Ai based Diabetes prediction system :

Introduction

Diabetes is a widespread and life-altering disease that affects millions of individuals worldwide. Early detection and accurate prediction of diabetes can play a pivotal role in preventing its onset and managing the condition effectively. The importance of reliable diabetes prediction has prompted the development of AI-based systems that leverage machine learning techniques to assess an individual's risk of developing diabetes. This project is an exploration of such a system, aiming to create an AI-based diabetes prediction model that can provide valuable insights into an individual's susceptibility to diabetes. The development of this system has progressed through various phases, encompassing data collection, preprocessing, feature selection, and machine learning model creation.

The primary objective of this project is to design, build, and evaluate an AI-based system capable of predicting the likelihood of a patient developing diabetes based on diagnostic measurements. The need for such a system arises from the growing prevalence of diabetes and the desire to enhance early diagnosis and intervention.

This document serves as a comprehensive account of our journey to develop this AI-based diabetes prediction system. It includes detailed descriptions of the data used, the methodology employed, the choice of machine learning algorithm, model training, evaluation metrics, and any innovative techniques or approaches we adopted during the project.

ABSTRACT

Diabetes is a common disease caused by a set of metabolic ailments where the sugar stages over a drawn-out period is high. It touches the diverse organs of the human body's system, in precise the blood strains and nerves. Early prediction in such a disease can be exact and save human life. However, it is not possible to monitor patients every day in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more patience, time, and expertise. To achieve the goal, this research work mainly discovers numerous factors associated with this disease using machine learning techniques. The early prognosis of diabetes patients can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine.

Keywords: Machine Learning, Random Forest Classification, Accuracy, Recall, Precision, Flask

LIST OF FIGURES:

LIST OF TABLES:

Table of Contents

CHAPTER 1: INTRODUCTION

Diabetes is a disease that affects the hormone insulin, follow-on in abnormal metabolism of carbohydrates, and advanced steps of sugar in the blood. This great blood sugar affects several organs of the human body which in turn complicates many sources of the body, in precise the blood strains and nerves. The details of diabetes are not nevertheless totally exposed, many researchers supposed that both the hereditary elements and environmental effects are complex therein. As exposed by the International Diabetes Federation[1], the extent of people having diabetes stretched 422 million out of 2021 that makes up 5.34 % of the world's total adult population. Early prediction of such diseases can be controlled over the diseases and save human life. To accomplish this goal, this research work mainly discovers the early prediction of diabetes by taking into account various risk factors related to this disease. For the willpower of the study, we gathered a diagnostic dataset having 16 attributes diabetic of different patients. Later, we debate about these attributes with their conforming values. Based on these attributes, we figure a prediction model by means of various machine learning techniques to predict diabetes. Machine Learning techniques provide well-organized results to extract knowledge by making prediction models from diagnostic medical datasets composed of diabetic patients. Though it is difficult to select the best techniques to predict based on such attributes, thus for the determination of the study different algorithms have been used for the model prediction.

1.1 Problem Definition

The major challenge in predicting diabetes cases is its detection. There are instruments available that can predict diabetes but either they are expensive or are not efficient to calculate the chance of diabetes in humans. Early detection of diabetes can decrease the mortality rate and overall complications. However, it is not possible to monitor patients every day in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more patience, time, and expertise. Since we have a good amount of data in today's world, we can use various machine learning algorithms to analyze the data for hidden patterns. The hidden patterns can be used for health diagnosis in medicinal data.

1.2 Motivation

Machine learning techniques have been around us and have been compared and used for analysis for many kinds of data science applications. This project is carried out with the motivation to develop an appropriate computer-based system and decision support that can aid in the early detection of diabetes, in this project we have developed a model which classifies if the patient will have diabetes based on various features (i.e. potential risk factors that can cause diabetes) using random forest classifier. Hence, the early prognosis of diabetes can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine.

1.3 Objectives

The main objective of developing this project are:

- Exploratory Data Analysis of PIMA Indian diabetes dataset.
- Train and evaluate a machine learning model to detect the possible cases of diabetes.
- To analyze the significant risk factors based on PIMA dataset which may lead to diabetes.
- To deploy the trained model using a web framework so that the model is accessible through the web browser.

