MINI PROJECT

(2020-2021)

“DIABETES PREDICTION”

Project Report

GLA University, Mathura

Institute of Engineering and Technology

Submitted By:-

Janvi Pangoriya (181500292)

Nidhi Gupta (181500422)

Under the Supervision

Of

Mohd. Amir Khan

(Technical Trainer)

Department of Computer Engineering & Application

Department of Computer Engineering and Applications

G LA University, 17 km. Stone NH-2, Mathura-Delhi Road,

Chaumuha, Mathura – 281406 U.P. (India)

Declaration

I/we hereby declare that the work which is being presented in the Bachelor of Technology. Project **"Diabetes Prediction"**, in partial fulfillment of the requirements for the award of the Bachelor of Technology in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my/our own work carried under the supervision of **Mohd. Amir Khan, Technical Trainer, Dept. of CEA, GLA University.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign: ***JanviPangoriya***

Name of Candidate: Janvi Pangoriya

University Roll No.:181500292

Sign: ***NidhiGupta***

Name of Candidate: Nidhi Gupta

University Roll No.:181500422

Department of Computer Engineering and Applications

GLA University, 17 km. Stone NH-2, Mathura-Delhi Road,

Chaumuha, Mathura – 281406 U.P. (India)

Certificate

This is to certify that the project entitled **“Diabetes Prediction”,** carried out in Mini Project – II Lab, is a bonafide work by Janvi Pangoriya (3rd B.Tech CSE Student) bearing roll no 181500292, and Nidhi Gupta (3rd B.Tech CSE Student) bearing roll no 181500422 and is submitted in partial fulfillment of the requirements for the award of the degree i.e. Bachelor of Technology (Computer Science & Engineering).

(Signature of the Supervisor)

Name:

Designation:

Department:

Date:

Department of Computer Engineering and Applications

GLA University, 17 km. Stone NH-2, Mathura-Delhi Road,

Chaumuha, Mathura – 281406 U.P. (India)

Acknowledgement

Presenting the ascribed project paper report in this very simple and official form, we would like to place my deep gratitude to GLA University for providing us the instructor Mohd. Amir Khan, our technical trainer and supervisor.

He has been helping us since Day 1 in this project. He provided us with the roadmap, the basic guidelines explaining on how to work on the project. He has been conducting regular meeting to check the progress of the project and providing us with the resources related to the project. Without his help, we wouldn’t have been able to complete this project.

And at last but not the least we would like to thank our dear parents for helping us to grab this opportunity to get trained and also my colleagues who helped me find resources during the training.

Thanking You

Sign: ***JanviPangoriya***	**Sign:** ***NidhiGupta***
Name of Candidate: Janvi Pangoriya	**Name of Candidate: Nidhi Gupta**
University Roll No.:181500292	**University Roll No.:181500422**

ABSTRACT

Diabetes has evolved as one the most dangerous threat to the human world. Many are becoming its victims and are unable to come out of it regardless of the fact that they are working to avoid it for growing further. Cloud Computing and Internet of Things (IoT) are two tools that play a very important role in today's life regarding many aspects and purposes including healthcare monitoring of patients and elderly society. Diabetes Healthcare Monitoring Services are very important nowadays because and that to remote healthcare monitoring because physically going to hospitals and standing in a queue is very ineffective version of patient monitoring. If a patient has very chronic diabetes and he spends his/her time standing in a queue anything dangerous can happen to him/her at any instance of time. So, this paper came up with different machine learning algorithms like logistic regression algorithm, random forest. Diabetes can also act as a means for other diseases like heart attack, kidney damage and somewhat blindness. This paper can make use of various machine learning algorithms such as support vector machine, logistic regression, decision tree, and random forest with the help of which can easily find out the total efficiency and accuracy of predicting that a human will suffer from diabetes or not. There are variously many traditional methods which are totally different from software methods that can diagnose diabetes and predict pre conditions of diabetic patients. A diabetic is caused due to a vast uphill in the blood portion containing glucose. There is an optimization scheme available through the use of train test split using sklearn learn method.

In this project we are creating a machine learning model for the prediction of diabetes in human beings and this model will be deployed on the cloud so as to provide a web based application to the user where the user can enter the parameters on which the prediction is being made and the model will return the probability of having diabetes.

CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER – 1

INTRODUCTION

1.1 CONTEXT

This Machine Learning Application "Diabetes Prediction" has been submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering at GLA University, Mathura supervised by Mohd. Amir Khan. This project has been completed approximately three months and has been executed in modules, meetings have been organized to check the progress of the work and for instructions and guidelines.

1.2 MOTIVATION

Diabetes is noxious diseases in the world. Diabetes caused because of obesity or high blood glucose level, and so forth. It affects the hormone insulin, resulting in abnormal metabolism of crabs and improves level of sugar in the blood. Diabetes occurs when body does not make enough insulin. According to (WHO) World Health Organization about 422 million people suffering from diabetes particularly from low or idle income countries. And this could be increased to 490 billion up to the year of 2030. However prevalence of diabetes is found among various Countries like Canada, China, and India etc. Population of India is now more than 100 million so the actual number of diabetics in India is 40 million. Diabetes is major cause of death in the world. Early prediction of disease like diabetes can be controlled and save the human life. To accomplish this, this work explores prediction of diabetes by taking various attributes related to diabetes disease. For this purpose we use the Pima Indian Diabetes Dataset, we apply various Machine Learning classification Techniques to predict probability of having diabetes. Machine Learning is a method that is used to train computers or machines explicitly. Various Machine Learning Techniques provide efficient result to collect knowledge by building various classification and ensemble models from collected dataset. Such collected data can be useful to predict diabetes. Various techniques of Machine Learning can capable to do prediction, however it's tough to choose best technique. Thus for this purpose we apply popular classification and ensemble methods on dataset for prediction.

1.3 OBJECTIVE

In this project, We are creating a machine learning model for the prediction of diabetes in human beings and this model will be deployed on the cloud so as to provide a web based application to the user where the user can enter the parameters on which the prediction is being made and the model will return the probability of having Probability. We are trying to predict the diabetes basedon the following parameters in this project. There are 8 factors we have taken into consideration in this project. They are pregnancies (no of times the woman has been pregnant, the glucose concentration (Plasma glucose concentration a 2 hours in an oral glucose tolerance test), diastolic blood pressure (pressure in the arteries when the heart rests between beats. This is the time when the heart fills with blood and gets oxygen. A normal diastolicblood pressure is lower than 80. A reading of 90 or higher means you have high blood pressure),skin thickness (Triceps skin fold thickness), Insulin (2-Hour serum insulin), Body mass index(weight in kg / (height in m) ^ 2), Diabetes pedigree function, Age (years). The last column isthe outcome which is the class variable which is a dependent variable results in 0 if the person is not diabetic and 1 if the person is diabetic.

1.4 LITERATURE REVIEW

K. Vijay Kumar et al. [11] proposed random Forest algorithm for the Prediction of diabetes develop a system which can perform early prediction of diabetes for a patient with a higher accuracy by using Random Forest algorithm in ma- chine learning technique. The proposed model gives the best results for diabetic prediction and the result showed that the prediction system is capable of predicting the diabetes disease effectively, efficiently and most importantly, instantly. Nonso Nnamoko et al. [13] presented predicting diabetes onset: an ensemble supervised learning approach they used five widely used classifiers are employed for the ensembles and a meta-classifier is used to aggregate their outputs. The results are presented and compared with similar studies that used the same dataset within the literature. It is shown that by using the proposed method, diabetes onset prediction can be done with higher accuracy.

N. Joshi et al. [12] presented Diabetes Prediction Using Machine Learning Techniques aims to predict diabetes via three different supervised machine learning methods including: SVM, Logistic regression, ANN. This project pro- poses an effective technique for earlier detection of the diabetes disease. Comparison of the different machine learning techniques used in this study reveals which algorithm is best suited for prediction of diabetes. Diabetes Prediction is becoming the area of interest for researchers in order to train the program to identify the patient are diabetic or not by applying proper classifier on the dataset. Based on previous research work, it has been observed that the classification process is not much improved. Hence a system is required as Diabetes Prediction is important area in computers, to handle the issues identified based on previous research.

1.5 SOURCES

The source of our project (including all the project work, documentations and presentations) will is available at the following link

https://github.com/JanviPangoriya/Diabetes-prediction

CHAPTER – 2

SOFTWARE REQUIREMENT

2.1 IMPACT OF DIABETES ON HUMAN

Diabetes can be effectively managed when caught early. However, when left untreated, it can lead to potential complications that include heart disease, stroke, kidney damage, and nerve damage.

Normally after you eat or drink, your body will break down sugars from your food and use them for energy in your cells. To accomplish this, your pancreas needs to produce a hormone called insulin. Insulin is what facilitates the process of pulling sugar from the blood and putting it in the cells for use, or energy.

If you have diabetes, your pancreas either produces too little insulin or none at all. The insulin can't be used effectively. This allows blood glucose levels to rise while the rest of your cells are deprived of much-needed energy. This can lead to a wide variety of problems affecting nearly every major body system.

Diabetes can also affect your skin, the largest organ of your body. Along with dehydration, your body's lack of moisture due to high blood sugar can cause the skin on your feet to dry and crack. It's important to completely dry your feet after bathing or swimming. You can use petroleum jelly or gentle creams, but avoid letting these areas become too moist.

Moist, warm folds in the skin are susceptible to fungal, bacterial, or yeast infections. These tend to develop between fingers and toes, the groin, armpits, or in the corners of your mouth. Symptoms include redness, blistering, and itchiness.

High-pressure spots under your foot can lead to calluses. These can become infected or develop ulcers. If you do get an ulcer, see your doctor immediately to lower the risk of losing your foot. You may also be more prone to boils, folliculitis (infection of the hair follicles), sties, and infected nails.

2.2 PROBLEM STATEMENT

Diabetes is an illness caused because of high glucose level in a human body. Diabetes should not be ignored if it is untreated then Diabetes may cause some major issues in a person like: heart related problems, kidney problem, blood pressure, eye damage and it can also affects other organs of human body. Diabetes can be controlled if it is predicted earlier. To achieve this goal this project work we will do early prediction of Diabetes in a human body or a patient for a higher accuracy through applying, Various Machine Learning Techniques. Machine learning techniques provide better result for prediction by constructing models from datasets collected from patients. In this work we will use Machine Learning Classification and ensemble techniques on a dataset to predict diabetes. The accuracy will different for every model when compared to other models. The project work will give the accurate or higher accuracy model shows that the model is capable of predicting diabetes effectively.

To solve this problem efficiently we have collected a dataset from UCI repository which is named as Pima Indian Diabetes Dataset. The dataset have many attributes of 768 patients. The 9th attribute is class variable of each data points. This class variable shows the outcome 0 and 1 for diabetics which indicates positive or negative for diabetics. Distribution of Diabetic patient- We made a model to predict diabetes however the dataset was slightly imbalanced having around 500 classes labeled as 0 means negative means no diabetes and 268 labeled as 1 means positive means diabetic. The graph below shows the graphical representation of the dataset we have collected.

2.3 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirement

- Processor : intel
- Operating System :Any Operating System
- RAM : 8 GB (or higher)
- Hard disk : 256GB

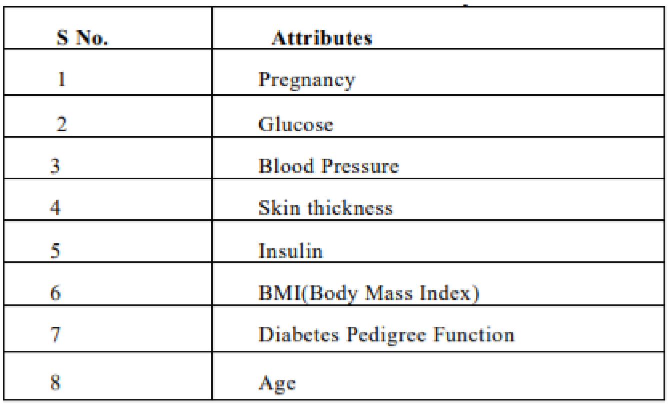
Software Requirement

- Software used: Anaconda (Jupyter Notebook)
- Language used: Python, HTML, CSS
- Deployed: Heroku
- Technology Used: Machine Learning

2.4 MODULES AND FUNCTIONALITIES

2.4.1 Dataset Description

The data is gathered from UCI repository which is named as Pima Indian Diabetes Dataset. The dataset have many attributes of 768 patients. The 9th attribute is class variable of each data points. This class variable shows the outcome 0 and 1 for diabetics which indicates positive or negative for diabetics. Distribution of Diabetic patient- We made a model to predict diabetes however the dataset was slightly imbalanced having around 500 classes labeled as 0 means negative means no diabetes and 268 labeled as 1 means positive means diabetic.



S No.	Attributes
1	Pregnancy
2	Glucose
3	Blood Pressure
4	Skin thickness
5	Insulin
6	BMI(Body Mass Index)
7	Diabetes Pedigree Function
8	Age

Table 1: Dataset Description

2.4.2 Data Preprocessing

Data pre-processing is most important process. Mostly healthcare related data contains missing vale and other impurities that can cause effectiveness of data. To improve quality and effectiveness obtained after mining process, Data pre-processing is done. To use Machine Learning Techniques on the dataset effectively this process is essential for accurate result and successful prediction. For Pima Indian diabetes dataset we need to perform pre-processing in two steps.

1). Missing Values removal- Remove all the instances that have zero (0) as worth. Having zero as worth is not possible. Therefore this instance is eliminated. Through eliminating irrelevant features/instances we make feature subset and this process is called features subset selection, which reduces dimensionality of data and help to work faster.

2). Splitting of data- After cleaning the data, data is normalized in training and testing the model. When data is spitted then we train algorithm on the training data set and keep test data set aside. This training process will produce the training model based on logic and algorithms and values of the feature in training data. Basically aim of normalization is to bring all the attributes under same scale

2.4.3 Apply Machine Learning

When data has been ready we apply Machine Learning Technique. We use different classification and ensemble techniques, to predict diabetes. The methods applied on Pima Indians diabetes dataset. Main objective to apply Machine LearningTechniques to analyze the performance of these methods and find accuracy of them, and also been able to figure out the responsible/important feature which play a major role in prediction. The following classification algorithm will be applied like Logistic Regression, K Nearest Neighbors, Random Forest, Decision Tree and many more to find the best accuracy.

2.4.4 Model Building:

This is most important phase which includes model building for prediction of diabetes. In this we have implemented various machine learning algorithms which are discussed above for diabetes prediction.

Procedure of Proposed Methodology

Step1: Import required libraries, Import diabetes dataset.

Step2: Pre-process data to remove missing data.

Step3: Perform percentage split of 80% to divide dataset as Training set and 20% to Test set.

Step4: Select the machine learning algorithm i.e. K Nearest Neighbor, Support Vector Machine, Decision Tree, Logistic regression, Random Forest and Gradient boosting algorithm.

Step5: Build the classifier model for the mentioned machine learning algorithm based on training set.

Step6: Test the Classifier model for the mentioned machine learning algorithm based on test set.

Step7: Perform Comparison Evaluation of the experimental performance results obtained for each classifier.

Step8: After analyzing based on various measures conclude the best performing algorithm.

2.4.5 Deploying the model on Cloud: After the best performing algorithm is selected, the built model is deployed on cloud on the Heroku platform and all the calculations will be taking place on cloud and we will be getting a live link to the web based application where on entering the parameters the model will predict whether the patient is diabetic or not. The front-end part of the model is designed using HTML and CSS and the backend uses Flask.

2.5 DIABETES PREDICTIONS AS A MACHINE LEARNING APPLICATION

Machine learning methods are widely used in predicting diabetes, and they get preferable results. Decision tree is one of popular machine learning methods in medical field, which has grateful classification power. Random forest generates many decision trees. So in this study, we used decision tree, random forest (RF) to predict the diabetes.

CHAPTER - 3

SOFTWARE DESIGN

3.1 USE CASE

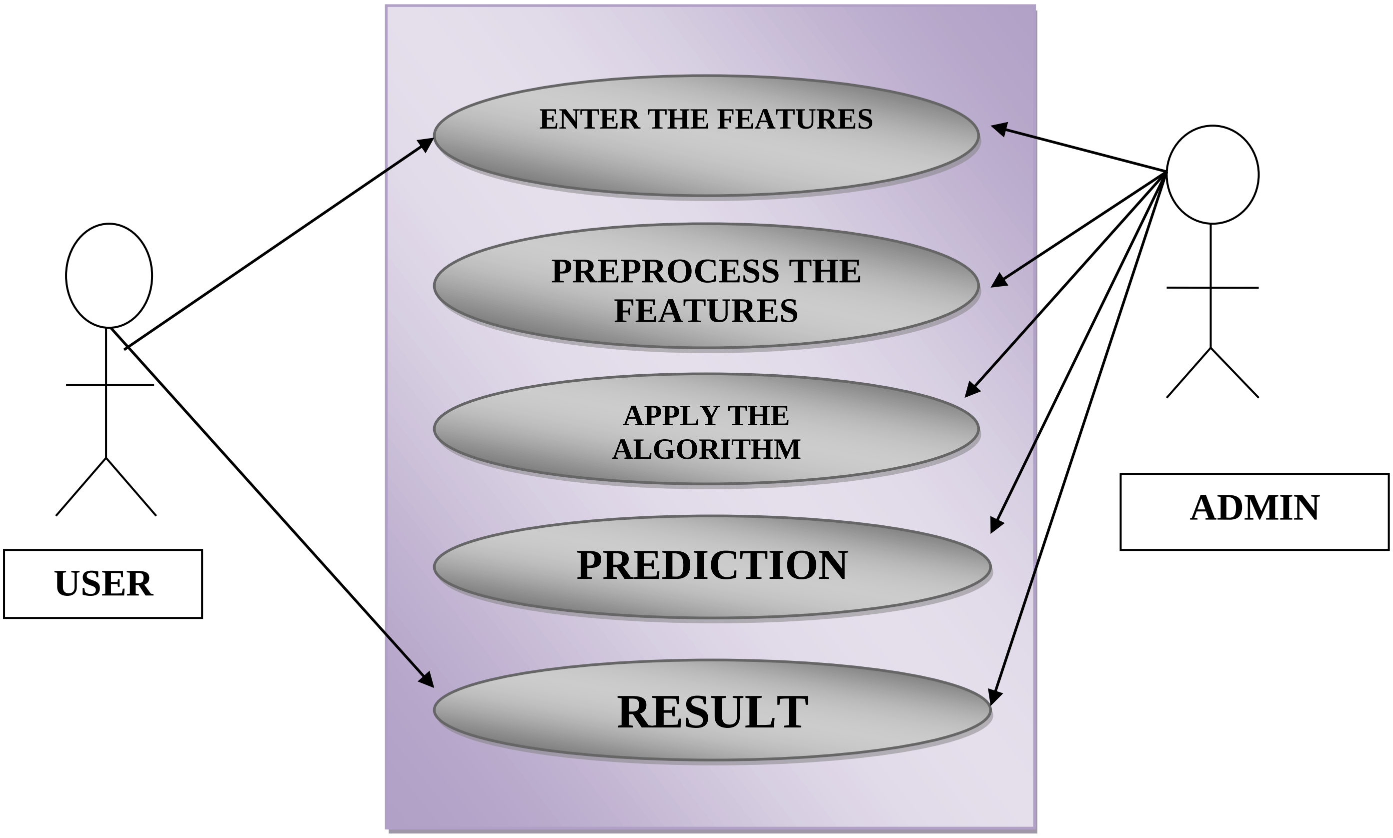


Figure 1: **Use Case Diagram**

Use Case Diagram is a visual representation of how one interacts with the model or software and which modules of the software can be accessed. The above use case diagram shows the visual representation of the "DIABETES PREDICTION" model. As we observed there will be two views, from which the person can interact one will be the view of the machine and other will be the view of the person trying to predict the probability of having diabetes.

3.2 DATA FLOW DIAGRAM

3.2.1 LEVEL-0 DFD

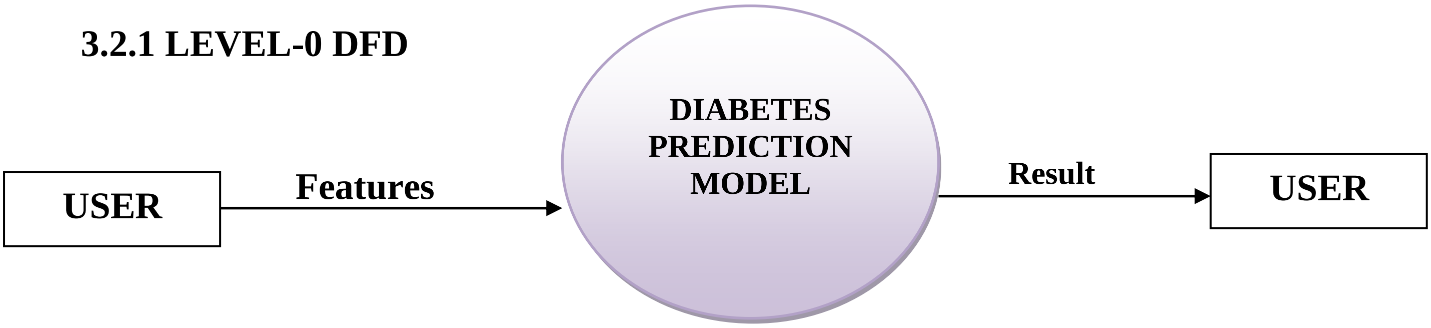


Figure 2: Level -0 DFD Diagram

3.2.2 LEVEL – 1 DFD

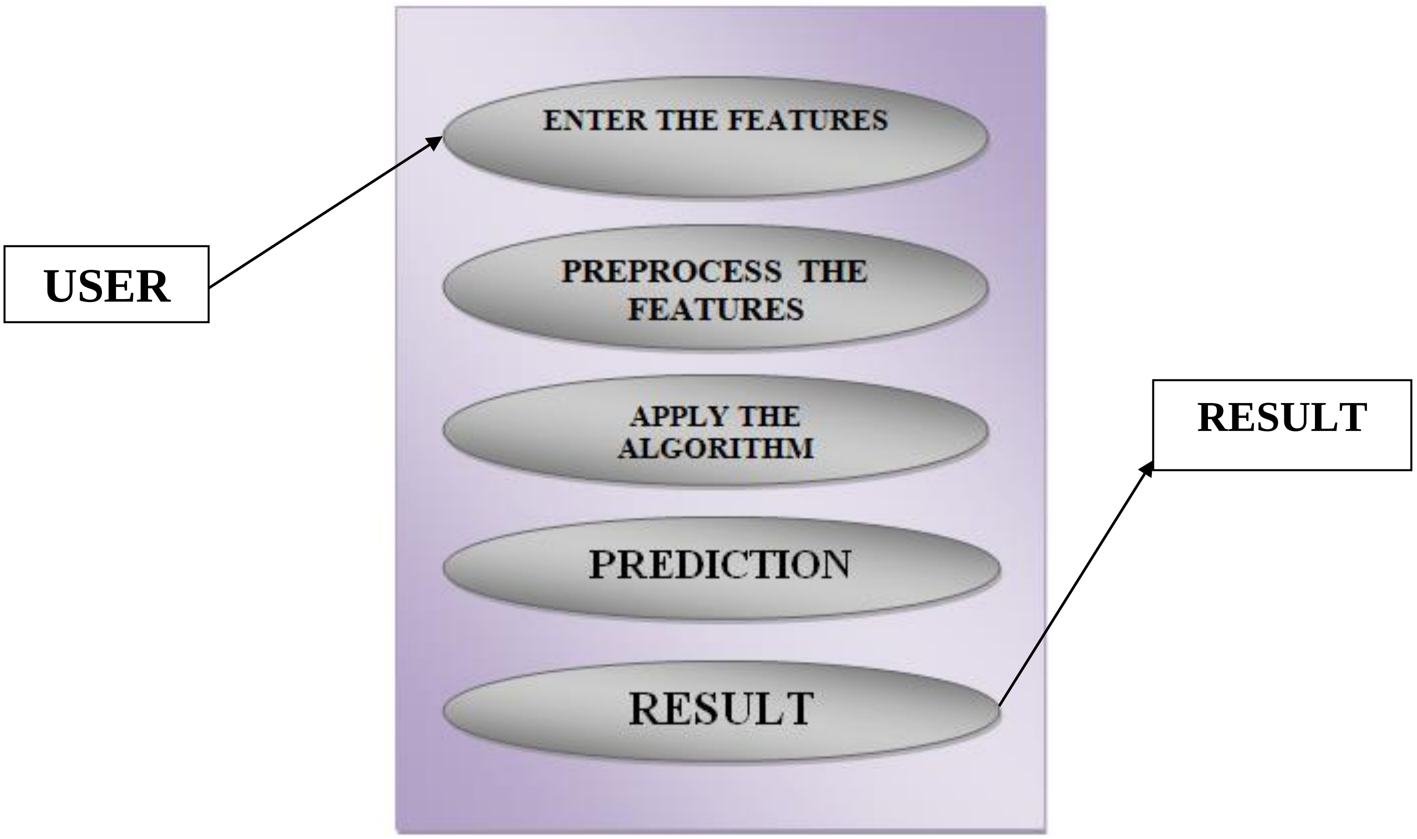


Figure 3: Level-1 DFD

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. As the user gives the input (Feature) the data is sent for the pre-processing then the model is Social Media Sentiment Analysis using the algorithm and then the features is taken as input for predicting the probability of having diabetes.

CHAPTER - 4

METHODOLOGY

In this we approached to predict the probability of having diabetes. Implementation of the same was carried out in five sub phases:

1) Collection of data from PIMA.

2) Perform pre-processing on features for classification and Training the Model

3) Evaluation of different model performance based on features

4) Improving performance

5) Discussion and Presentation of results

This project was developed in Python using Sci-kit libraries. Python has a huge set of libraries and extensions, which can be easily used in Machine Learning. Sci-Kit Learn library is the best source for machine learning algorithms where nearly all types of machine learning algorithms are readily available for Python, thus easy and quick evaluation of ML algorithms is possible.

4.1 Machine Learning Workflow:

There are five core tasks in the common ML workflow:

1. Get Data: The first step in the Machine Learning process is getting data. This process depends on your project and data type.

2. Clean, Prepare & Manipulate Data: Real-world data often has unorganized, missing, or noisy elements. Therefore, for Machine Learning success, after we chose our data, we need to clean, prepare, and manipulate the data. This process is a critical step, and people typically spend up to 80% of their time in this stage. Having a clean data set helps with your model's accuracy down the road. After getting the data to a state you like, you need to convert the data sets into valid formats for your chosen ML platform. For example, you may need to translate the data into a .CSV file. Finally, you split your data into training and test data sets. The training set is used to train the model in the next step, while the test data is used to validate the model in the fourth step. The typical default is a 70/30 split between training and test sets.

3. Train Model: This step is where the magic happens! The data set connects to an algorithm, and the algorithm leverages sophisticated mathematical modeling to learn and develop predictions.

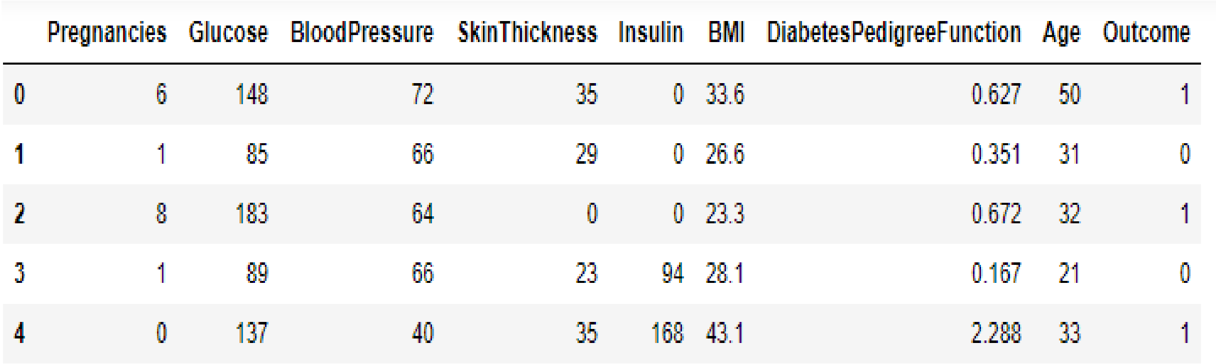
4. Test Model: Now, it’s time to validate your trained model. Using the test data from Step 3, we check the model’s accuracy. If the results are not satisfactory, you need to improve and retrain your ML model.

5. Improve: Practice makes perfect! Here are a few things you can do to refine your model and improve accuracy: Review your model’s results with your business stakeholders. Are there other data elements worth adding to your model to make it more accurate?

4.2 FLOW CHART

4.2.1 COLLECTING THE DATA

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. Thedatasets consists of several medical predictor variables and one target variable, Outcome. Predictor variable includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.



	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

Figure 4: Dataset

4.2.2 DATA PRE-PROCESSING

Basically it is the process of making the data ready to be fed to the machine learning algorithm. This includes the cleaning of the dataset like to check if there a null value in any of the attribute which is being considered for evaluation. Through eliminating irrelevant features/instances we make feature subset and this process is called features subset selection, which reduces dimensionality of data and help to work faster.

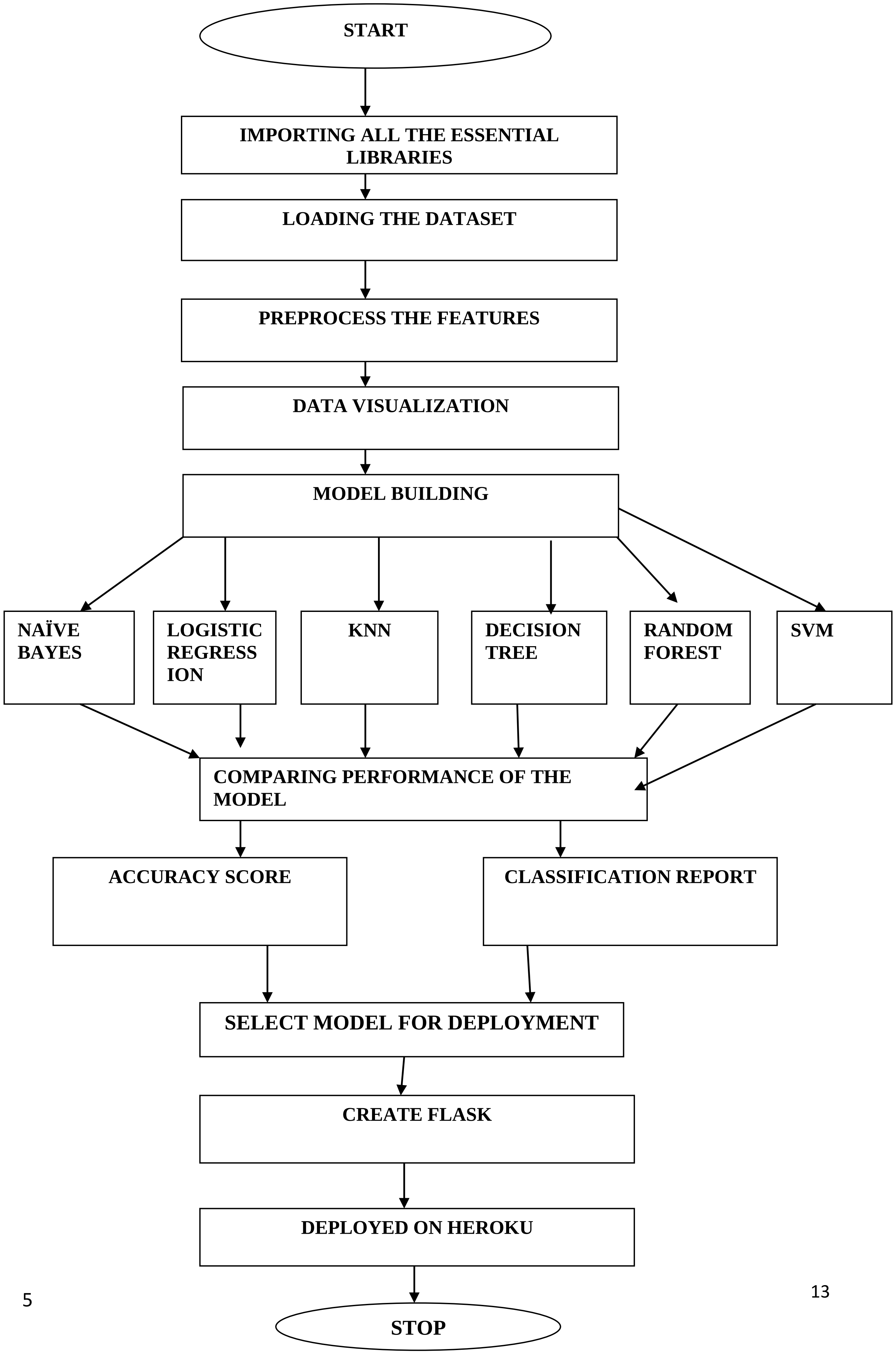

START
IMPORTING ALL THE ESSENTIAL LIBRARIES
LOADING THE DATASET
PREPROCESS THE FEATURES
DATA VISUALIZATION
MODEL BUILDING
NAÏVE BAYES
LOGISTIC REGRESS ION
KNN
DECISION TREE
RANDOM FOREST
SVM
COMPARING PERFORMANCE OF THE MODEL
ACCURACY SCORE
CLASSIFICATION REPORT
SELECT MODEL FOR DEPLOYMENT
CREATE FLASK
DEPLOYED ON HEROKU
STOP

Figure 5: Flow Chart

4.2.3 DATA VISUALIZATION

Data visualization gives us a clear idea of what the information means by giving it visual context through maps or graphs. This makes the data more natural for the human mind to comprehend and therefore makes it easier to identify trends, patterns, and outliers within large data sets. Data visualization takes the raw data, models it, and delivers the data so that conclusions can be reached. In advanced analytics, data scientists are creating machine learning algorithms to better compile essential data into visualizations that are easier to understand and interpret.

Specifically, data visualization uses visual data to communicate information in a manner that is universal, fast, and effective. This practice can help to identify which areas need to be improved, which factors affect customer satisfaction and dissatisfaction, and what to do with specific areas (outliners). Visualized data gives decision-makers a better prediction of future growth.

<u>Visualizing the outcome column</u>

The data shows that these are records of 768 patients out of which 500 are non-diabetic and 268 are diabetic patients.

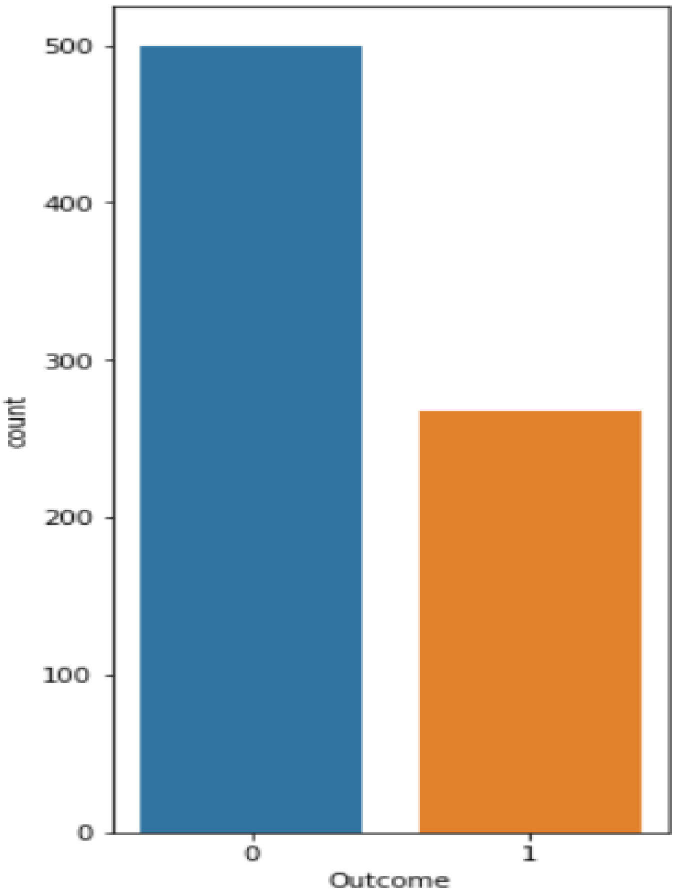


Figure 6: Bar Plot of Outcome

The dataset also shows that 65.1% are diabetic and 39.4% people are healthy on plotting a pie chart according to the given dataset.

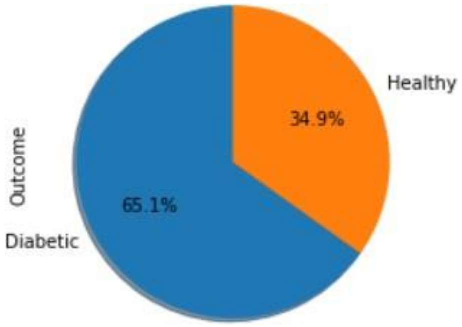


Figure 7: Pie Plot of Outcome

Analysis of “Pregnancies” Parameter with respect to the Outcome

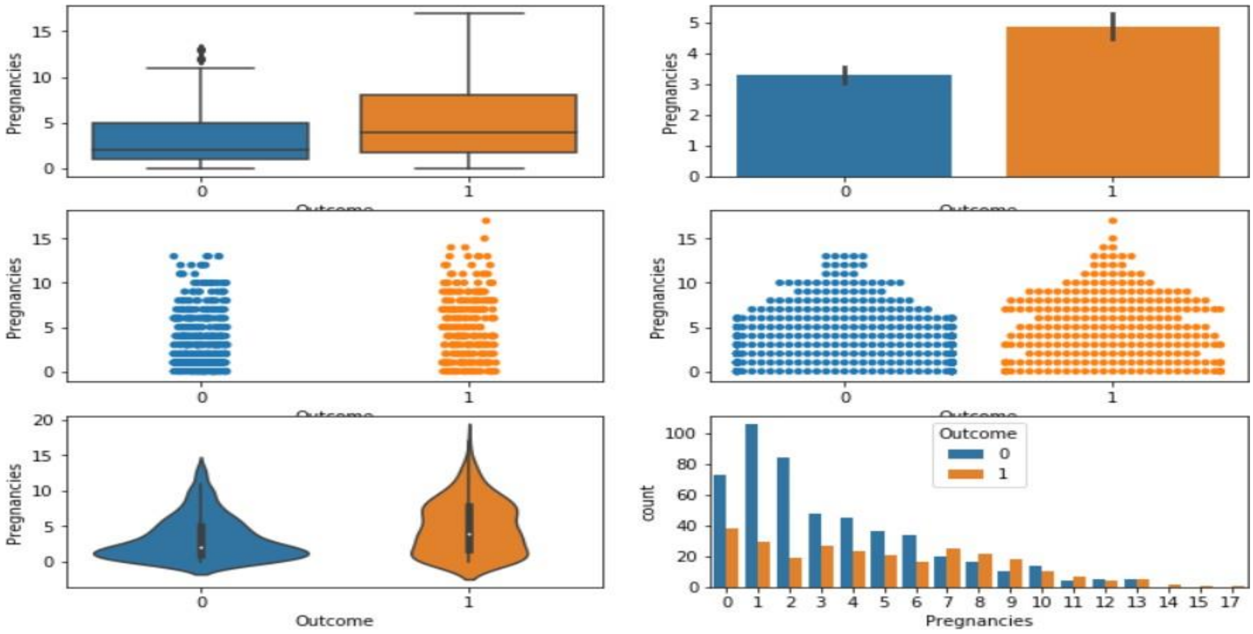


Figure 8: Visualization of Pregnancy

Analysis of “Glucose” parameter with respect to outcome

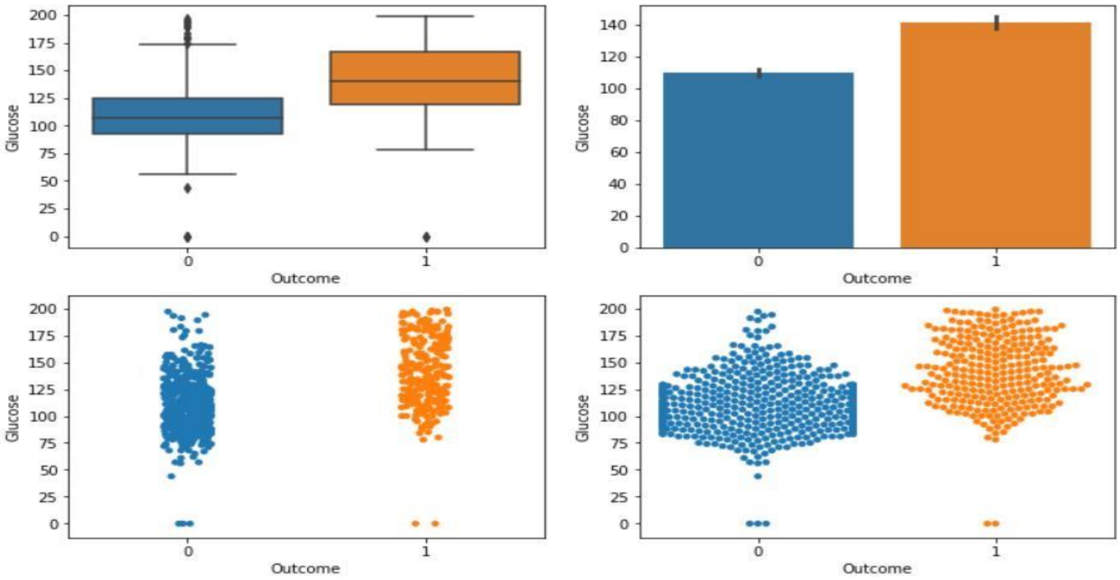


Figure 9: Visualization of Glucose

Analysis of "Blood Pressure" parameter with respect to Outcome

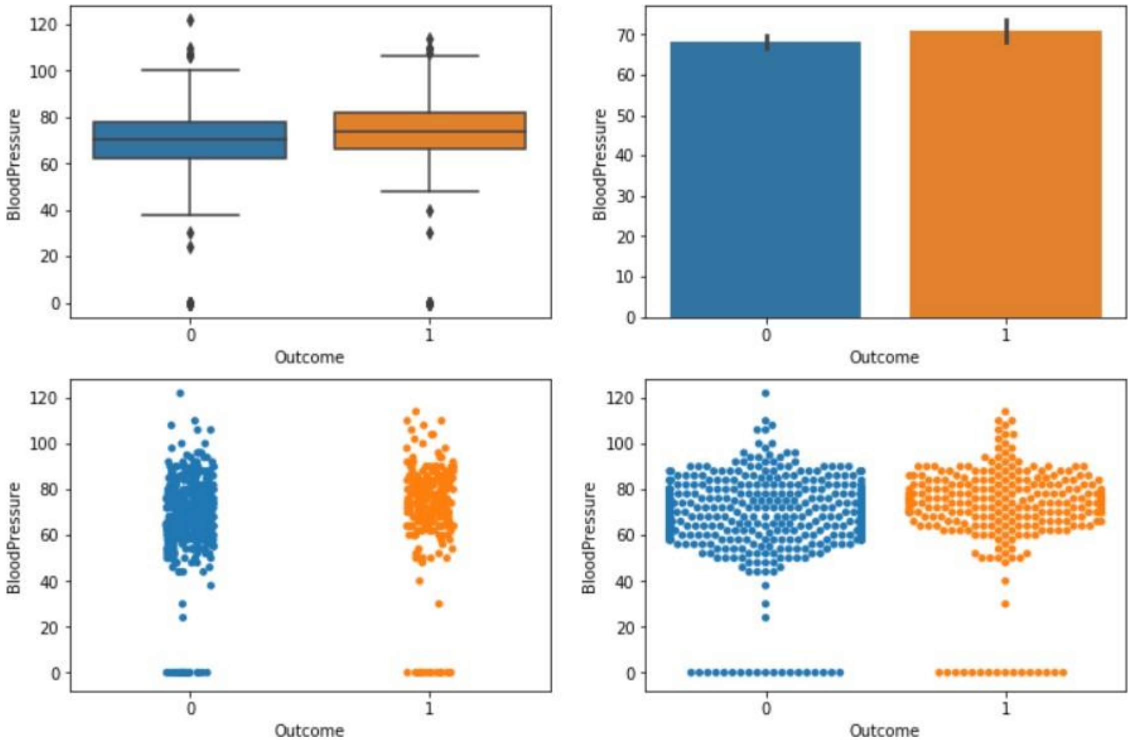


Figure10: **Visualization of Blood Pressure**

Analysis of "Skin Thickness" parameter with respect to Outcome

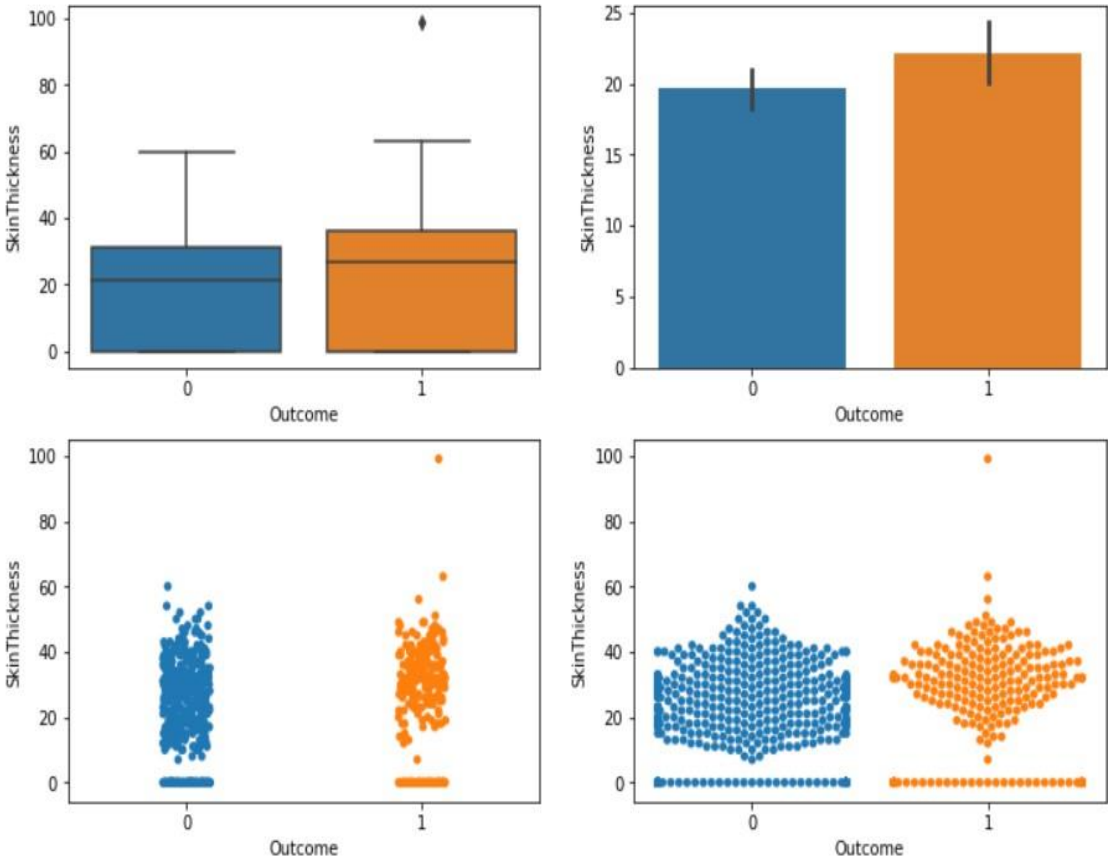


Figure11: **Visualization of Skin Thickness**

Analysis of “Insulin” parameter with Outcome

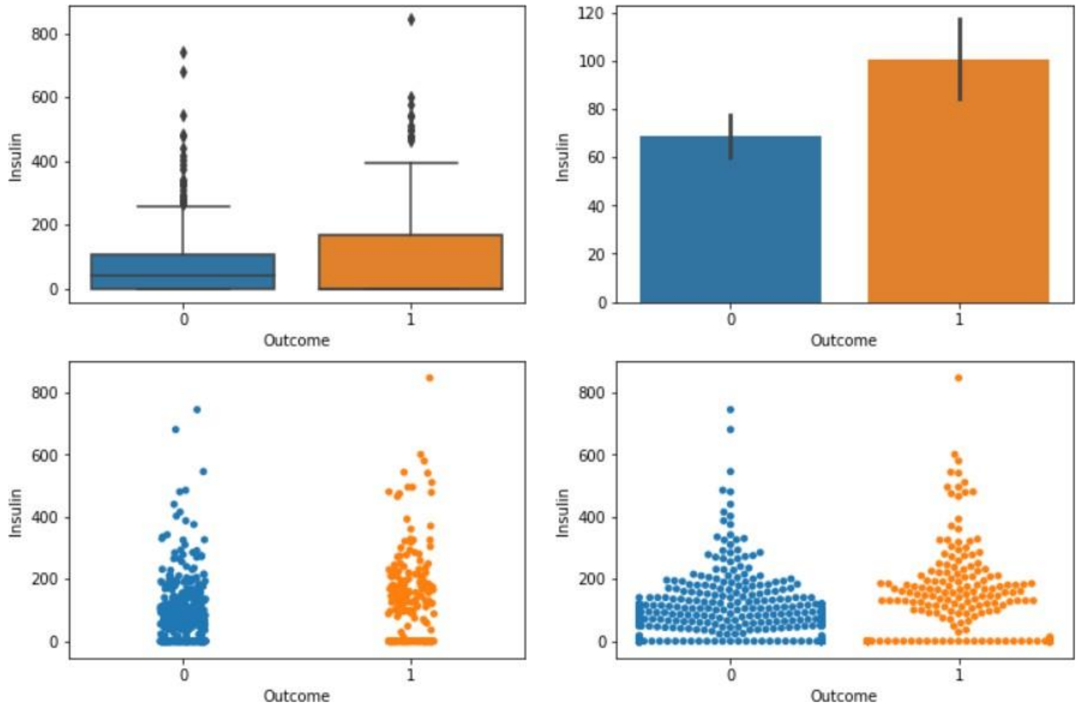


Figure 12: **Visualization of Insulin**

Analysis of “BMI” parameter with respect to Outcome

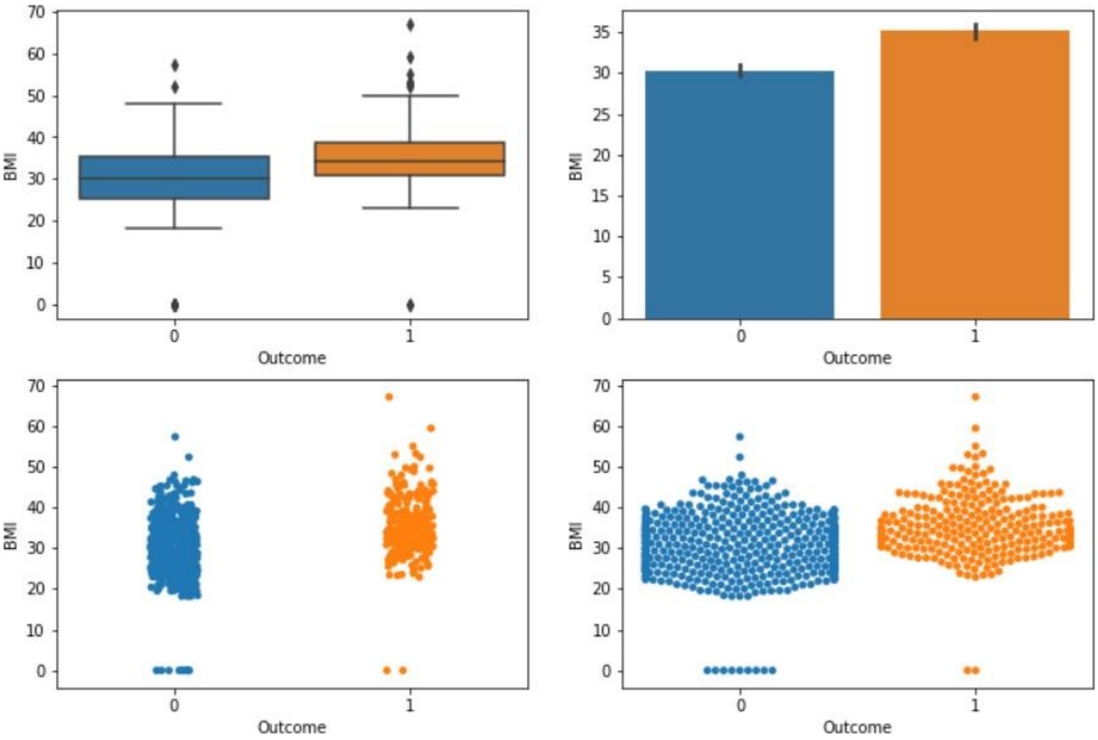


Figure 13: **Visualization of B.M.I**

Analysis of “DiabetesPedigreeFunction Parameter” with Outcome

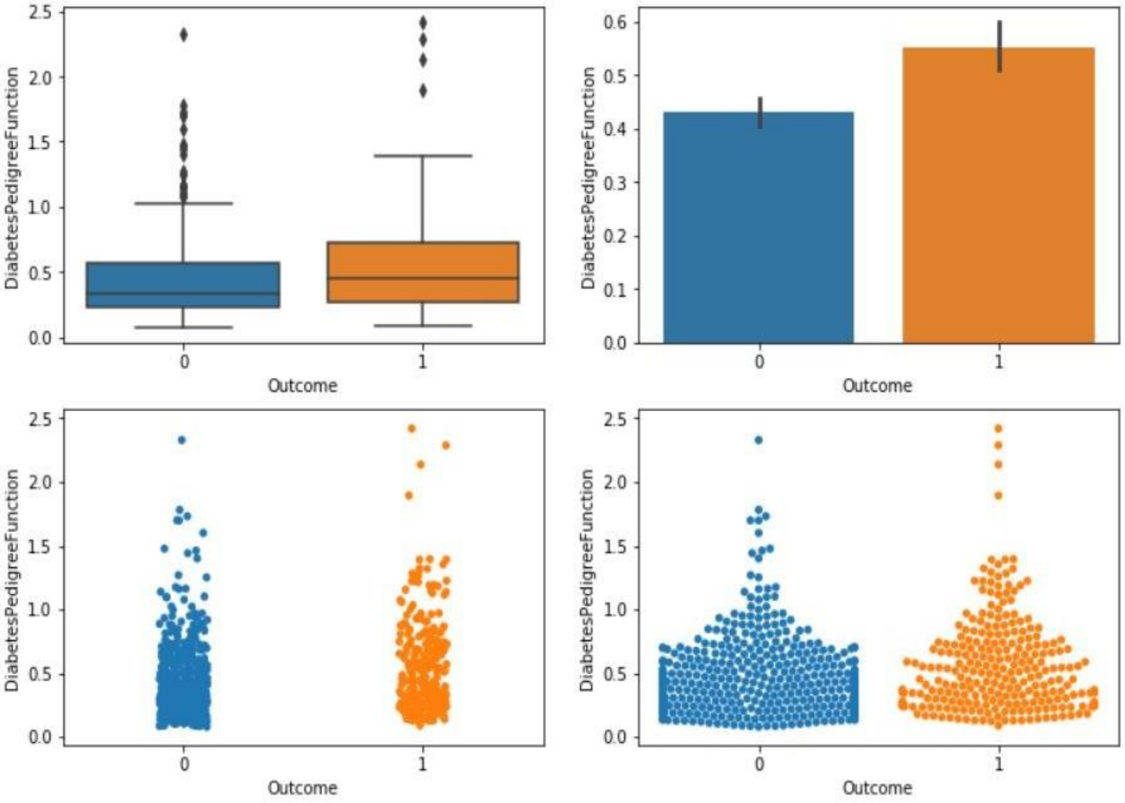


Figure 14: **Visualization of DiabetesPredictionFunction**

Visualizing the Outliners

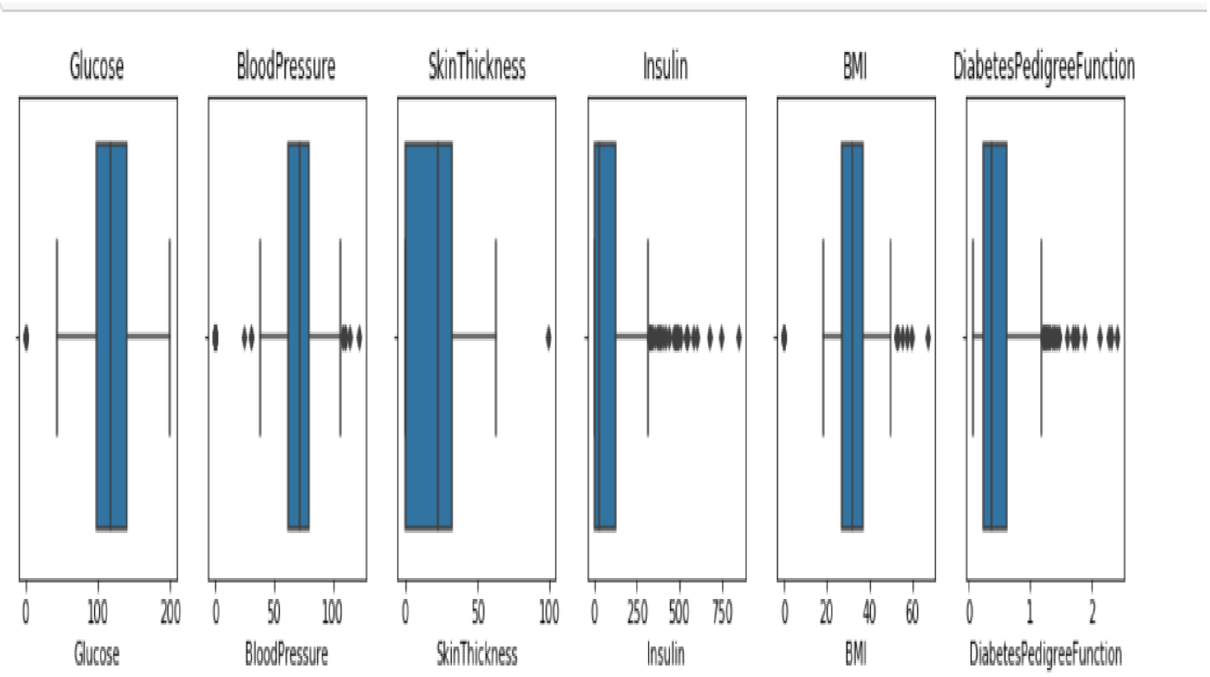


Figure 15: **Visualization of Outliners**

4.2.4 Splitting the Dataset

The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset. The second subset is not used to train the model; instead, the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset is referred to as the test dataset. This has been implemented in our project**.** We have considered 30% our dataset as the test dataset and 70% of our dataset as training dataset.

4.2.5 ALGORITHM USED

In this following project of predicting the probability of having diabetes, we have used the following classification algorithms after the data pre-processing. The algorithms are:

- Naïve Bayes Algorithm
- Logistic Regression
- K Nearest Neighbors
- Decision Tree Classification
- Random Forest
- Support Vector Machine

4.2.5.1 NAÏVE BAYES ALGORITHM

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other. Bayes' Theorem is stated as

$$P\left(\frac{h}{d}\right) = \frac{\left(P\left(\frac{d}{h}\right) * P(h)\right)}{P(d)}$$

Where

❖ P (*h d*) is the probability of hypothesis h given the data d. This is called the posterior probability.

❖ P (*d h*) is the probability of data d given that the hypothesis h was true.

❖ P (h) is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.

❖ P (d) is the probability of the data (regardless of the hypothesis).

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values.

It is called naive Bayes or idiot Bayes because the calculation of the probabilities for each hypothesis is simplified to make their calculation tractable. Rather than attempting to calculate the values of each attribute value P (d1, d2, d3|h), they are assumed to be conditionally independent given the target value and calculated as P (d1|h) * P (d2|H) and so on.

This is a very strong assumption that is most unlikely in real data, i.e. that the attributes do not interact. Nevertheless, the approach performs surprisingly well on data where this assumption does not hold.

4.2.5.2 LOGISTIC REGRESSION

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes. Sigmoid activation• In order to map predicted values to probabilities, we use the sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

