Diabetes Prediction

ASV Siva Sai (RA2011033010037)

- Data collection, User interface design

Adidela Isaac Arun (RA2011033010043)

- Research and Development

Sarang Parameswaran (RA2011033010050)

- Documentation and Implementation

Problem statement:

Diabetes is a chronic disease with the potential to cause a worldwide health care crisis. According to International Diabetes Federation 382 million people are living with diabetes across the whole world. By 2035, this will be doubled as 592 million. Diabetes mellitus or imply diabetes is a disease caused due to the increase level of blood glucose. Various traditional methods, based on physical and chemical tests, are available for diagnosing diabetes. Data science methods have the potential to benefit other scientific fields by shedding new light on common questions. One such task is to help make predictions on medical data.

The problem statement for a project on diabetes prediction is to develop a model that can accurately predict the likelihood of a patient developing diabetes based on their medical history, lifestyle factors, and other relevant data. The aim of this project is to create a tool that can help healthcare professionals identify individuals at risk of developing diabetes early on, so that they can take preventive measures and provide timely treatment to manage the condition. The project will involve collecting and analyzing large amounts of data, using machine learning algorithms to develop the prediction model, and testing the model's accuracy and effectiveness in real-world scenarios.

Literature survey:

S.No	Paper Name	Journal Name	Year
1	"A comparative study of machine learning algorithms for diabetes prediction" by Hadi Zare	Journal of Medical Systems	May 2020
2	"Prediction of diabetes mellitus using machine learning algorithms" by Saurabh Pal	Journal of Medical Systems	June 2019

Literature survey:

S.No	Paper Name	Journal Name	Year
3	"Predicting diabetes with machine learning techniques: A review" by Asmaa Abbas	Journal of Diabetes Research	March 2020
4	"Predicting Type 2 Diabetes Mellitus using Machine Learning Techniques" by R. Kavitha and S. Kulothungan	International journal of Engineering and Technology	2018

Literature survey:

S.No	Paper Name	Journal Name	Year
5	"A Machine Learning-based Predictive Model for Type 2 Diabetes Mellitus" by H. V. Shinde and S. R. Kulkarni	International Journal of Advanced Research in Computer Science	2021
6	"Early Diabetes Prediction using Machine Learning Algorithms" by K. Gunasekaran and S. Muthukumar	International Journal of Advanced Science and Technology	2019

Limitations:

- Data availability: The accuracy of the prediction model is highly dependent on the quantity and quality of data used to train the algorithm. If the dataset is too small or does not represent the population, the prediction model may not generalize well to new cases.
- