients are sorted as severe. In the second column, 20 diagnosed patient cases, 19 of which are sorted as moderate and 1 patients are sorted faultily. In the third column, 20 diagnosed patient cases, 18 of which are sorted as minor and 2 patients are sorted faultily. In the fourth column, 20 diagnosed patient cases, 18 of which are sorted as good and 2 patients are sorted faultily. Overall out of 80 diagnosed patients' cases, 75 diagnosed patients' cases are adequately sorted and 5 diagnosed patients are sorted inadequately.

12. Disease prediction using machine learning techniques [12]. The process of discovering useful information from backend that can meaningfully comprehend the data is known as Data mining, process of discovering intriguing patterns and knowledge from a wide variety of data. These techniques are a part of three main categories that include supervised, semi-supervised and unsupervised learning techniques. This paper proposes new hybrid method using Principal Component Analysis (PCA), Classification and Regression Trees (CART), Gaussian mixture model with expectation maximization (EM) and Fuzzy-rule based techniques. They then apply these models or algorithms on real-world datasets, which is obtained from the University of California, Irvine (UCI) called the Pima Indian Diabetes.

13. Multi-layer classifier for disease prediction [13]. This paper focuses on the mixture of heterogeneous classifiers for disease prediction and classification, hence overcoming the limitation of single or individual classifiers. The combination of classifiers is presented which is Naive Bayes, Quadratic Discriminant Analysis, Linear Regression, Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Decision Tree. The multiple layers are used at multiple layers to further enhance the level of disease prediction accuracy. An application has been developed for the disease prediction which is proposed on the HMV (Hierarchical Majority Voting) ensemble framework. The proposed ensemble consists of three modules that includes data acquisition and preprocessing, classifier training and HMV ensemble model for disease prediction and classification using three layered approach.

14. Medical expert system used for diabetes diagnosis [14]. This paper proposes a medical expert system for the identification or diagnosis of diabetes. The OWL format with 9 subclasses were used in developing the diabetes ontology. The in-between results with weighted OWA similarity model or algorithm are expressed for easy understanding by the users. The expert system was developed as a web-based application with a web service architecture. The results were, the overall consistency rate achieved was around 90.7% with 65 patients from the test data. The new-found results show that this system can be used to diagnose diabetes early and serve as a guide for people with diabetes to monitor the disease.

15. Performance analysis of the various classification algorithms [15]. This paper takes three different datasets with different number of features for three completely different diseases namely breast cancer, lung disease and heart illness. The majority of lung cancer is due to high exposure to tobacco smoking but 10-15% cases occur in non-smokers. The dataset put together consists of the genetic codes of the person or patient with or without cancer. The next major disease is the breast cancer and the dataset for this has different features computed from a digitized image in a fine needle aspirate (FNA) of breast mass. To diagnose if the cancer is benign or malignant they check the features of the cell nuclei present on the image. The thris disease is the heart illness, the dataset used in this helps us to diagnose if the they patient leads to a heart attack or not. The dataset ahs 76 features out of which only

4 were considered. All the three datasets were obtained from the UCI machine learning repository. The techniques used are logistic regression, random forest, classification and regression trees (CART), logistic regression.

CHAPTER 3

SYSTEM ANALYSIS

FUNCTIONAL REQUIREMENTS

In order for every software application to run properly, it needs to satisfy a lot of functions that are to be deployed in it. These functions are nothing but various operations that are performed in each step while developing the application. This step comes under the best practices of developing an application. Functional and Non-Functional Requirements together set a list of rules that govern the smooth running of an application and it also helps the developer and the user to determine the software and hardware requirements that are needed to run the application. Functional Requirements that are required are:

Python:

Python programming language was developed in the year 1991 by Guido Van Rossum. The syntaxes used in the language makes it very comfortable and easier for developers to work with. Because of this very reason, this programming language can be used both in small and large scale. They are dynamic and garbage collected.

Numpy:

Numpy is a universally useful array processing package. It gives an elite multidimensional cluster object, and devices for working with these arrays. It is the principal package for logical processing with Python.

Matplotlib:

Matplotlib is a stunning perception library in Python for 2D plots of arrays. Matplotlib is a multi-stage information perception library based on NumPy arrays and intended to work with the more extensive SciPy stack. It was presented by John Hunter in the year 2002.

NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are used to set conditions to monitor the performance characteristic of the application. It describes how a specific function in the application works. They also determine the overall quality of the project and hence it is a very important aspect in any software development process. The Non-Functional Requirements include

1. **Usability:** It refers to the easiness of the application of models and determines the ease with which it can be used by the user. Usability can be said to be high when the knowledge required to use the models is less and the efficiency of its functionality is high. It is also a main criterion which can determine the accuracy of the results.

2. **Accuracy:** Accuracy determines the relative closeness of the value produced by the system to that of the ideal value. It is also one way to determine how the classification models works better compared to the other similar models.

3. **Responsiveness:** Responsiveness is determined by completing the software operations with minimal errors or no errors. It is directly proportional to the stability and the performance of the application. The Robustness and Recoverability can also be determined by this criterion.

4. **Scalability:** Scalability is used to determine the growth of the project. It determines how much room the application can have in order to include more features in the future. It determines the sustainability of the project.

HARDWARE REQUIREMENTS

Processor: Intel I5 processor
Storage Space: 500 GB.
Screen size: 15” LED
Devices Required: Monitor, Mouse and a Keyboard
Minimum Ram: 8GB

SOFTWARE REQUIREMENTS

OS: Windows 7 and above /LINUX

Programming Language: Python

Software: Jupyter Notebook

Additional requirements: Numpy, Matplotlib

ISSUES IN EXISTING SYSTEM

The very beginning of figuring out diabetes or the traditional method to identify that a person has diabetes is that patients need to visit a diagnostic center, consult their doctor, and sit tight for a day or more to get their reports. Moreover, every time they want to get their diagnosis report, they have to waste their money in vain. But with the rise of Machine Learning approaches we have the ability to find a solution to this issue, we have developed a system using data mining which has the ability to predict whether the patient has diabetes or not. Moreover predicting the disease early leads to treating the patients before it becomes critical. Data mining has the ability to extract hidden knowledge from a huge amount of diabetes-related data.

The aim of this research is to develop a system which can predict the diabetic risk level of a patient with a higher accuracy. This research has focused on developing a system based on various classification methods namely, Naïve Bayes, Random Forest, Support Vector Machine algorithms etc. Generally in a large datasets, there are possibilities where one may find multiple outliers, But most of the times these outliers are not taken in to accountduring the training or testing the data.

This could be or cause an error on the accuracy of the prediction of diabetes. Whether to take these outliers into account or remove from the dataset is a topic to debate upon. If it is obvious that the outlier is due to incorrectly entered or measured data, we should dropthe outlier: If the outlier does not change the results but does affect assumptions, we may tend towards dropping the outlier.

Generally in a large datasets, there are possibilities where one may find multiple outliers, But most of the times these outliers are not taken in to account during the training or testing the data is one of the issues. This could be or cause an error on the accuracy of the prediction of diabetes. Whether to take these outliers into account or remove from the dataset is a topic

to debate upon. If it is obvious that the outlier is due to incorrectly entered or measured data, we should drop the outlier: If the outlier does not change the results but does affect assumptions, we may tend towards dropping the outlier.

Another issue is that some values in the dataset taken can have null or nil values this could be a mistake or wrongly entered data which can sabotage the outcome of the algorithm and hence producing wrong results for a patient.

To change these values to make meaning full rows of data is by finding the mean of that column and entering it in the mis-entered cells for better accuracy and efficiency. The changed values can be anything equation such as mean, standard deviation etc. There can also be redundant data in the dataset that has not been taken into account, which should be changed or removed from the dataset. These results have been verified by incorporating the confusion matrix to check for the true positives, true negatives, false positives and false negatives. The result throws the accuracy of each of these models on both the training data and the test data.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM WORKFLOW

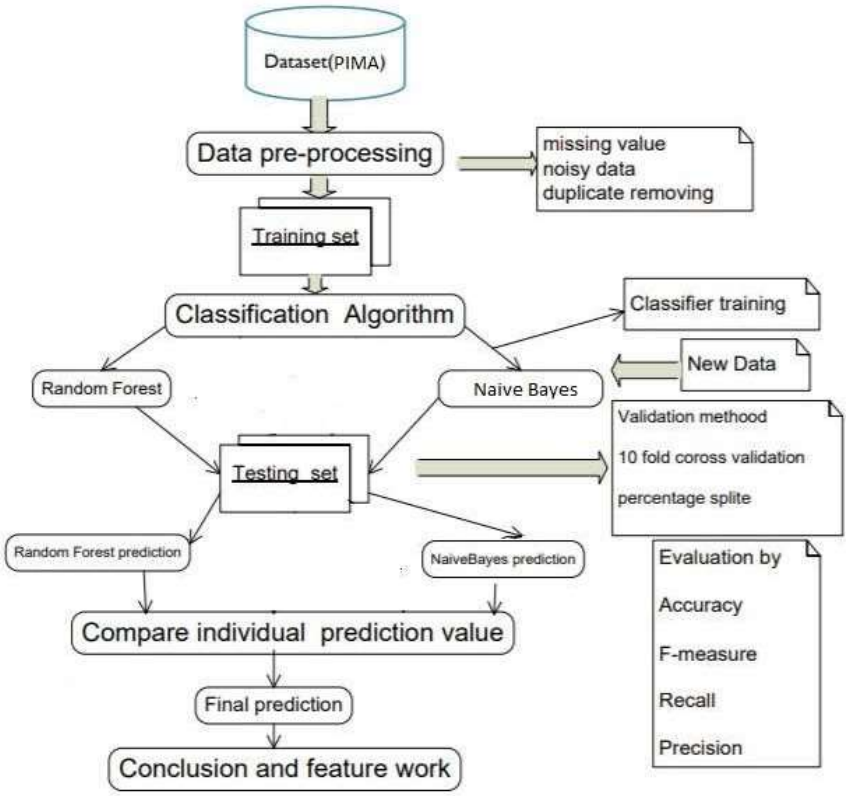

Fig 4.1: Proposed System Architecture

The proposed system architecture describes the workflow of the project we are working on.

First, we procure the dataset, which is the PIMA Indian dataset. It is a dataset which is used mainly for diabetes prediction. The dataset contains up to 1000 rows and mainly depicts the features required for the prediction of diabetes.

We split the dataset into training and testing data where part of the dataset is trained and part of the dataset is used for testing. We train the dataset in order to find the accuracy of the percentage of people having and not having diabetes.

Many methods are used for the purpose of the prediction of diabetes such as Naïve Bayes, Random Forest, Logistic Regression, Decision Trees etc. We mainly focus on Naïve Bayes and Random Forest as these two are the most efficient in getting an efficient result for the prediction.

We perform Naïve Bayes and Random Forest on the training and testing data and find the accuracy percentage of both the data for finding the best evaluation method among the two for the analysis of the dataset.

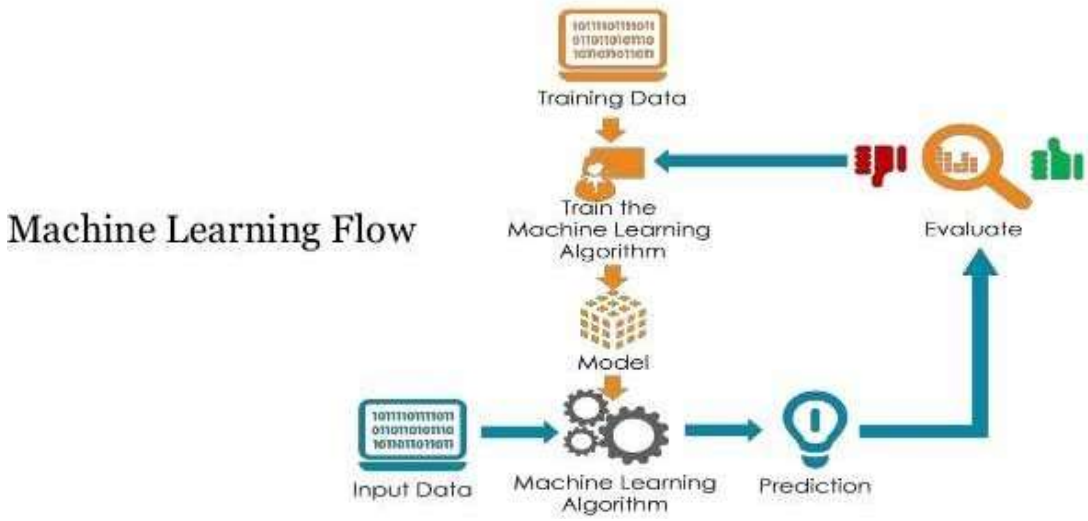

Fig 4.2: Machine Learning Flow

CHAPTER 5

PROPOSED METHODOLOGY

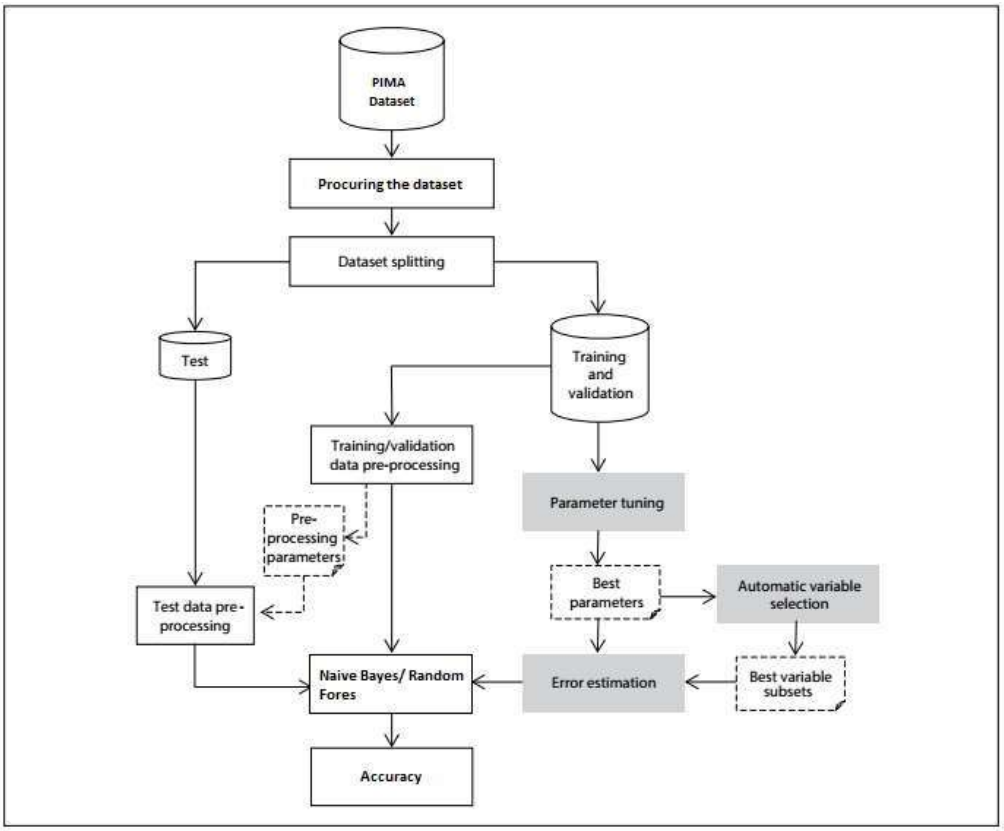

Fig 5.1: Workflow

PROCURING THE DATASET

The dataset used here is the PIMA Indian Dataset. It is the data obtained from the National Institute for Diabetes. It contains of several medical predictor variables and one target variable. The various medical variables are BMI, Glucose levels, Blood Pressure etc. It contains 768 rows and 9 columns. The columns that are present in the dataset are as follows:

SKIN THICKNESS

Skin thickness is a column in the dataset which denotes the thickness of an individual's skin. Skin thickness varies from person to person depending upon their health and various other factors which can affect the skin. A person's skin thickness can play a factor in

denoting whether the person has diabetes or not, but in the dataset, there are a few rows where the skin thickness is set to 0. Skin thickness cannot be 0 for a person, so we try to avoid this column mainly to get the accurate results while performing prediction. While performing analysis, the skin thickness column is removed from the code we write so as to get a more accurate prediction result using Naïve Bayes and Random Forest.

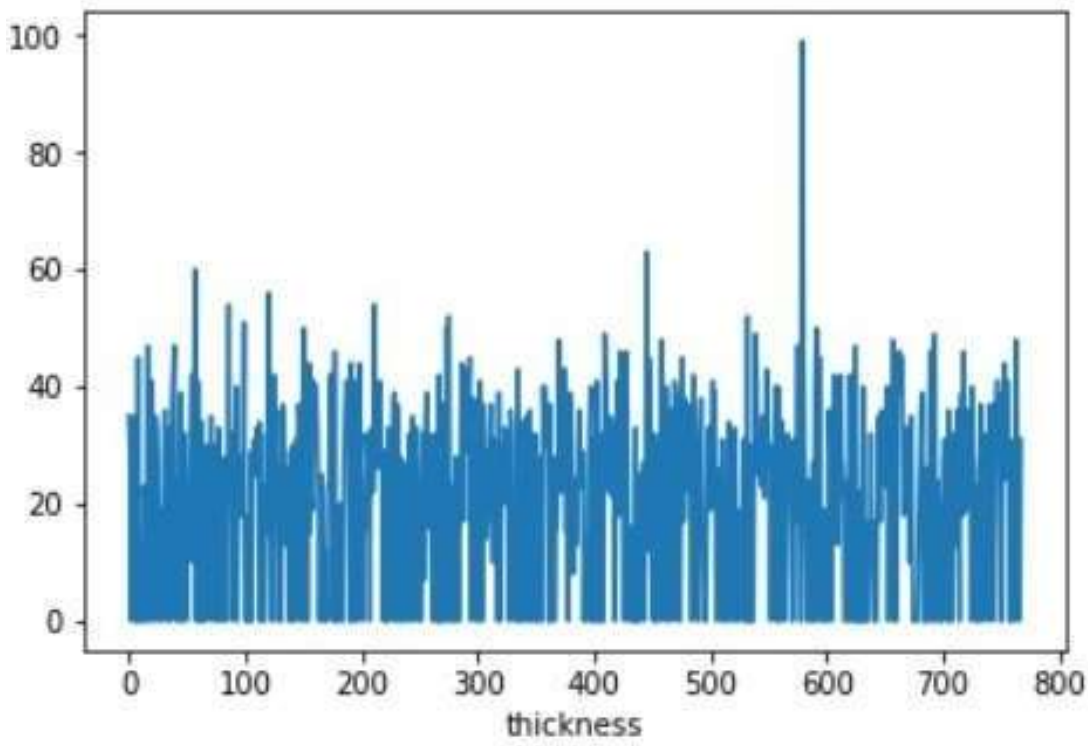

Fig 5.2: Skin Thickness

3.1.2 NUMBER OF PREGNANCIES

When a woman gets pregnant, they may or may not go through gestational pregnancy. Gestational pregnancy is a common form of pregnancy where the woman develops diabetes. After the birth, the diabetes usually goes away. The diabetes is caused due to the high levels of sugar in the body which does not happen when the woman is not pregnant. This is due to the making of hormones by the placenta. The number of pregnancies plays a key factor when it comes to diabetes in women. So we record the number of pregnancies and if it is a male, the pregnancy is set to 0 in the dataset. It can also denote that a woman has not been pregnant during her life.

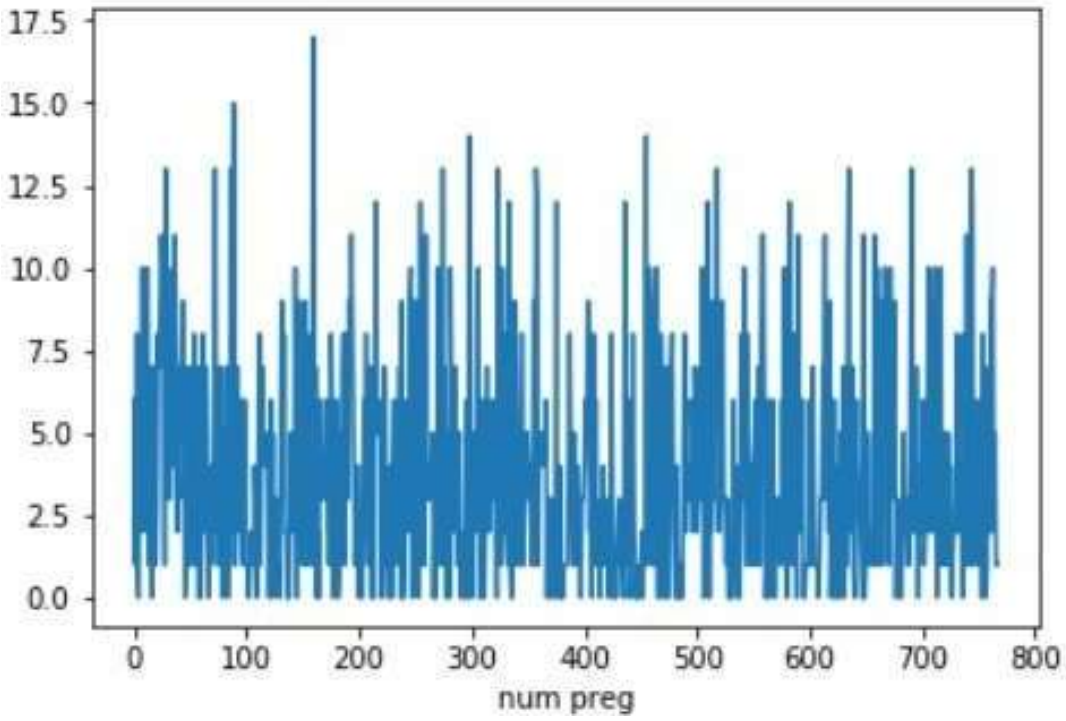

Fig 5.3: Number of Pregnancies

GLUCOSE CONCENTRATION

The glucose concentration is the level of glucose that is present in a person's blood. A teaspoon of glucose is required for a human body to function normally per day. The glucose present in the body travels through the bloodstream to other parts of the body. The glucose level is required to determine the amount of insulin present in the body. If the insulin is not able to handle the amount of glucose in the body, then this causes diabetes. The glucose levels in a person's body is an important factor in determining if the person has diabetes or not. In the dataset, we have a column to represent the glucose level of each person.

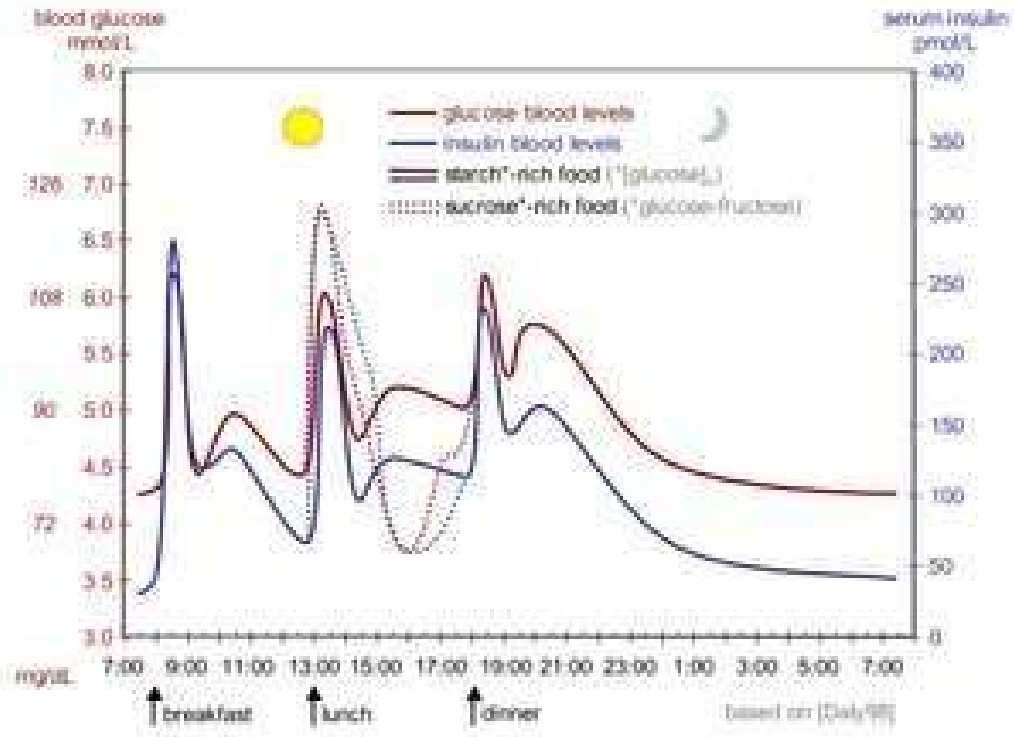

Fig 5.4: Blood Sugar Levels

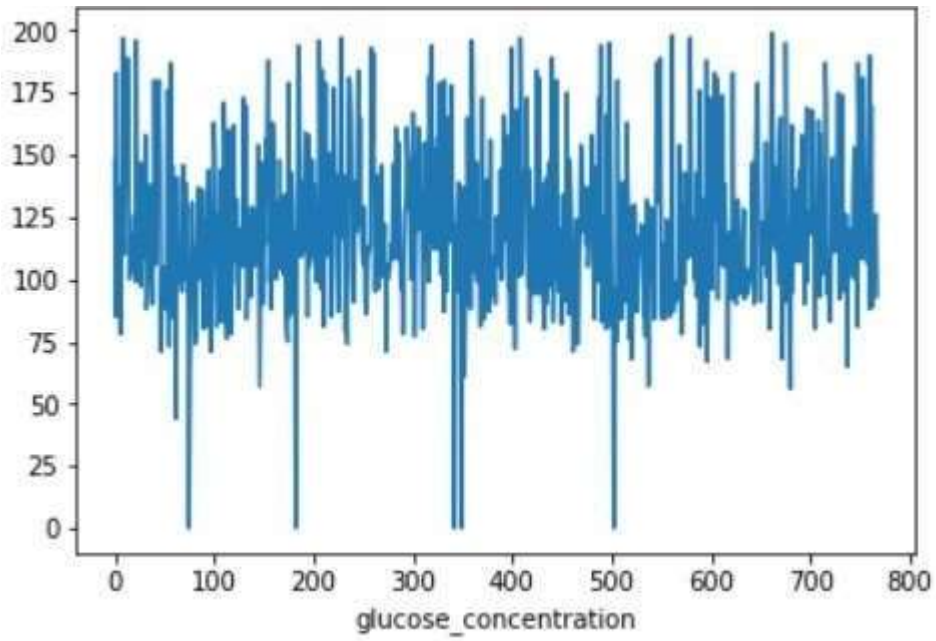

Fig 5.5: Glucose Concentration

BLOOD PRESSURE

The blood in our body moves through our body by the means of blood pressure. It helps in the movement of oxygen and nutrients throughout our body through the blood. The white blood cells in our body are also delivered by the means of blood pressure. The normal blood pressure for a person is usually below 120 mm Hg systolic and 80 mm Hg diastolic. Variations in blood pressure can be a major cause of diabetes mellitus. So we take this factor in our dataset for the prediction of diabetes.

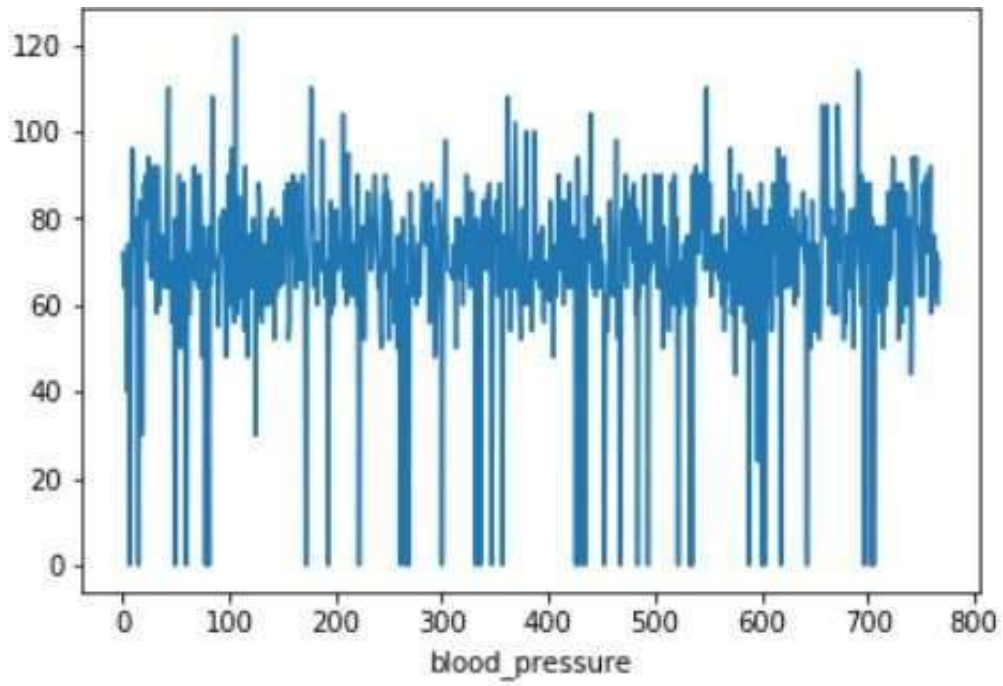

Fig 5.6: Blood Pressure

INSULIN

To control the blood sugar in our body, insulin is required. Insulin is a hormone created by the pancreas to balance all the sugars in our body. The insulin controls the glucose concentration, which is a major factor in development of diabetes. If the insulin is not able to keep up with the levels of glucose in our body, it causes diabetes. Insulin also helps in the breaking down of fats and proteins in our body to form energy. Insulin resistance is the inability of insulin to exert its effects on the tissues in our body. In the dataset, the insulin level plays a key role in the prediction of diabetes.

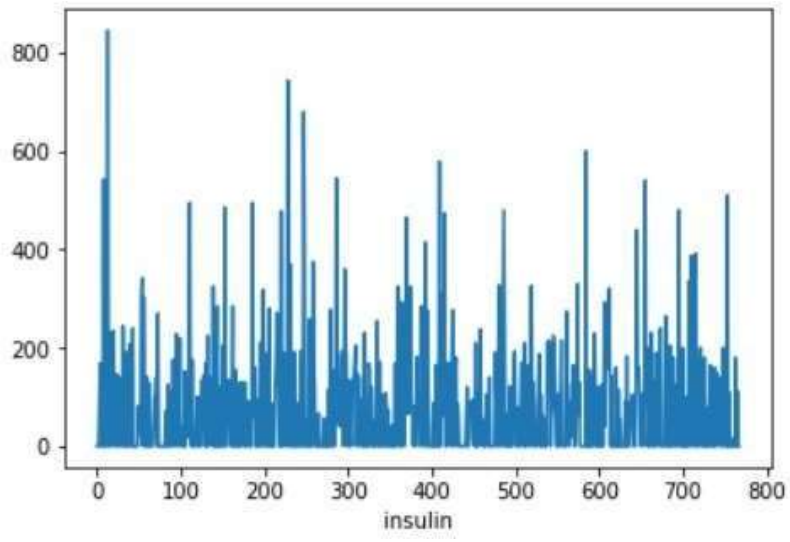

Fig 5.7: Insulin

BODY MASS INDEX (BMI)

The Body Mass Index of a person can be defined as the person's weight divided by the square of the height. The weight is defined in kgs and the height is defined in meters. The BMI of a person varies according to the weight and height and it calculates whether the person is normal or obese.

The following table denotes the various BMIs that distinguish a person into four categories:

Table 5.1: BMI & its Category

BMI	**Category**
Under 18.5 kg/m^2	Underweight
18.5 to 25	Normal Weight
25 to 30	Overweight
Over 30	Obese

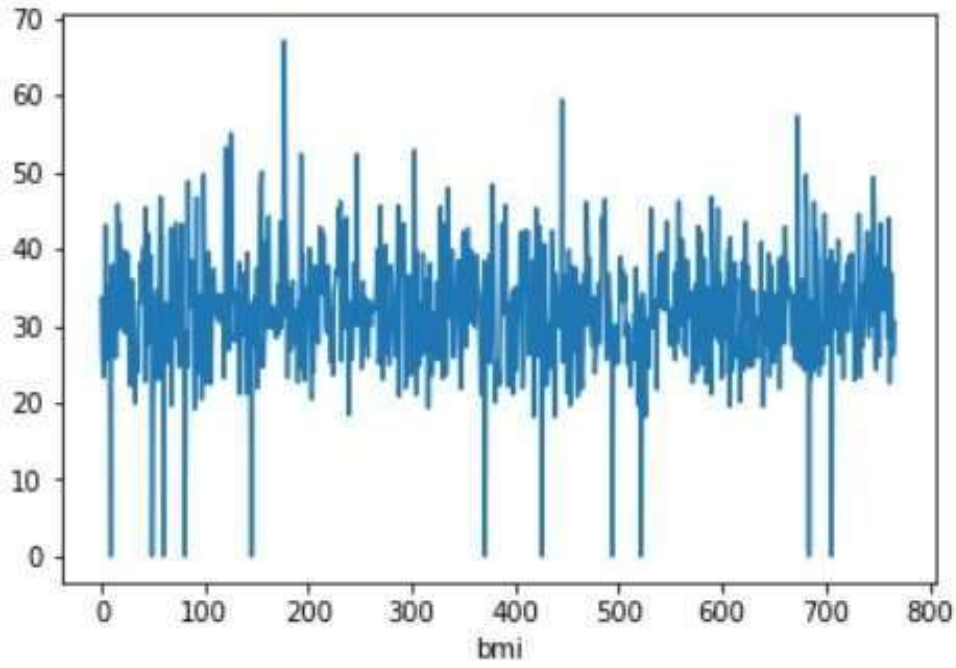

Fig 5.8: Body Mass Index(BMI)

DIABETES PEDIGREE FUNCTIN

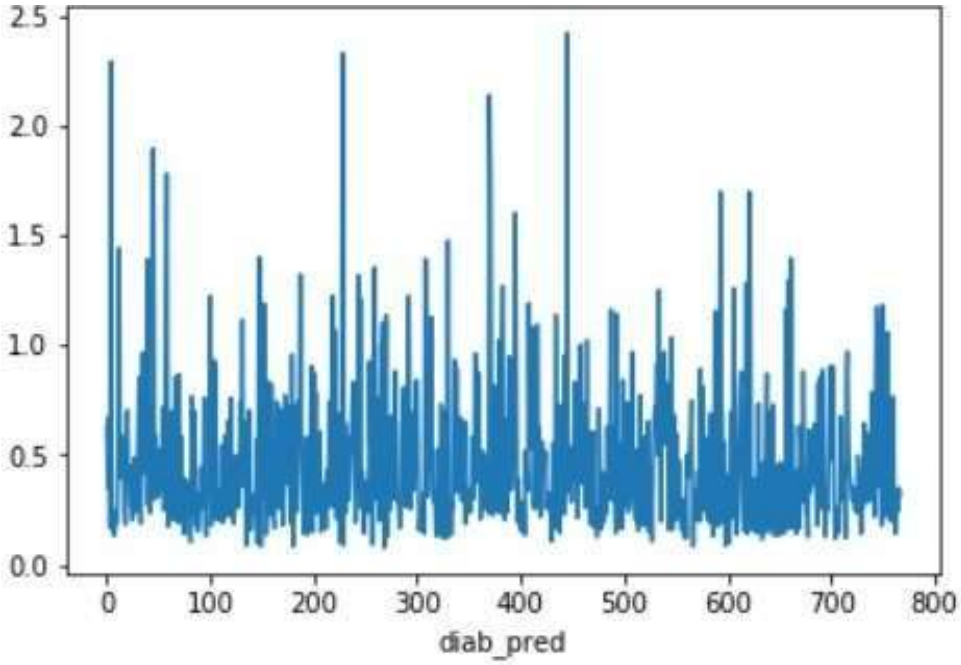

Fig 5.9: Diabetes Pedigree Function

AGE

Age is a common factor for diabetes. When it comes to age, usually, people above the age of 40 are diagnosed with diabetes. But, sometimes, even people who are younger are diagnosed with diabetes. Type 1 diabetes usually occurs in people above the age of 40 but sometimes, people at ages as young as 15 – 16 can also be diagnosed. This all depends on factors such as family history, diet etc.

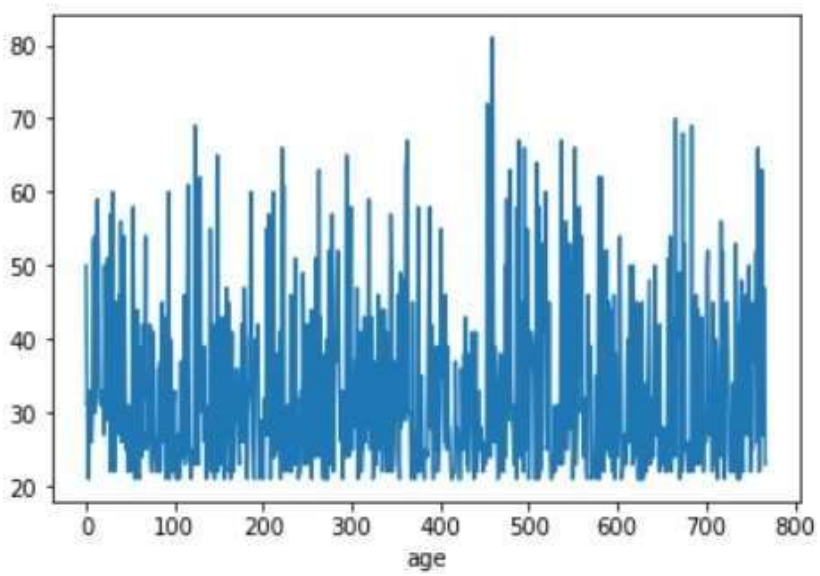

Fig 5.10: Age

VALUE OF DIABETES DISEASES

In the dataset, this column is used to define if the person has diabetes or not. We define it using True or False. The dataset has predefined values for each person whether the person has diabetes or not and our project is to find whether the given values are accurate or not.

The following features are the key to finding whether a person has diabetes or not. There are various other factors as to determining diabetes, but in our project, we are mainly focusing on these features for the prediction.

The dataset file is in a .csv(Comma Separated Values) format. Using the help of Python's inbuilt library Pandas, which is a dataframe library, we import the file into our Python environment. The other libraries that are imported into the environment are:

Numpy – a library that is used mainly to operate with large dimensional arrays and matrices, providing high level mathematical functionalities to work on data.

Matplotlib – the library that provides Python with the functionality of plotting graphs and plots. It works in tandem with NumPy. Pandas has a function named read_csv(), which essentially reads a file of the format (.csv).

Once the dataset is loaded into the environment, we can check the dimensions of the dataset by the function .shape() which returns the number of rows and columns. Basic lookup of the data is done, by using the inbuilt commands .head() and .tail() which print the number of rows from the start of the dataset and the bottom of the dataset respectively.

Number of Features	*Features*	*Descriptions and Features values*
1	*Number of times a person was pregnant*	*Numeric value*
2	*Glucose Concentration*	*Numeric value*
3	*Blood Pressure*	*Numeric value (in mm Hg)*
4	*Skin Thickness*	*Numeric value (in mm)*
5	*Insulin*	*Numeric value*
6	*Body Mass Index (BMI)*	*Numeric value (weight in kg/(height in m)^2)*
7	*Diabetes Pedigree Function*	*Numeric value*
8	*Age*	*Numeric value*
9	*Value of Diabetes Diseases*	*Yes = True No = False*

Fig 5.11: Features of Pima Indians Diabetes for Diagnosing Diabetes Disease Type 2

Number of Attributes	Attributes Name	Mean	Standard Deviation
1	Number of times a person was pregnant	3.8	3.4
2	Glucose Concentration	120.9	32.0
3	*Blood Pressure*	69.1	19.4
4	*Skin Thickness*	20.5	16.0
5	*Insulin*	79.8	115.2
6	*Body Mass Index (BMI)*	32.0	7.9
7	*Diabetes Pedigree Function*	0.5	0.3
8	*Age*	33.2	11.8

Fig 5.12: Statistical Analysis for Mean and Standard Deviation in Pima Indians Diabetes Data Set

After procuring the dataset, we see if we can make any changes to the dataset. Operations such as initialization of the variables, cleansing the data, making appropriate labels for the data takes place. In our case, the dataset contains a parameter skin thickness, which is column that has a weak correlation to the contribution of a person being diabetic. Hence, we remove the column for our analysis. In this stage, we can calculate the numeric aspects of the data, such as the average of a particular column, number of cases of the column based on conditions etc.

The dataset contains the values for the people having diabetes and people who don't. Hence, we calculated the count for each case. the result turned out like this:

People with Diabetes: 268

People without Diabetes: 500

In the given data, around 35% of the people have been diagnosed with diabetes.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Fig 5.13: Dataset sample

SPLITTING THE DATA

Splitting the data into training and test data, is one of the most crucial steps in the analysis. The split of the training data is more than the training data. The training data undergoes through learning. This data which is trained is later generalized on the other data, based on which the prediction is made. The dataset in our case, is split into multiple variants and prediction is performed accordingly. The dataset has multiple column that are medical predictors and one target column, that of the diabetes outcome. The medical predictors are given as inputs to a variable and the target variable is input to another variable.

Using the inbuilt function, train_test_split, the dataset is split into arrays and is mapped to training and test subsets. In our case, we are performing splits of 80/20,70/30,75/25,60/40 and the accuracy of each is recorded. It was noticed that the dataset contained values that were null, hence in order to streamline the analysis and the prediction, the null values were filled with the mean values of the respective columns.

NAÏVE BAYES

We import the model into a variable. Input the training data, fit it accordingly to the Gaussian Naïve Bayes model using the function .fit(). Then a classification of the array present in each of the training and test variables is performed. This classification is the key operation taking place as it is performing the prediction of the input data according the Naïve Bayes model. The accuracy of the model is then predicted by comparing the predicted model with the original model. This is done with the help of the metrics library present in sklearn.

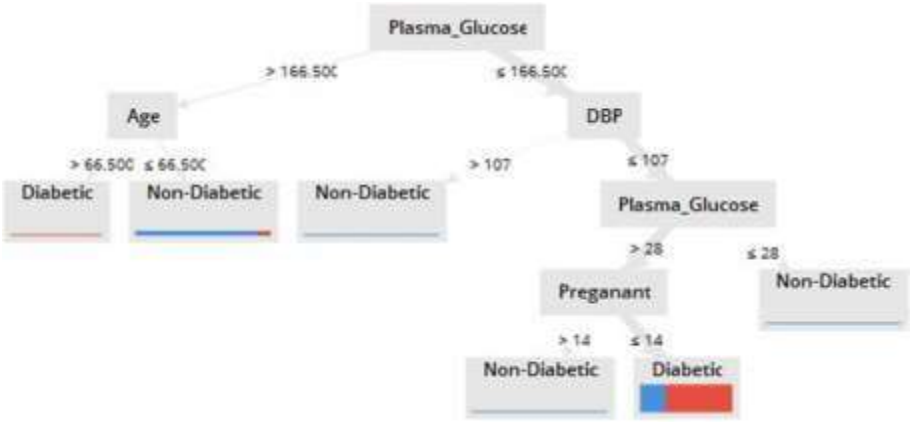

Fig 5.14: Tree structure for Random Forest

The model.fit() function trains the model for a given number of epochs (iterations on a dataset).