$$S(z) = \frac{1}{1 + e^{z}}$$

- **Decision boundary**

 Our current prediction function returns a probability score between 0 and 1. In order to map this to a discrete class (true/false, cat/dog), we select a threshold value or tipping point above which we will classify values into class 1 and below which we classify values into class 2. $p \geq 0.5$, class=1

$p < 0.5$, class=0

For example, if our threshold was .5 and our prediction function returned .7, we would classify this observation as positive. If our prediction was .2 we would classify the observation as negative. For logistic regression with multiple classes we could select the class with the highest predicted probability.

- **Making predictions**

Using our knowledge of sigmoid functions and decision boundaries, we can now write a prediction function. A prediction function in logistic regression returns the probability of our observation being positive, true, or “Yes”. We call this class 1 and its notation is P (class=1) P (class=1). As the probability gets closer to 1, our model is more confident that the observation is in class 1.

- **COST FUNCTION**

Since the hypothesis function for logistic regression is sigmoid in nature hence, The First important step is finding the gradient of the sigmoid function. We can see from the derivation below that gradient of the sigmoid function follows a certain pattern.

$$h_\theta(x) = \frac{1}{1+e^{(-\theta^T x)}}$$

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h\_\theta(x^{(i)}), y^{(i)})$$

$$Cost(h_\theta(x), y) = -log\big(h_\theta(x)\big) \qquad if\ y = 1$$

$$Cost(h_\theta(x), y) = -log\big(1 - h_\theta(x)\big) \quad if\ y = 0$$

This is the required cost function for the logistic regression.

4.2.5.3 K NEAREST NEIGHBORS

K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection. It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data. We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute. As an example, consider the following table of data points containing two features:

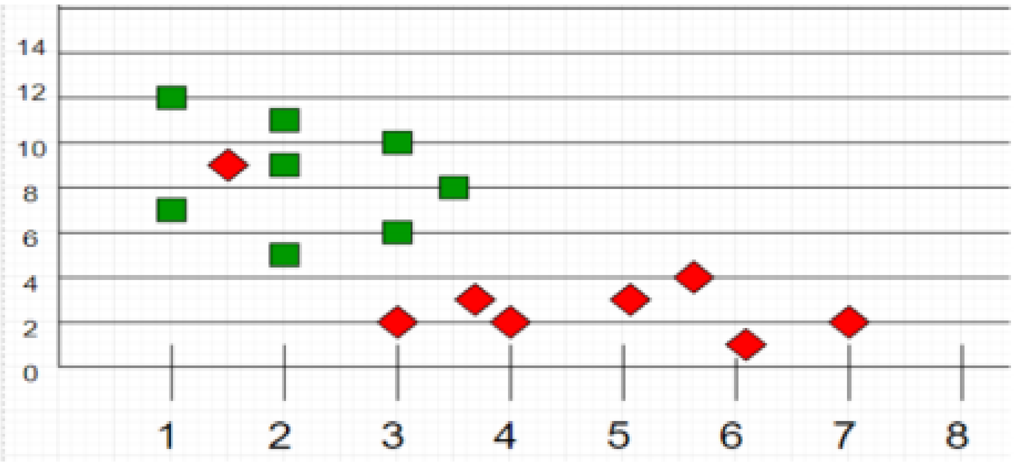


Figure 16: Representing KNN on training dataset

Now, given another set of data points (also called testing data), allocate these points a group by analyzing the training set. Note that the unclassified points are marked as ‘White’

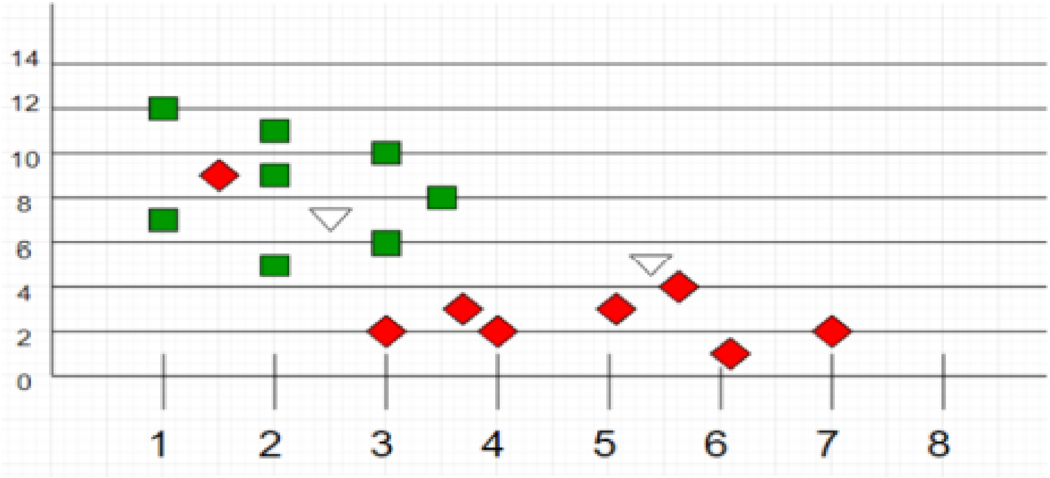


Figure 17: Representing KNN on testing dataset

If we plot these points on a graph, we may be able to locate some clusters or groups. Now, given an unclassified point, we can assign it to a group by observing what group its nearest neighbors belong to. This means a point close to a cluster of points classified as ‘Red’ has a higher probability of getting classified as ‘Red’. Intuitively, we can see that the first point (2.5, 7) should be classified as ‘Green’ and the second point (5.5, 4.5) should be classified as ‘Red’. K can be kept as an odd number so that we can calculate a clear majority in the case where only two groups are possible (e.g. Red / blue). With increasing K, we get smoother, more defined boundaries across different classifications. Also, the accuracy of the above classifier increases as we increase the number of data points in the training set.

4.2.5.4 DECISION TREE CLASSIFIER

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

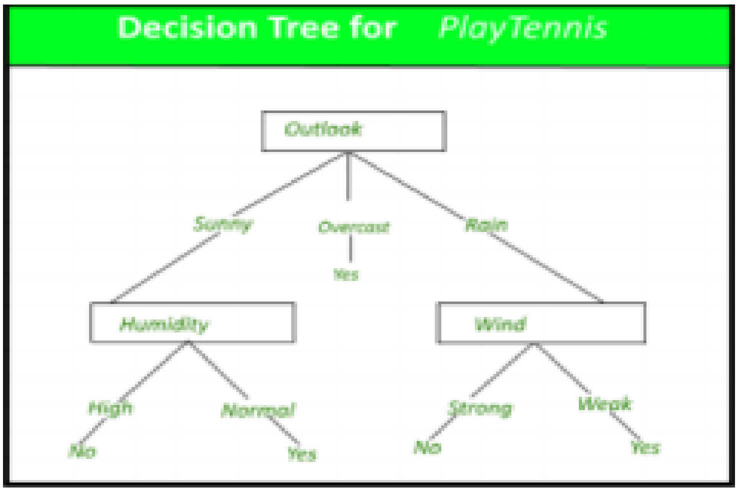


Figure 18: Decision Tree Example

A classification model is typically used to:

- Predict the class label for a new unlabeled data object.
- Provide a descriptive model explaining what features characterize objects in each class.

A tree can be “learned” by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions.

The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. In general decision tree classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification.

4.2.5.5 RANDOM FOREST CLASSIFIER

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees The Random Forest (RF) classifiers are suitable for dealing with the high dimensional noisy data in text classification. An RF model comprises a set of decision trees each of which is trained using random subsets of features. Given an instance, the prediction by the RF is obtained via majority voting of the predictions of all the trees in the forest.

However, different test instances would have different values for the features used in the trees and the trees should contribute differently to the predictions. This diverse contribution of the trees is not considered in traditional RFs. Many approaches have been proposed to model the diverse contributions by selecting a subset of trees for each instance. Random forest is like bootstrapping algorithm with Decision tree (CART) model.

Say, we have 1000 observation in the complete population with 10 variables. Random forest tries to build multiple CART models with different samples and different initial variables. For instance, it will take a random sample of 100 observation and 5 randomly chosen initial variables to build a CART model. It will repeat the process (say) 10 times and then make a final prediction on each observation. Final prediction is a function of each prediction. This final prediction can simply be the mean of each prediction.

Step 1 − First, start with the selection of random samples from a given dataset.

Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

Step 3 - In this step, voting will be performed for every predicted result.

Step 4 − At last, select the most voted prediction result as the final prediction result.

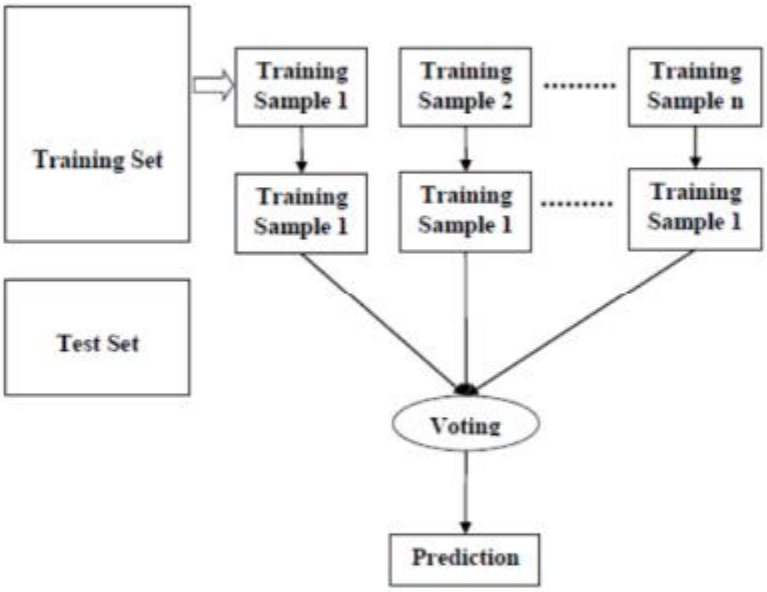


Figure 19: Working of Random Forest

CHAPTER 5

EXPERIMENTAL AND RESULT ANALYSIS

In this section we are going to deal with testing, testing is finding out how well something works. In terms of human beings, testing tells what level of knowledge or skill has been acquired. In computer hardware and software development, testing is used at key checkpoints in the overall process to determine whether objectives are being met. There are various techniques for testing the accuracy but we are going to use some of them.

5 .1 ACCURACY

It is the most common evaluation metric for classification problems. It is defined as the number of correct predication as against the number of total predictions. However, this metric alone cannot give enough information to decide whether the model is a good one or not. It is suitable when there are equal numbers of observation in every class.

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

$$\textit{Accuracy} = \frac{\textit{Number of correct predictions}}{\textit{Total number of prediction}}$$

5.2 F1_SCORE

The accuracy of a machine learning classification algorithm is one way to measure how often the algorithm classifies a data point correctly. Accuracy is the number of correctly predicted data points out of all the data points. More formally, it is defined as the number of true positives and true negatives divided by the number of true positives, true negatives, false positives, and false negatives. A true positive or true negative is a data point that the algorithm correctly classified as true or false, respectively. A false positive or false negative, on the other hand, is a data point that the algorithm incorrectly classified. For example, if the algorithm classified a false data point as true, it would be a false positive. Often, accuracy is used along with precision and recall, which are other metrics that use various ratios of true/false positives/negatives. Precision and recall are two numbers which together are used to evaluate the performance of classification or information retrieval systems. Precision is defined as the fraction of relevant instances among all retrieved instances. Recall, sometimes referred to as ‘sensitivity, is the fraction of retrieved instances among all relevant instances. Perfect classifiers have precision and recall both equal to 1. Precision and Recall Formulas:

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP + FP}}$$

$$\boldsymbol{Recall} = \frac{\mathbf{TP}}{\mathbf{TP + FN}}$$

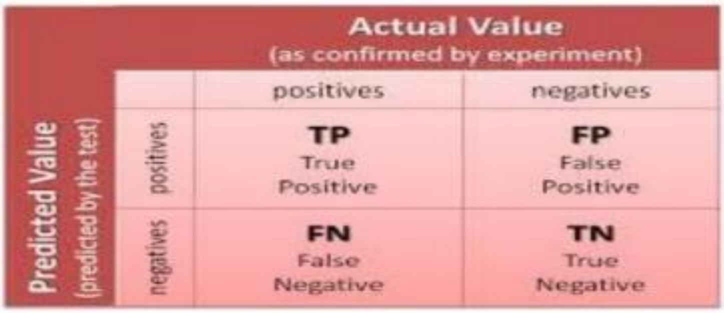
The F-score, also called the F1-score, is a measure of a model’s accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into ‘positive’ or ‘negative’. The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model’s precision and recall. The F-score is commonly used for evaluating information retrieval systems such as search engines, and also for many kinds of machine learning models, in particular in natural language processing. The formula for the standard F1-score is the harmonic mean of the precision and recall. A perfect model has an F-score of 1.

$$\mathbf{F_1} = \mathbf{2} * \frac{\mathbf{precision} * \mathbf{recall}}{\mathbf{precision} + \mathbf{recall}}$$

$$\mathbf{F_1} = \frac{\mathbf{TP}}{\mathbf{TP} + \frac{1}{2}\,(\mathbf{FP + FN})}$$

5.3 CONFUSION MATRIX

It is also known as Error matrix, which is a table representation that shows the performance of the model. It is special kind of Contingency table 21 having two dimensions- “actual”, labeled on x-axis and “predicted” on y-axis. The cells of the table are the number of predictions made by the algorithm. True Positives: It is correctly predicted positive values. True Negatives: It is correctly predicted negative values. False Positives: It is incorrectly predicted negative values as positive values. False Negatives: It is incorrectly predicted negative values as positive values.



		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	TP True Positive	FP False Positive
	negatives	FN False Negative	TN True Negative

Figure 20: Confusion Matrix

5.4 RESULT ANALYSIS

We have performed six algorithms on our dataset namely Naïve Bayes algorithm, K Nearest Neighbors, Logistic Regression, Passive Aggressive Classifier, Decision tree and Random forest. And here we have tabulated the results.

5.4.1 ACCURACY SCORE

NAME OF THE ALGORITHM	ACCURACY (IN %)
NAÏVE BAYES	59.7
LOGISTIC REGRESSION	76.6
K- NEAREST NEIGHBORS	73.5
DECISION TREE	72.2
RANDOM FOREST	77.9
SUPPORT VECTOR MACHINE	75.32

Table 2: Table for accuracy

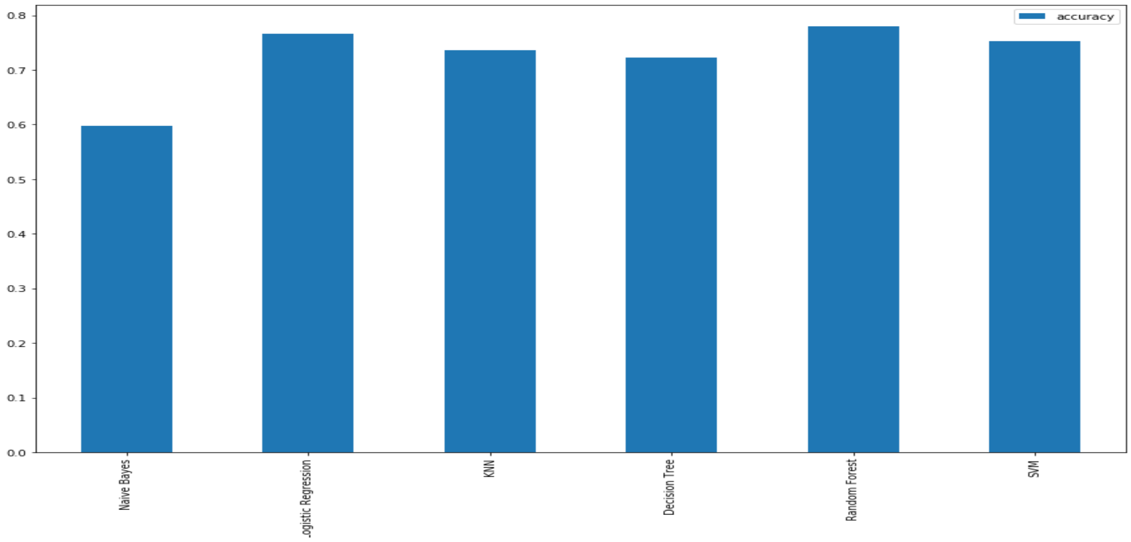
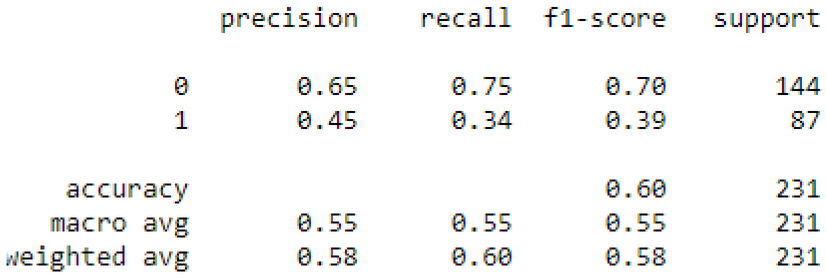


Figure 21: Bar plot for Accuracy

5.4.1 CLASSIFICATION REPORT

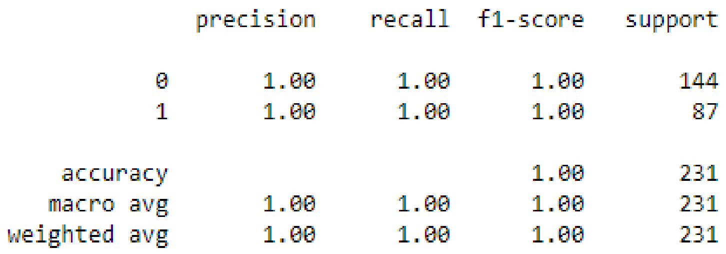
- For Naïve Bayes



```
              precision    recall  f1-score   support

           0       0.65      0.75      0.70       144
           1       0.45      0.34      0.39        87

    accuracy                           0.60       231
   macro avg       0.55      0.55      0.55       231
weighted avg       0.58      0.60      0.58       231
```

Figure 22: Classification Report for Naïve Bayes

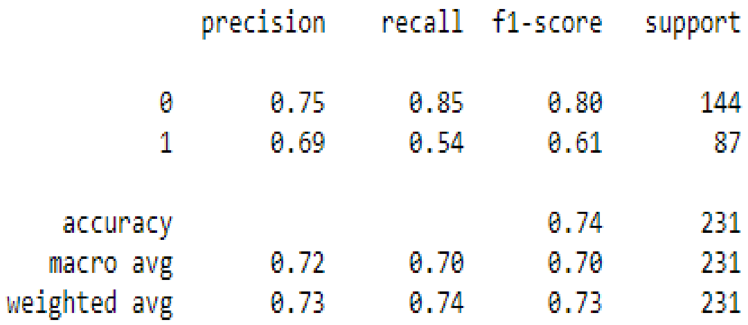
- For Logistic Regression



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       144
           1       1.00      1.00      1.00        87

    accuracy                           1.00       231
   macro avg       1.00      1.00      1.00       231
weighted avg       1.00      1.00      1.00       231
```

Figure 23: Classification Report for Logistic Regression

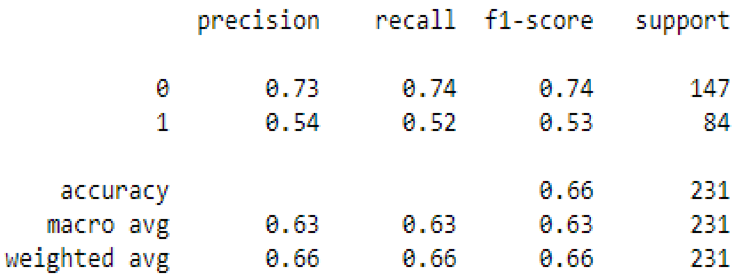
- For K- Nearest Neighbors



```
              precision    recall  f1-score   support

           0       0.75      0.85      0.80       144
           1       0.69      0.54      0.61        87

    accuracy                           0.74       231
   macro avg       0.72      0.70      0.70       231
weighted avg       0.73      0.74      0.73       231
```

Figure 24: Classification Report for K-nearest neighbors

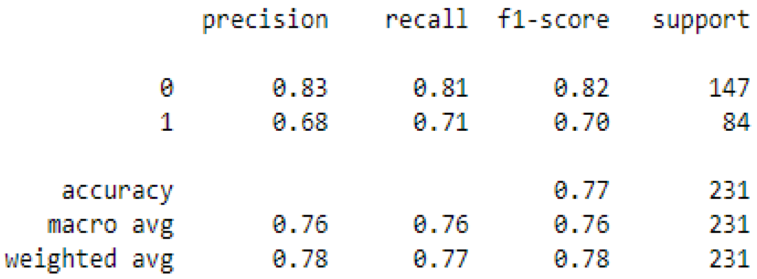
- For Decision Tree



```
              precision    recall  f1-score   support

           0       0.73      0.74      0.74       147
           1       0.54      0.52      0.53        84

    accuracy                           0.66       231
   macro avg       0.63      0.63      0.63       231
weighted avg       0.66      0.66      0.66       231
```

Figure 25: Classification Report for Decision Tree

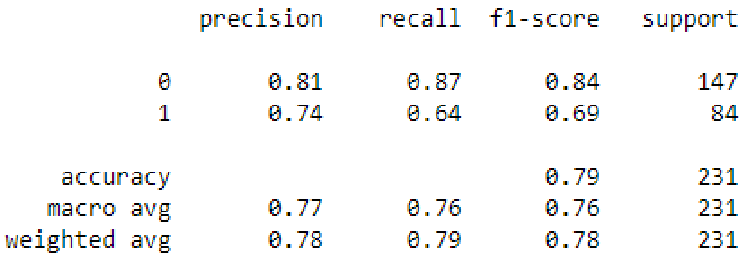
- For Random Forest



```
              precision    recall  f1-score   support

           0       0.83      0.81      0.82       147
           1       0.68      0.71      0.70        84

    accuracy                           0.77       231
   macro avg       0.76      0.76      0.76       231
weighted avg       0.78      0.77      0.78       231
```

Figure 26: Classification Report for Random Forest

- For Support Vector Machine



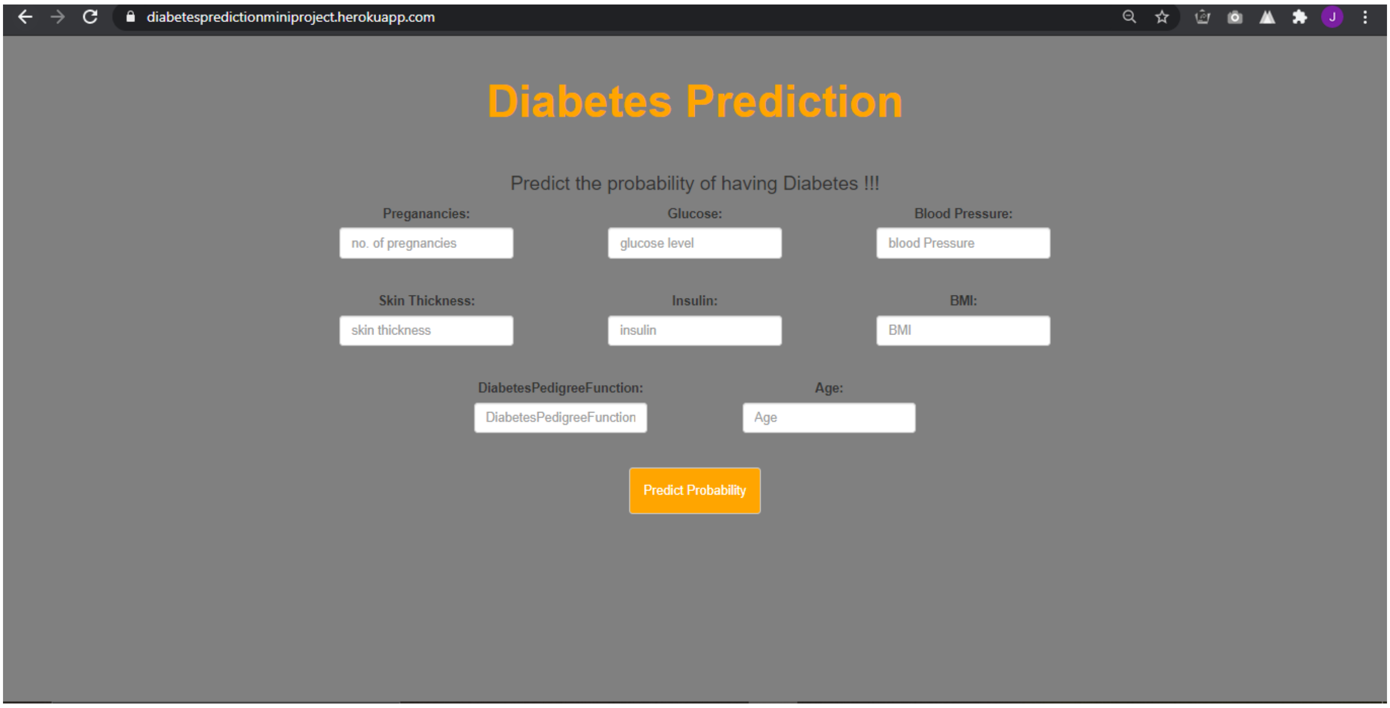
```
              precision    recall  f1-score   support

           0       0.81      0.87      0.84       147
           1       0.74      0.64      0.69        84