CHAPTER 2: LITERATURE REVIEW

Various researchers have been shown revision in the area of diabetes by using machine learning techniques to extract knowledge from existing medical data. For illustration, Marina Skurichina and Ludmila Kuncheva and Robert P W Duin. Bagging and Boosting for the Nearest Mean Classifier: Effects of Sample Size on Diversity and Accuracy[2]. Michael Lindenbaum and Shaul Markovitch and Dmitry Rusakov. Selective Sampling Using Random Field Modelling[3]. Michael Lindenbaum and Shaul Markovitch and Dmitry Rusakov. Selective Sampling Using Random Field Modelling[4]. In this work, we examine real diagnostic medical data based on numerous risk factors using popular machine learning classification techniques to assess their performance for predicting diabetes cases.

CHAPTER 3: DATASETS

In this work, Pima Indian Diabetes Dataset[1] has been used. The dataset was collected among the Pima Indian female population near Phoenix, Arizona. This particular dataset has been widely used in machine learning experiments and is currently available through the UCI repository of standard datasets. This population has been studied continuously by the National Institute of Diabetes, Digestive, and Kidney. UCI repository contains 768 instances of observations and total 9 attributes with no missing values reported. Datasets contain 8 particular variables which were considered high-risk factors for the occurrence of diabetes and 1 target variable containing '1' for diabetic and '0' for non-diabetic patients. The 8 feature variables along with the target variable are shown in the following table (Table 1).

No	Name	Description	Type
1	Pregnancies	Number of times pregnant	Numeric
2	Glucose	Plasma glucose concentration a 2 hours in an oral glucose tolerance test Numeric	Numeric
3	Blood Pressure	Diastolic blood pressure	Numeric
4	Skin Thickness	Triceps skinfold thickness	Numeric
5	Insulin	2-hour serum insulin	Numeric
6	BMI	Body mass index	Numeric
7	DiabetesPedigreeFunction	Diabetes pedigree function	Numeric
8	Age	Patient's age	Numeric
9	Outcome	Target variable (1 if diabetic, else 0)	Binary (0 or 1)

Table 1. Dataset Columns Description

	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
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

Figure 1: Original Dataset Snapshot

CHAPTER 4: METHODS AND ALGORITHMS USED

The main purpose of designing this system is to prognose the risk of future diabetes. We have used various methods and algorithms throughout the different phases of the machine learning pipeline which will be discussed below.

4.1 Synthetic Minority Over-sampling Technique (SMOTE)

Our dataset is imbalanced. The number of samples belonging to the non-diabetic class (500) is more than the number of samples belonging to the diabetic class (268). A machine learning algorithm trained on an imbalanced dataset is biased towards the majority class.
To avoid the dominance of the majority class over the minority class, the samples in the minority class were oversampled using Synthetic Minority Over-sampling Technique (SMOTE).

The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen. The current implementation uses five nearest neighbors. For instance, if the amount of over-sampling needed is 200%, only two neighbors from the five nearest neighbors are chosen and one sample is generated in the direction of each. Synthetic samples are generated in the following way: Take the difference between the feature vector (sample) under consideration and its nearest neighbor. Multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration. This causes the selection of a random point along the line segment between two specific features. This approach effectively forces the decision region of the minority class to become more general.[3]

```
(* Compute k nearest neighbors for each minority class sample only. *)
for i ← 1 to T
        Compute k nearest neighbors for i, and save the indices in the nnarray
        Populate(N, i, nnarray)
endfor

Populate(N, i, nnarray) (* Function to generate the synthetic samples. *)
while N ≠ 0
        Choose a random number between 1 and k, call it nn. This step chooses one of
        the k nearest neighbors of i.
        for attr ← 1 to numattrs
                Compute: dif = Sample[nnarray[nn]][attr] − Sample[i][attr]
                Compute: gap = random number between 0 and 1
                Synthetic[newindex][attr] = Sample[i][attr] + gap * dif
        endfor
        newindex++
        N = N − 1
endwhile
return (* End of Populate. *)
End of Pseudo-Code.
```

4.2 Random Forest Classifier

Machine learning (ML) is the study of computer algorithms that improve automatically through experience and by the use of data. Traditional programming approaches take in data and rule to produce the desired output. Whereas, machine learning approaches take in data and the desired output as the input and output the necessary rules.

Supervised Learning Algorithms use the input features along with the target output for training, in contrast to the unsupervised learning algorithms that take only the input features for training. Supervised Learning Algorithms try to find the mapping between the inputs and the outputs. Example: regression, classification.

Classification is a supervised learning problem where the output to be predicted is discrete numeric values representing different classes. For example, predicting whether an image is a dog or a cat if a student will pass an examination or not if a review is positive or negative, and so on.

Random Forest Classifier is an ensemble supervised learning method for classification that operates by constructing a multitude of decision trees at training time. To classify a particular sample, the class predicted by most of the decision trees is selected.

To understand the working of Random Forest Classifiers, we need to have an idea on how decision trees work.

A decision tree spits the feature space recursively to build a classification tree.

Steps:

1. Make a bootstrap table initially.
2. Randomly select m features from T, where m<<T. Here T is the total number of predicted variables and out of total predicted variables, we will select randomly few features.
3. For nodded, calculate the best split point among the m features.
4. Split the node into two daughter nodes using the best split
5. Repeat the firsts three steps until n number of nodes has been reached.
6. Build the model by repeating steps 1 to 5 for D number of times.

CHAPTER 5: METHODOLOGY

5.1 Software Development Model

The prototyping model is a systems development method in which a prototype is built, tested, and then reworked as necessary until an acceptable outcome is achieved from which the complete system or product can be developed. This model works best in scenarios where not all of the project requirements are known in detail ahead of time. It is an iterative, trial-and-error process that takes place between the developers and the users.

Figure 2: Prototype Model

5.2 Software Development Schedule (GANTT CHART)

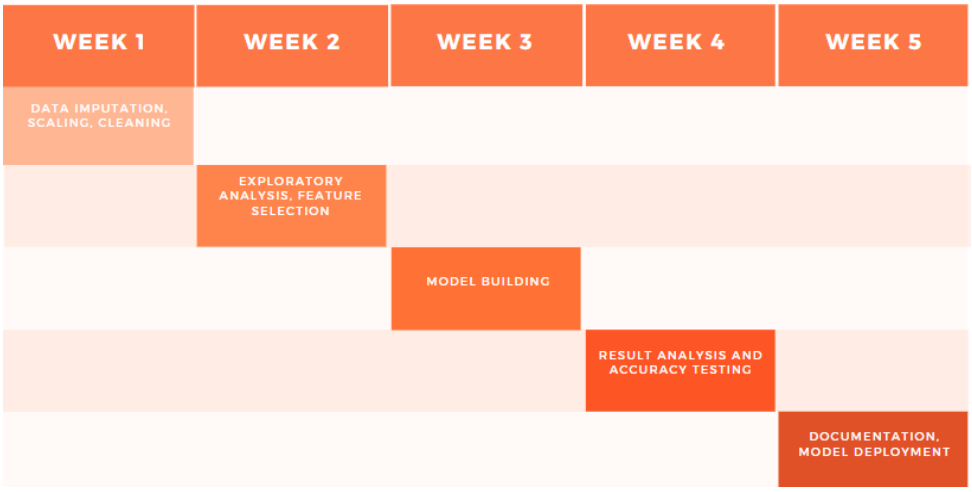


Figure 3: GANTT CHART

CHAPTER 6: EXPERIMENTS

6.1 Exploratory Data Analysis

Distribution of the target variable 'Outcome'

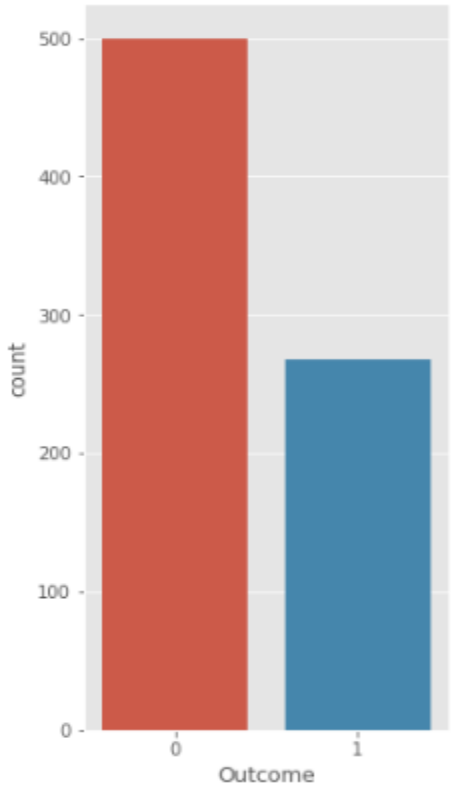


Figure 4: Histogram of the Outcome variable

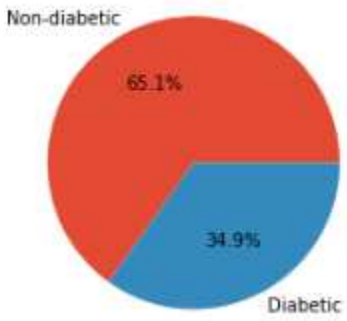


Figure 5: Pie Chart of Outcome variable

We can see that the number of diabetic samples is less than the number of non-diabetic samples.
If we plot the pie chart of the number of diabetic and non-diabetic patients, we can see that 65.1% of the samples are non-diabetic and 34.9% of the samples are diabetic.

Distribution of the feature variables

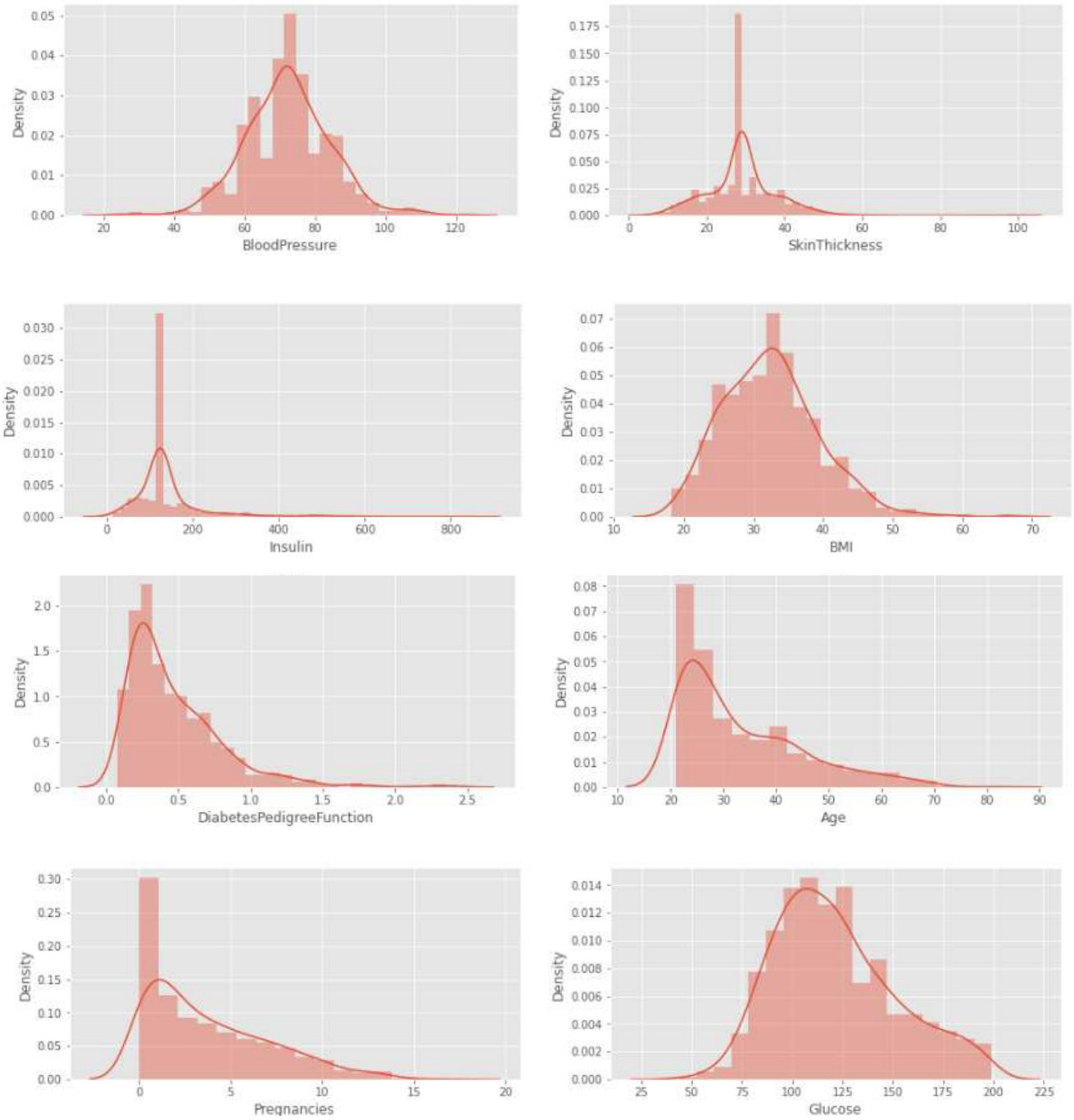


Figure 6: Distribution plot of feature variables

These plots show that the features glucose, blood pressure, BMI are approximately normally distributed and pregnancies, insulin, age, DiabetesPedigreeFunction are rightly skewed.

Correlation Matrix

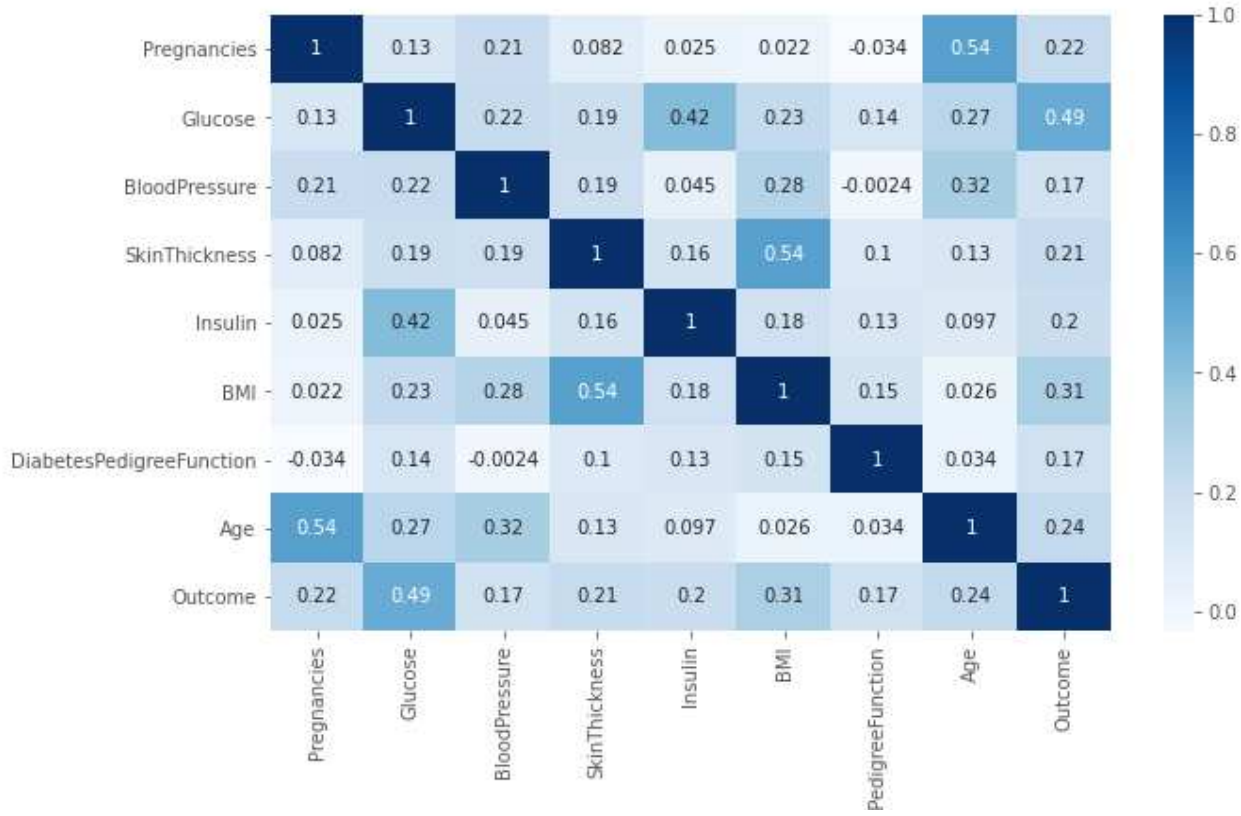


Figure 7: Correlation graph

The correlation matrix shows how each feature is correlated with other features and the target outcome variable. Glucose, Age, BMI, and Pregnancies are the most correlated features with the Outcome. Insulin and DiabetesPedigreeFunction have relatively less correlation with the outcome and Blood Pressure and SkinThickness are the least correlated with the outcome. The matrix also shows that there is a correlation between features as well.