ARGUMENTS

x: Numpy array of training data (if the model has a single input), or list of Numpy arrays (ifthe model has multiple inputs). If input layers in the model are named, you can also pass a dictionary mapping input names to Numpy arrays. X can be None (default) if feeding from framework-native tensors (e.g. TensorFlow data tensors).

y: Numpy array of target (label) data (if the model has a single output), or list of Numpy arrays (if the model has multiple outputs). If output layers in the model are named, you can also pass a dictionary mapping output names to Numpy arrays. Y can be None (default) if feeding from framework-native tensors (e.g. TensorFlow data tensors).

Y_train.ravel() returns contiguous flattened array(1D array with all the input-
To predict values using the training data, we use the predict function. A class prediction is: given the finalized model and one or more data instances, predict the class for the data instances.

We do not know the outcome classes for the new data. That is why we need the model in the first place. We can predict the class for new data instances using our finalized classification model in scikit-learn using the *predict()* function.

To perform prediction, we make use of the tool sklearn. Scikit-learn is probably the most useful library for machine learning in Python. It is on NumPy, SciPy and matplotlib, this library contains a lot of effiecient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

The module **sklearn.metrics** also exposes a set of simple functions measuring a prediction error given ground truth and prediction:

- functions ending with _score return a value to maximize, the higher the better.

- functions ending with _error or _loss return a value to minimize, the lower the better. When converting into a scorer object using **make_scorer**, set the greater_is_better parameter to False (True by default; see the parameter description below).

FINDING THE ACCURACY

The next step is to find the accuracy of the training and testing data. To find the accuracy, we use a function called metrics.accuracy_score. In multilabel classification, this function computes subset accuracy: the set of labels predicted for a sample must *exactly* match the corresponding set of labels in y_true.

First, we check the accuracy of the training data by passing the arguments for the training data split. After that, we check the accuracy of the testing data by doing the same with the testing data as the parameters. By comparing both, we print a confusion matrix.

A confusion matrix is used to evaluate the accuracy of a classification. By definition a confusion matrix C is such that $C_{i,i}$ is equal to the number of observations known to be in group i but predicted to be in group j. Thus in binary classification, the count of true negatives is $C_{0,0}$, false negativesis $C_{1,0}$, true positives is $C_{1,1}$ and false positives is $C_{0,1}$.

Parameters:

y_true : array, shape = [n_samples]. Ground truth (correct) target values.

y_pred : array, shape = [n_samples]. Estimated targets as returned by a classifier.

labels : array, shape = [n_classes], optional. List of labels to index the matrix. This may be used to reorder or select a subset of labels. If none is given, those that appear at least once in y_true or y_pred are used in sorted order.

sample_weight : array-like of shape = [n_samples], optional.

Returns: C : array, shape = [n_classes, n_classes].Confusion matrix

RANDOM FOREST

In our case, we have split the labels into two variables, which is input to the classifier. One of the greatest strengths of Random Forest classifiers is its ability to be used with practically in any kind of data, especially with feature selection.

In our dataset we have a feature selection process, hence the use of the model. The model is available in the sklearn library, hence we utilize that to begin our prediction. In our case, we use the RandomForestClassifier() function to perform prediction on the data. The input training and test data, is fitted to the model using fit(). The training data is then classified into arrays during prediction. The accuracy of the model is obtained by comparing the predicted values against the original set of values.

The model.fit() function trains the model for a given number of epochs (iterations on a dataset).

ARGUMENTS

x: Numpy array of training data (if the model has a single input), or list of Numpy arrays (ifthe model has multiple inputs). If input layers in the model are named, you can also pass a dictionary mapping input names to Numpy arrays. X can be None (default) if feeding from framework-native tensors (e.g. TensorFlow data tensors).

also pass a dictionary mapping output names to Numpy arrays. Y can be None (default) iffeeding from framework-native tensors (e.g. TensorFlow data tensors).

Y_train.ravel() returns contiguous flattened array(1D array with all the input-array elementsand with the same type as it). A copy is made only if needed.

To predict values using the training data, we use the predict function. A class prediction is: given the finalized model and one or more data instances, predict the class for the data instances.

We do not know the outcome classes for the new data. That is why we need the model in the first place. We can predict the class for new data instances using our finalized classification model in scikit-learn using the *predict()* function.

To perform prediction, we make use of the tool sklearn. Scikit-learn is probably the most useful library for machine learning in Python. It is on NumPy, SciPy and matplotlib, this library contains a lot of effiecient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

The goal of ensemble methods is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability / robustness over a single estimator. We use the sklearn.ensemble tool to import RandomForestClassifier.

Train_test_split splits arrays or matrices into random train and test subsets. That means that everytime we run it without specifying random_state, we will get a different result, this is expected behavior.

If use random_state = some number, then we can guarantee that the outputs will be equal i.e. the split will be always the same. It doesn't matter what the actual random_state number is 42, 0, 21, ... The important thing is that everytime we use 42, we will always get the same output the first time we make the split. This is useful if we want reproducible results, for example in the documentation, so that everybody can consistently see the same numbers when they run the examples. In practice I would say, you should set the random_state to

some fixed number while we test stuff, but then remove it in production if we really need a random (and not a fixed) split.

FINDING THE ACCURACY

The next step is to find the accuracy of the training and testing data. To find the accuracy, we use a function called metrics.accuracy_score. In multilabel classification, this function computes subset accuracy: the set of labels predicted for a sample must *exactly* match the corresponding set of labels in y_true.

First, we check the accuracy of the training data by passing the arguments for the training data split. After that, we check the accuracy of the testing data by doing the same with the testing data as the parameters. By comparing both, we print a confusion matrix.

A confusion matrix is used to evaluate the accuracy of a classification. By definition a confusion matrix C is such that $C_{i,i}$ is equal to the number of observations known to be in group i but predicted to be in group j. Thus in binary classification, the count of true negatives is $C_{0,0}$, false negativesis $C_{1,0}$, true positives is $C_{1,1}$ and false positives is $C_{0,1}$.

Parameters:

y_true : array, shape = [n_samples]. Ground truth (correct) target values.

y_pred : array, shape = [n_samples]. Estimated targets as returned by a classifier.

labels : array, shape = [n_classes], optional. List of labels to index the matrix. This may be used to reorder or select a subset of labels. If none is given, those that appear at least once in y_true or y_pred are used in sorted order.

sample_weight : array-like of shape = [n_samples], optional. Sample weights.

Returns: C : array, shape = [n_classes, n_classes].Confusion matrix

CLASSIFICATION REPORT

The classification report visualizer shows the exactness, review, F1, and bolster scores for the model. So as to help simpler elucidation and issue recognition, the report coordinates numerical scores with a shading coded heatmap. All heatmaps are in the range (0.0, 1.0) to encourage simple examination of classification models crosswise over various classificationreports.

```
Confusion Matrix
[[38 17]
 [18 81]]

Classification Report

              precision    recall  f1-score   support

           1       0.68      0.69      0.68        55
           0       0.83      0.82      0.82        99

   micro avg       0.77      0.77      0.77       154
   macro avg       0.75      0.75      0.75       154
weighted avg       0.77      0.77      0.77       154
```

Fig 5.15: Classification Report for Naïve Bayes

```
Confusion Matrix
[[37 18]
 [19 80]]

Classification Report

              precision    recall  f1-score   support

           1       0.66      0.67      0.67        55
           0       0.82      0.81      0.81        99

   micro avg       0.76      0.76      0.76       154
   macro avg       0.74      0.74      0.74       154
weighted avg       0.76      0.76      0.76       154
```

Fig 5.16: Classification Report for Random Forest

CHAPTER 6

SOFTWARE TESTING

UNIT TESTING

It is the way toward testing every single module created by the designers. The whole program is divided into numerous bundles which comprise of little units of code. It improves the general structure of the module and refactors the code wherever essential. These modules are tried autonomously independent of different modules. They are tried in a successive request also, it checks for repetition. If there should arise an occurrence of redundancy it erases the copy records. It too checks for run time blunder and checks if the connection gave take them to the individual page. Preferred standpoint of performing unit testing is its capacity to check every module exclusively which is supportive in finding the littlest of littlest mistakes. Since unit testing is done at an in all respects early stage the expense of testing is negligible when contrasted with other testing. Modules which are as well enormous for unit testing can be assessed utilizing integration testing.

INTEGRATION TESTING

This is subsequent stage after unit testing is performed. Once, every module tried autonomously is clear of mistakes, these individual modules are consolidated together and tried in general. The fundamental explanation behind playing out this test is to check for issues when every one of the units are joined. There are diverse manners by which these units can be coordinated. They are

1. Top Down Integration - Top-down mix joins and tests every one of the modules start to finish. However, one inconvenience of this testing is that it needs more stubs.
2. 2. Bottom Up Integration - The base up methodology is the other way around of top-down approach. Significant modules are tried last which can make issues amid combination.