    accuracy                           0.79       231
   macro avg       0.77      0.76      0.76       231
weighted avg       0.78      0.79      0.78       231
```

Figure 27: Classification Report for Support Vector Machine

CHAPTER 6

USER INTERFACE AND SOFTWARE TESTING

After developing our machine learning model, training it with the available data-set and testing it with the test dataset we have received satisfactory accuracy and so to convert the model to be implemented as a application we have used a flask framework to provide the user interface to the model. And deployed the model for use and testing.



diabetespredictionminiproject.herokuapp.com

Diabetes Prediction

Predict the probability of having Diabetes !!!

Preganancies: no. of pregnancies

Glucose: glucose level

Blood Pressure: blood Pressure

Skin Thickness: skin thickness

Insulin: insulin

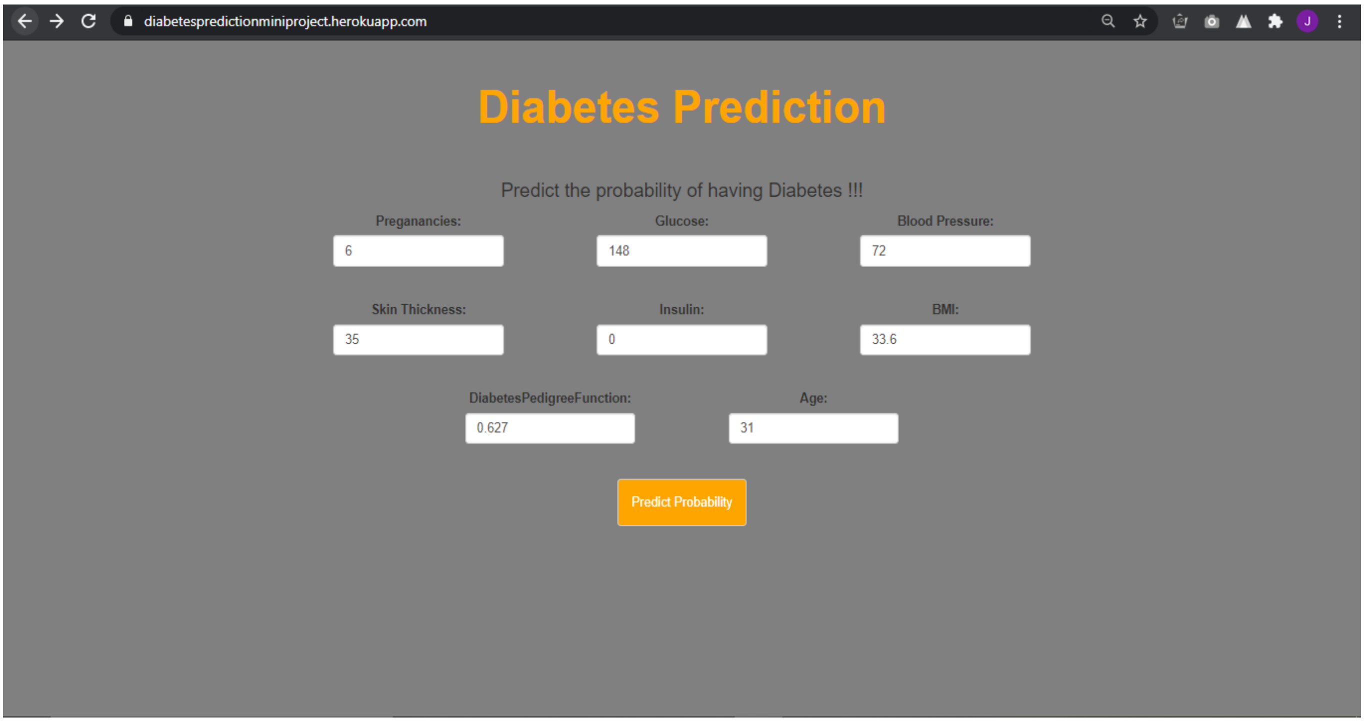
BMI: BMI

DiabetesPedigreeFunction: DiabetesPedigreeFunction

Age: Age

Figure 28: Home Screen of deployed Model

This is the first screen that appears at our deployed model and asks for the input which is to be tested.



diabetespredictionminiproject.herokuapp.com

Diabetes Prediction

Predict the probability of having Diabetes !!!

Preganancies: 6

Glucose: 148

Blood Pressure: 72

Skin Thickness: 35

Insulin: 0

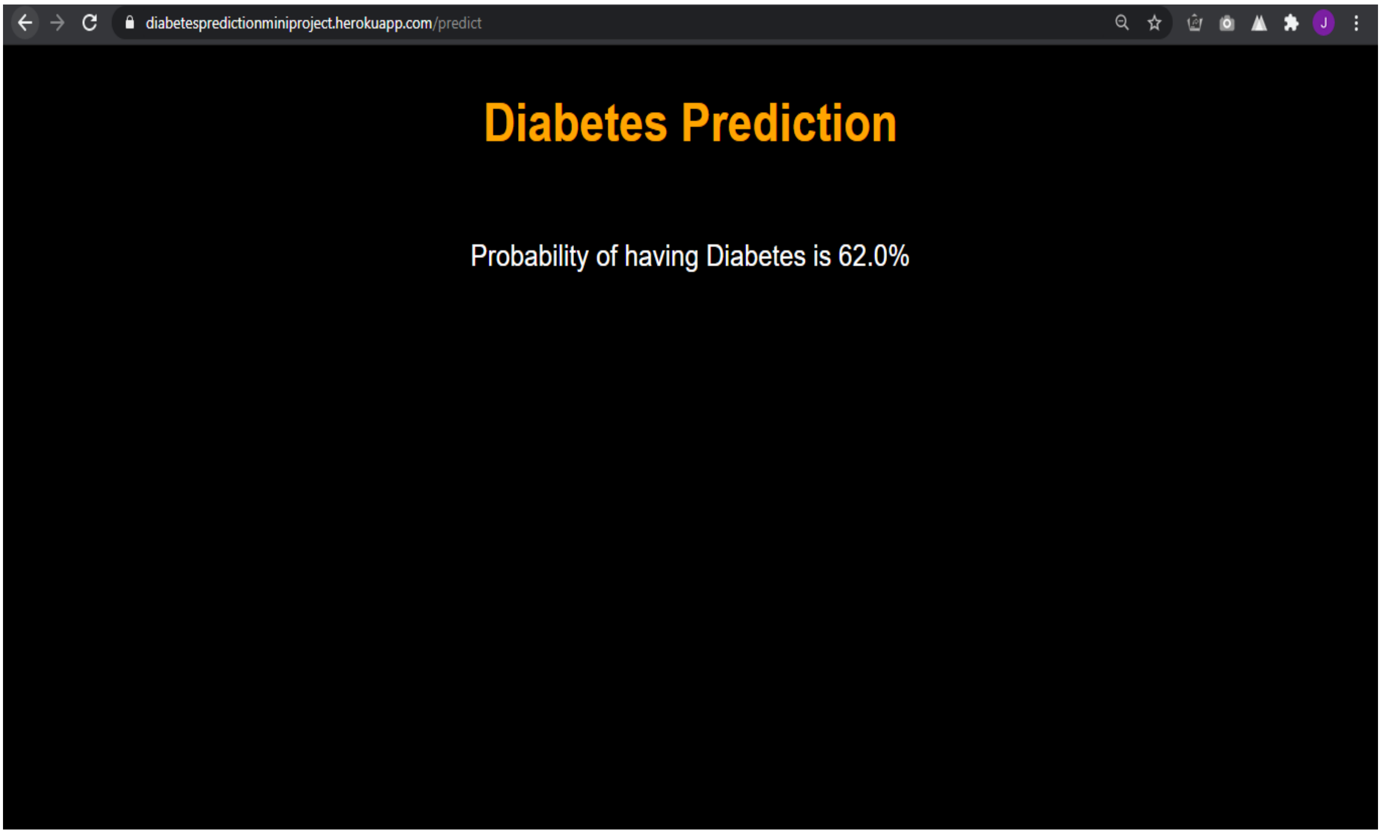
BMI: 33.6

DiabetesPedigreeFunction: 0.627

Age: 31

Figure 29: Inputting the features to detect

After we enter the features inside the box and click on the predict the result appears.



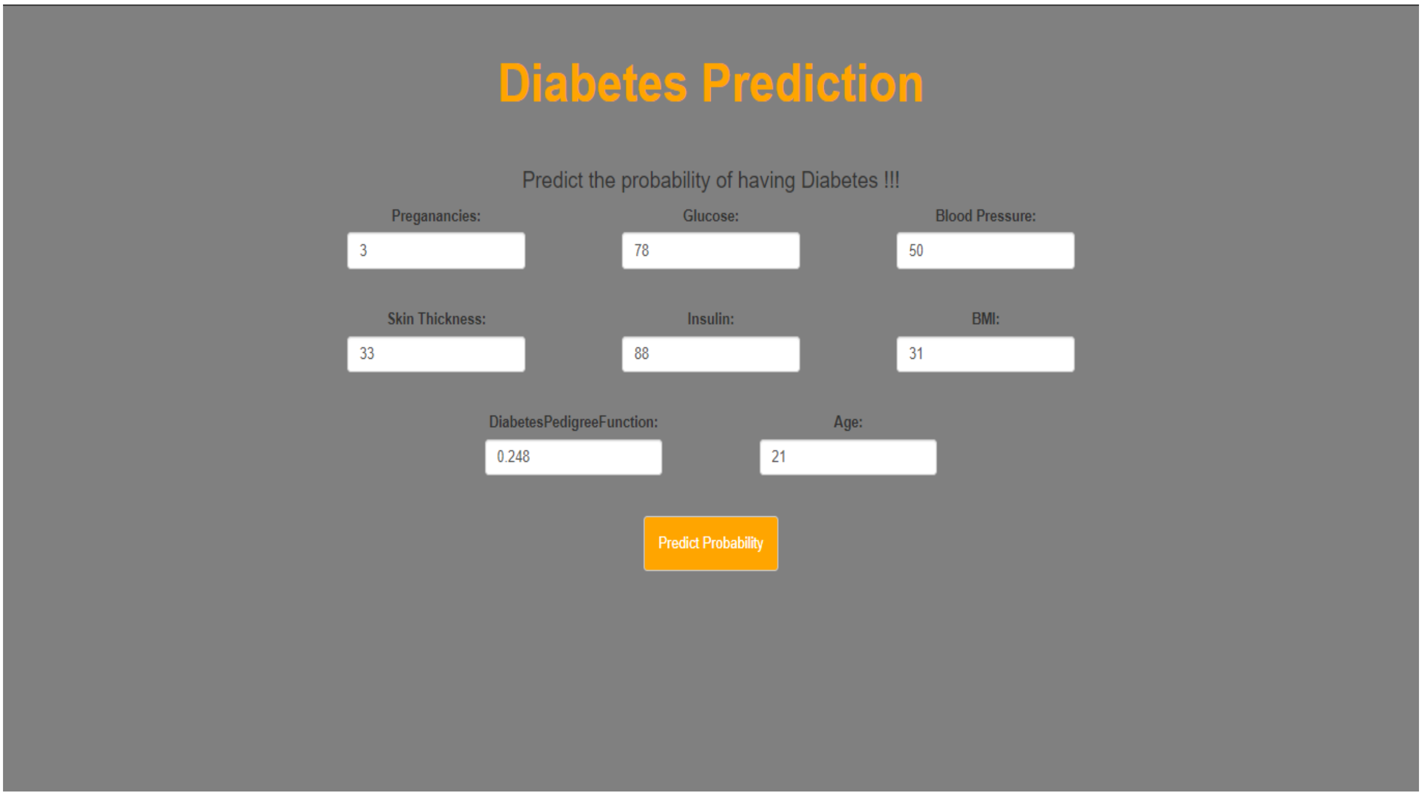
diabetespredictionminiproject.herokuapp.com/predict

Diabetes Prediction

Probability of having Diabetes is 62.0%

Figure 30: Probability of having Diabetes

Another example: Features Entered



Diabetes Prediction

Predict the probability of having Diabetes !!!

Preganancies: 3

Glucose: 78

Blood Pressure: 50

Skin Thickness: 33

Insulin: 88

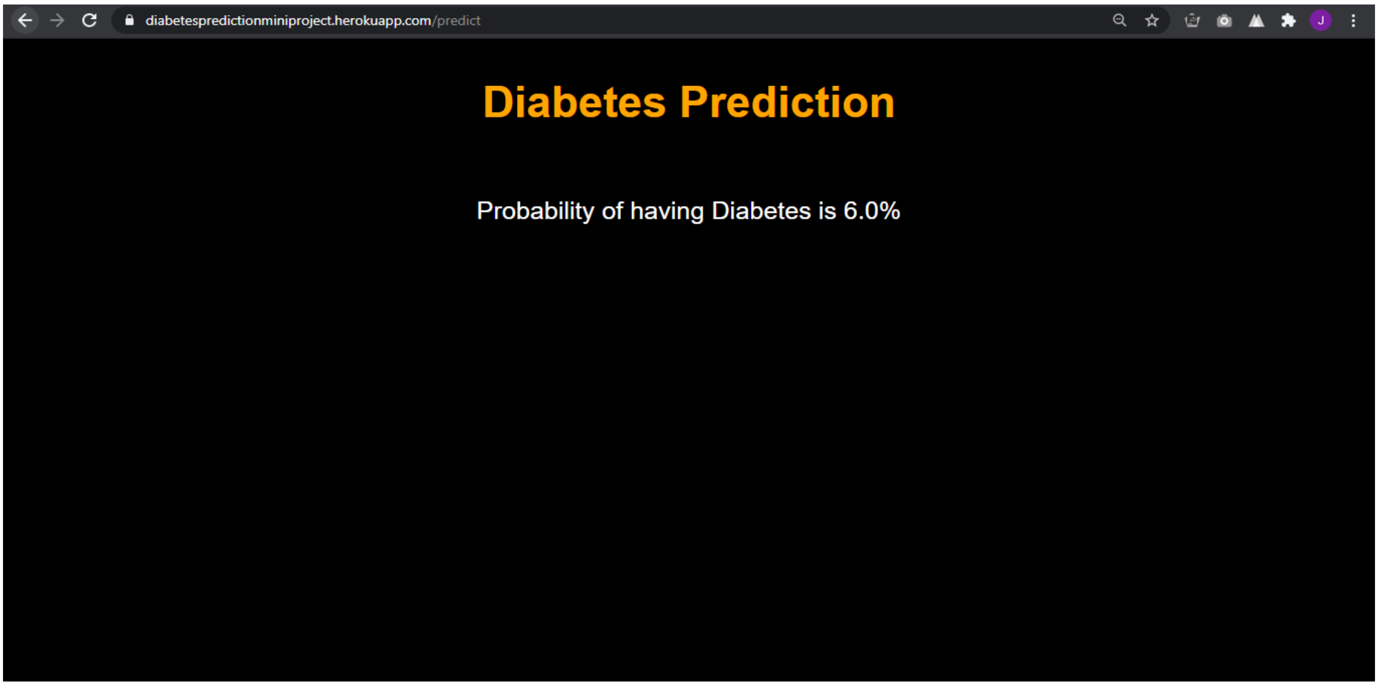
BMI: 31

DiabetesPedigreeFunction: 0.248

Age: 21

Figure 31: Inputting the features to detect

The result will be printed as:-



Diabetes Prediction

Probability of having Diabetes is 6.0%

Figure 32: Probability of having Diabetes

CHAPTER-7
CONCLUSION

- The dataset have nine attributes (parameters) in which there are eight independent variables (Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, Age) and one dependent variable (Outcome).
- BMI and Diabetes Pedigree Function are a float data type and other parameters areinteger data type.
- The parameters do not contain any null values (missing values). However, this cannot be true. As Insulin, Skin Thickness, Blood Pressure, BMI, Glucose have zero values.
- The Outcome parameter shows that there are 500 healthy people and 268 Diabeticpeople. It means that 65% people are diabetic and 34.9% people are healthy.
- The parameters Glucose, Blood Pressure, BMI are normally distributed. Pregnancies, Insulin, Age, and Diabetes Pedigree Function are rightly skewed.
- The missing values '0' is replaced by the mean of the parameter to explore the dataset.
- Blood Pressure, Skin Thickness, Insulin, BMI have outliers.
- There is no convincing relationship between the parameters. Pregnancies and age have some kind of a linear line. Blood Pressure and age have little relation. Most of the aged people have Blood Pressure. Insulin and Glucose have some relation.
- Glucose, Age BMI and Pregnancies are the most Correlated features with the Outcome. Insulin and Diabetes Pedigree Function have little correlation with the outcome. Blood Pressure and Skin Thickness have tiny correlation with the outcome.
- Age and Pregnancies, Insulin and Skin Thickness, BMI and Skin Thickness, Insulin and Glucose are little correlated

  At last by using all these five machine learning algorithms we had measured different parameters within the dataset and we had came through better accuracy rate with random forest with nearly 78%. This work can be extended by adding any other algorithm which can give better accuracy than random forest.

REFERENCES

- Mani Butwall and Shraddha Kumar,” A Data Mining Approach for the Diagnosis of Diabetes Mellitus using Random Forest Classifier”, International Journal of Computer Applications, Volume 120 - Number 8,2015.
- K.Rajesh and V. Sangeetha, “Application of Data Mining Methods and Techniques for Diabetes Diagnosis”, International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 3, September 2012.
- Naive Bayes for Diabetes Prediction, Medium Corporation,2018, https://medium.com/@martinpella/naivelink -Bayes for-diabetes-prediction