6.2 Missing Value Imputation

The dataset had some missing values which had been encoded as zeros. Features like Glucose, Blood Pressure, SkinThickness, Insulin, and BMI had some values of zero which was not possible. So the zeros in these features were replaced by null values. These null values were then imputed using the medians of the corresponding feature's values. Median was preferred over mean as some of the features were skewed and centered more around the median.

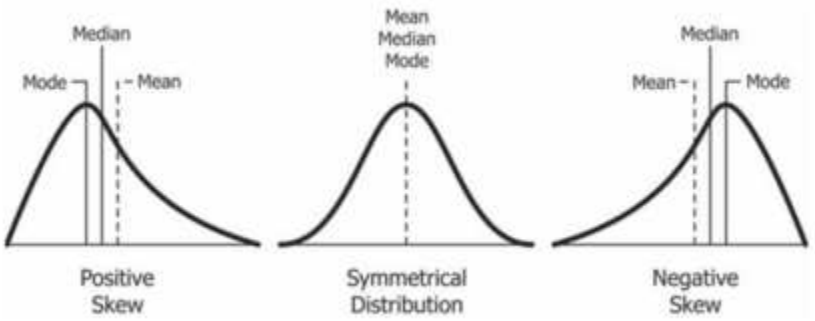


Figure 8: Mean, Median, Mode of skewed and symmetrical

6.3 Training and testing

The dataset used to build the model is usually divided in multiple data sets. In particular, two data sets are used in different stages of the creation of the model: **training (80%) and test set (20%)**.

The model is initially fit on a training data set, which is a set of examples used to fit the parameters (e.g. splits of trees of the random forest) of the model. The model is trained on the training data set using a supervised learning method.

Finally, the test data set is a data set used to provide an unbiased evaluation of a final model fit on the training data set. If the data in the test data set has never been used in training (for example in cross-validation), the test data set is also called a holdout data set. The term "validation set" is sometimes used instead of "test set" in some literature (e.g., if the original data set was partitioned into only two subsets, the test set might be referred to as the validation set).

CHAPTER 7: EVALUATION METRICS

7.1 Confusion Matrix

A confusion matrix, also known as an error matrix, is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. It allows easy identification of confusion between classes e.g. one class is commonly mislabeled as the other. The key to the confusion matrix is the number of correct and incorrect predictions are summarized with count values and broken down by each class, not just the number of errors made.

TP=54	**FP=31**
FN=15	**TN=92**

Table 2: Confusion Matrix

7.2 Accuracy

The accuracy is calculated as:

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

Where,

- True Positive (TP) =Observation is positive and is predicted to be positive.
- False Negative (FN) = Observation is positive but is predicted negative.
- True Negative (TN) = Observation is negative and is predicted to be negative.
- False Positive (FP) =Observation is negative but is predicted positive

The obtained accuracy during training the data after balancing and extracting the feature was 76%.

7.3 Recall

Recall can be defined as High Recall indicates the class is correctly recognized (a small number of FN). Recall is calculated as:

$$\text{Recall} = \frac{TP}{TP+FN}$$

The obtained recall during training the data after feature selection by balancing the data was 0.78.

7.4 Precision

To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High Precision indicates an example labeled as positive is indeed positive (a small number of FP). Precision is calculated as:

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

The obtained precision during training the data after feature selection by balancing the data was 0.64.

Evaluation Summary:

Evaluation Metrics	Values
Accuracy	76%
Recall	0.78
Precision	0.64

Table 3: Evaluation of Random Forest Mode

CHAPTER 8: DEPLOYMENT

The best-performing model was downloaded as a pickle file. A **flask** web application was built in order to get the input data from the users and provided to the model for prediction. The model took the input data and classified the user either as diabetic or non-diabetic along with the probability of the user to be diabetic. The user interface of the flask web application was created using HTML and CSS.