3. 3. Big-Bang Integration - In this type of testing every one of the functionalities are incorporated and tried at the same time. This methodology is subject to the quantity of modules present. Lesser The modules progressively successful it is.
4. Hybrid Integration – It is a mix of all the above methodologies.

SYSTEM TESTING

System Testing is the subsequent stage after coordination testing. In this procedure the entire item is tried for issues and mistakes. They are of two kinds:

1. Black box testing

2. White box testing

A case for this is assembling of ballpoint pen. The top, the ink cartridge, the body, the tail is created independently and tried independently (unit testing). Whenever at least two modules are prepared, they are consolidated and Integration Testing is finished. At the point when the total pen is collected, System Testing is finished. It thinks about the entire system as single element.

1. Black Box Testing

It is a testing method which is completed by the analyzers. This product can be tried without knowing the inward structure of program. Programming Knowledge isn't expected to do this type of testing procedure. Its fundamental desire is to check for the activity that is performed by the system. It is less tedious. Black box testing is generally called functional test or external testing. It isn't best for algorithm testing. It very well may be tried on preeminent dimensions of testing like acceptance testing.

2. White Box Testing

It is a testing technique which is done by s/w engineers. The usefulness of the program must be known to the designer. Programming learning is an unquestionable requirement to perform White Box Testing. It is generally called inside testing or basic testing. Its principle

point is to check program code, circles, conditions, branches and how framework is performing. It tends to be tried on more elevated amounts of testing like acknowledgment testing and acknowledgment testing.

REGRESSION TESTING

This is a standout amongst the most significant sort of testing with regards to the correct advancement of a product. We can likewise consider it as one significant advance in the Software Development Life Cycle (SDLC). Each product has a particular sort of functionalities which should be refreshed without fail. This is typically done to guarantee its security in all stages. Along these lines, for this to be guaranteed, these functionalities need to refreshed with new bit of code without fail. In this manner, so as to guarantee that the new code doesn't influence the new usefulness, relapse testing is completed. This is normally done by specialists or programming engineers who have profound comprehension of the product activities in and out.

SMOKE TESTING

It is additionally one angle to ensure that the usefulness is simply working fine independent of the new code that is added to change it. A standout amongst the most significant motivation to play out this type of testing is to expel every one of those lines of code that isn't required any longer and make sure that they try not to influence the usefulness of the product. It covers the greater part of the critical elements of the programming however does not dissect them in detail. The outcome of this test is used to pick whether to proceed with further testing. If the smoke test passes, continue with further testing. In case it misses the mark, end further tests and demand another structure with the required fixes.

ACCEPTANCE TESTING

This is the last period of testing which is performed by or before customers. This testing is fundamentally done to check whether the created item fulfills the customer's necessity. They are 4 distinctive manners by which acknowledgment testing can be performed. They are Client acceptance testing, Business acceptance testing, Alpha testing and Beta testing

Table 6.1: Different Testcases with their obtained outcomes

TEST CASE ID	TEST CASE DESCRIPTION	STEP DETAILS	EXPECTED RESULT	ACTUAL RESULT	STATUS
001	Check if the data is being correctly mapped to the dataframe	We check if the dataframe contains data or not	The assertion is to be True as the dataframe does contains the imported data	The data is present in the dataframe.	Pass
002	Check if the number of rows and columns in the csv are matching with the entries in the dataset	Check if there are 768 rows and 10 columns in the dataframe	The assertion is to be True as the dataframe must contain 768 columns and 10 rows	The dataframe consists of 768 columns and 10 rows.	Pass
003	Check if there are any null values/empty spaces in the table	Check if there are any null values/empty spaces in the table	The assertion must return False as there aren't any empty spaces in the dataframe	The assertion returns True	Fail
004	Check if the total number of cases are the correct number	Check if total=number of positive + number of negative	The assertion must return True, as the total is 768	The assertion returned is True, proving the total to be 768	Pass
005	Check if the values of the column are of type array	Check if the feature column names values are of type array	The assertion must return True, as the values in the variable is an array	The assertion returns True as the variable contains values of type Array	Pass
006	Check if the values of the column are of type array	Check if the predicted column names values	The assertion must return True, as the values in the	The assertion returns True as the variable	Pass

		are of type array	variable is an array	contains values of type Array	
007	Check if the accuracy rate for training is more than 60%	Check if the accuracy rate for Naïve Bayes training is more than60%	The assertio n must return True, as theaccuracy expected is more than 60%	The assertio n returned is True as the accuracy rate is morethan 60%	Pass
008	Check if the accuracy rate for testing is more than 60%	Check if the accuracy rate for Naïve Bayes testing is more than 60%	The assertio n must returnTrue, as the accuracy expected is more than60%	The assertio n returned is True as the accuracy rate is morethan 60%	Pass
009	Check if the accuracy rate for training is more than 70%	Check if the accuracy rate for Random Forest training is more than 70%	The assertio n must returnTrue, as the accuracy expected is more than 70%	The assertio n returned is True as the accuracy rate is morethan 70%	Pass
010	Check if the accuracy rate for testing is more than 70%	Check if the accuracy rate for Random Forest testingis more than 70%	The assertion must returnTrue, as the accuracy expected is more than 70%	The assertion returned is True as the accuracy rate is morethan 70%	Pass

Since machine learning is more of a heuristic process, it is not possible to do a definitive testing for the analysis, we can only assume a certain parameter. Here the testing of the data is performed as a test split, which in itself can be called an operation of testing.

Testing Cases, above are performed to check and validate if the operations and functions involved in performing the analysis are being in the correct manner or not.

CHAPTER 7

EXPERIMENTAL RESULTS

TABULATED RESULTS

After performing the Random Forest and Naive Bayes algorithms, we are generating the following results for the different splits of training and testing data:

Table 7.1: Prediction Using Naïve Bayes

Train	Test	Train_Result (% value)	Test_Result (% value)
60	40	75.22%	77.27%
70	30	75.98	74.89
75	25	75.87	74.48
80	20	75.57	77.27

In the above table, we can see that for the four different splits, we get results that are close to 75% in the training set and 74-77% in the test results.

This depicts that the training set has been trained up to 75% accuracy which means that the data that has been trained has been used to predict the test results which have a 75% average accuracy in the analyzing of the dataset.

For each split, the percentage of test results depicts that 74-77% of the dataset prediction is accurate and rest of the 25% approx. cannot be predicted due to various other reasons.

Table 7.2: Prediction Using Random Forest

Train	Test	Train_Result (% value)	Test_Result (% value)
60	40	97.61	77.27 (best)
70	30	98.70	73.16
75	25	98.61	72.92
80	20	98.37	74.68

In the above table, we can see that for the four different splits, we get results that are close to 98% in the training set and 72-77% in the test results.

This depicts that the training set has been trained up to 98% accuracy which means that the data that has been trained has been used to predict the test results which have a 75% average accuracy in the analyzing of the dataset.

For each split, the percentage of test results depicts that 72-77% of the dataset prediction is accurate and rest of the 25% approx. cannot be predicted due to various other reasons.

While analyzing both the tables, we can understand that the Random Forest algorithm has a better training set result which in turn gives a better accuracy of the prediction and analysis. The dataset is trained to the maximum accuracy where all variables are taken into aspect without excluding missing data as Random Forest algorithm will make sure that there is no missing data in large datasets.

Naïve Bayes algorithm tends to ignore missing data which does not provide accurate results while performing analysis. From the tables, we can find out that the best prediction result is giving by the 60/40 split while performing Random Forest.

COMPARISON GRAPHS

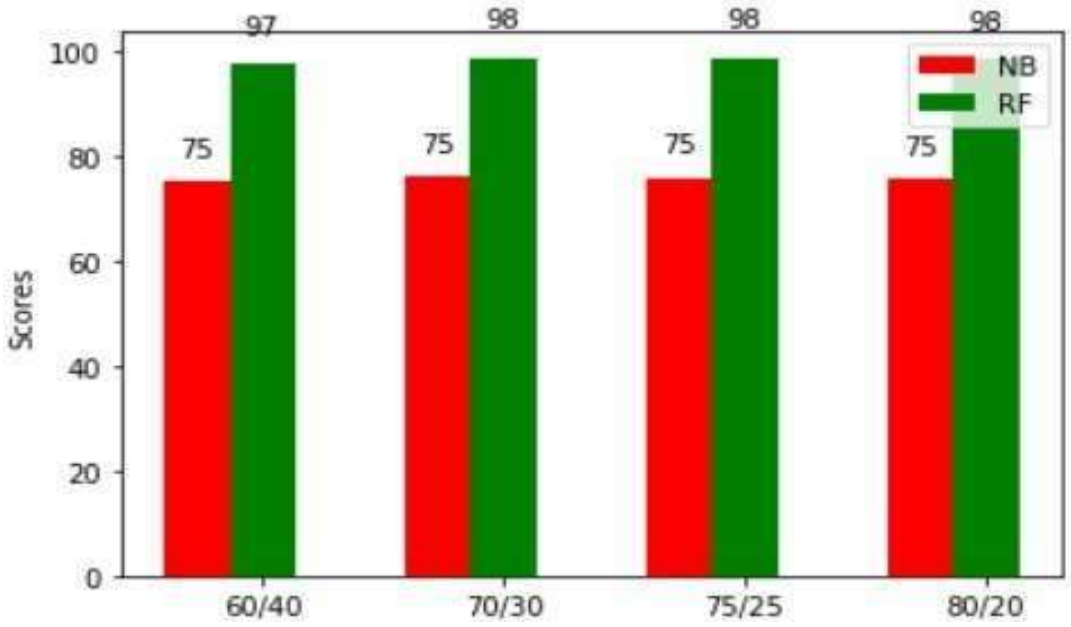

Fig 7.1: Comparison of Training results for various splits

The above graph depicts the comparison graph for the training results for both Naïve Bayes and Random Forest for various splits. We can understand that the Random Forest

training results are more accurate when compared to that of Naïve Bayes as it gives a 98% accuracy when it comes to training the dataset.