Figure 9: Deployment Architecture

User Interface:

Figure 10: User Interface

Prediction:

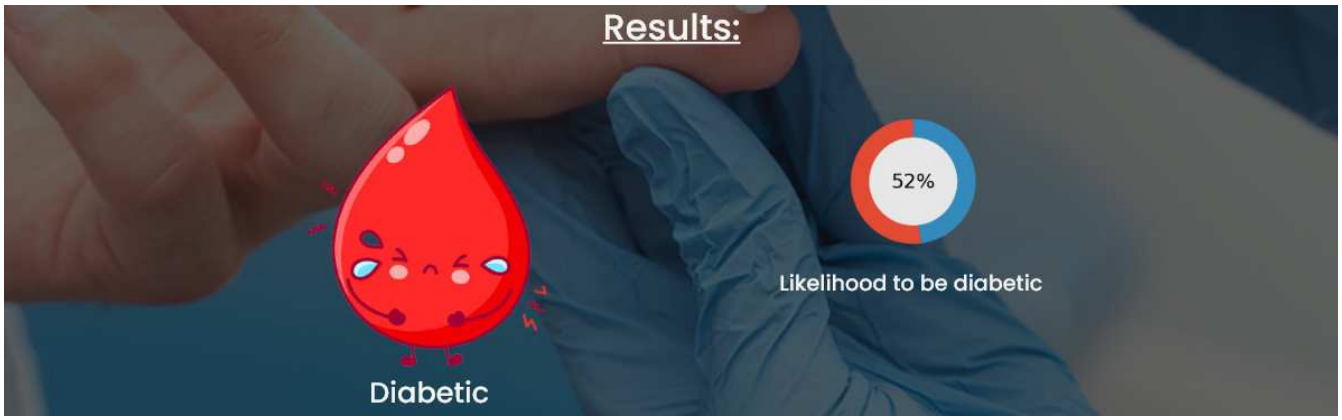
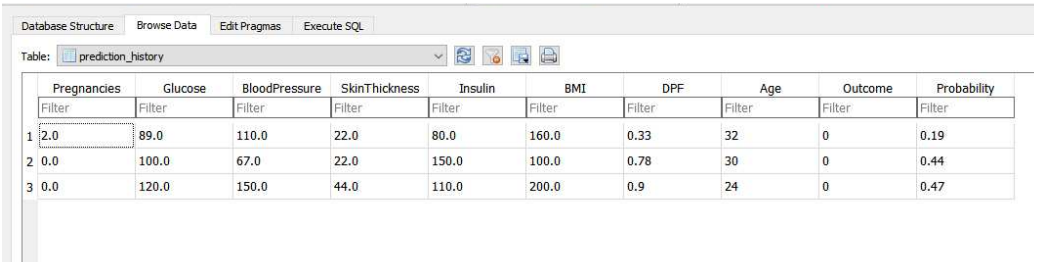


Figure 11: Prediction result

The prediction history was stored in **sqlite** database for future use and possible training.



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DPF	Age	Outcome	Probability
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	2.0	89.0	110.0	22.0	80.0	160.0	0.33	32	0	0.19
2	0.0	100.0	67.0	22.0	150.0	100.0	0.78	30	0	0.44
3	0.0	120.0	150.0	44.0	110.0	200.0	0.9	24	0	0.47

Figure 12: SQLite Database snapshot

CHAPTER 9: CODE

The coding portion was carried out to prepare the data, visualize it, pre-process it, build the model and then evaluate it. The code has been written in the Python programming language using Jupyter Notebook as IDE. The experiments and all the models building are done based on python libraries. The code is available in the Git repository given in the following link:

https://github.com/sththapa/Diabetes-Cases

Libraries used:

1. NumPy
2. Plotly
3. Matplotlib
4. Seaborn
5. Pandas
6. Sklearn
7. Flask

Modules used:	Imported class from respective modules:
a. Sklearn.model_selection	Train Test Split,KFold, Cross Val Score
b. Sklearn.metrics	Classification Report,Confusion Matrix
c. Sklearn.pipeline	Pipeline
d. Imblearn.over_sampling	SMOTE
e. Sklearn.preprocessing	Robust Scalar
f. Sklearn.model_selection	Train_test_split, StratifiedKFold
g.Sklearn.ensemble	Random Forest Classifier

Table 4: Major modules and classes used

CHAPTER 10: CONCLUSION

The early prognosis of diabetes cases can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine. This project resolved the feature selection with balanced data behind the models and successfully predicted the diabetes cases with around 80% accuracy. The model was used to find the best machine learning algorithm but among all Random Forest model gave the best result among all the models and hence Random Forest model has been selected as the best model for this project.