The training result of Naïve Bayes is very low compared to that of Random Forest as there are errors that occur in the Naïve Bayes algorithm while performing training. Sometimes, it cannot detect missing data so there are fluctuations and errors in the accuracy of the result, but in the case of Random Forest, it gives the proper accuracy even when it comes to large datasets like PIMA dataset.

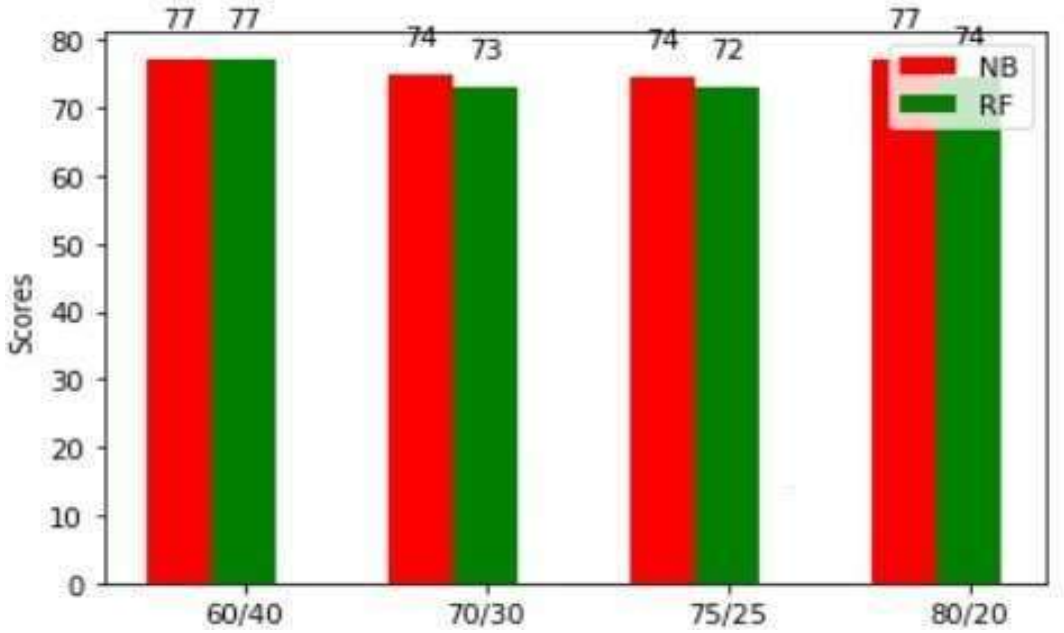

Fig. 7.2: Comparison of Test results for various splits

The above graph depicts the comparison graph for the testing results for both Naïve Bayes and Random Forest for various splits. We can understand that the Random Forest and Naïve Bayes test results are almost the same and they differ by 2-3%.

Eventhough the Naïve Bayes testing results are greater compared to the Random Forest results, the training result for Naïve Bayes was lesser than that of Random Forest, so the accuracy of the results when compared, is greater for Random Forest since the training data was much more accurate when compared to Naïve Bayes.

After analyzing the results, we can come to the conclusion that the Random Forest algorithm is a more efficient method to analyze the dataset using means of splitting it into training and testing sets. It serves as a more accurate method of prediction of diabetes.

CHAPTER 8

CONCLUSION

Diabetes is one of the most chronic and the largest growing disease in India. According to the World Health Organization (WHO), India had 69.2 million people living with diabetes as of 2015. A study conducted by the American Diabetes Association states that India will see a great increase in the number of people diagnosed with diabetes by the 2030. Identifying diabetes or predicting the upcoming of a diabetic life can be propelled by using various machine learning techniques like Naïve Bayesian Network, Random Forest etc.

From this project, we can conclude that the best method of prediction of diabetes is Random Forest. This method gives us an approximate result after the splitting and analysis of the training and testing data. The efficiency of this method is much better compared to that of Naïve Bayes.

The analysis done from the PIMA dataset is really important. The aim of splitting the dataset is to find the highest/best accuracy of the Algorithms and as to how they would respond if the data split set is varied. Procuring the dataset is done to make sure that there are no empty values In the data set so that the accuracy of our prediction model is high. Preprocessing of the dataset makes sure that all the attributes (columns) are taken into account while predicting. From the above prediction and analysis, we can observe that the results obtained using Random Forest Algorithm give us an accuracy of 98%. The several decision trees that are part of Random Forest are used to result in this maximum efficiency value. Thereby , we can conclude that it is more efficient than Naïve Bayes. Hence this proposed method will give us an efficient method for both analysis and prediction of diabetes.

CHAPTER 9

FUTURE ENHANCEMENT

Diabetes is one of the most chronic and the largest growing disease in India. According to the World Health Organization (WHO), India had 69.2 million people living with diabetes as of 2015. A study conducted by the American Diabetes Association states that India will see a great increase in the number of people diagnosed with diabetes by the 2030.

So early identification, detection and diagnosis is of atmost importance. So we can do prediction and analysis by using other algorithms on our dataset. The process of Feature Selection can be done more efficiently so that we can reduce the number and type of attributes. This will increase the performance of our algorithms. We can also narrow down the most important attributes or features that are useful for diabetes prediction.

Healthcare professions found it hard to find healthcare data and perform analysis on them due to lack of tools, resources. But using ML, we can overcome this and can perform analysis on real-time data leading to better modelling, predictions. This enhances and improves the overall healthcare services. Now, IOT is being integrated with ML in order to make smart healthcare devices which sense if there is any change in the person's body, health data when he uses the device (Pacemaker, Stethoscope, etc.) and this will notify the person regarding this through an app. This helps in easy monitoring, advanced prediction and analysis thereby reducing errors, saving time and life of people.

CHAPTER 10

REFERENCES

[1] Priyanka Indoria, Yogesh Kumar Rathore (2018). Asurvey: Detection and Prediction of diabetes using machine learning techniques. IJERT

[2] Khaleel, M.A., Pradhan, S.K., G.N Dash (2Ol3). A Survey of Data Mining Techniques on Medical Data for Finding frequent diseases. IJARCSSE.

[3] K. Vembandasamy, R. Sasipriya, E. Deepa (2015). Heart Diseases Detection using Naïve Bayes Algorithm. IJISET.

[4] Tawfik Saeed Zekia, Mohammad V. Malakootib, Yousef Ataeipoorc, S. Talayeh Tabibid. An Expert System for Diabetes Diagnosis. American Academic & Scholarly Research Journal Special Issue Vol. 4, No. 5, Sept 2012.

[5] Vishali Bhandari and Rajeev Kumar. Comparative Analysis of Fuzzy Expert Systems for Diabetic Diagnosis. International Journal of Computer Applications (0975 – 8887) Volume 132 – No.6, December 2015.

[6] Ioannis Kavakiotis, Olga Tsave, Athanasios Salifoglou, "Machine Learning and Data Mining Methods in Diabetes Research", Jan 8, 2017.

[7] Eka Miranda, Edy Irwansyah, Alowisius Y. Amelga, Marco M. Maribondang, Mulyadi Salim (2016). Detection of cardiovascular Disease Risk's Level for Adults using naïve Bayes Classifier, The Korean Society of Medical informatics (KOSMI).

[8] Zheng T, Xie W, Xu L, He X, Zhang Y, You M, Yang G, Chen Y (2017). A Machine Learning-Based Framework to identify Type 2 Diabetes through Electronic Health Records, International Journal of medical informatics (IJMI) Vol 9, pages 120-127.

[9] Francesco Mercaldo, Vittoria Nardone, Antonella San- tone (2017). Diabetes Mellitus Affected Patients Classification and Diagnosis through Machine Learning.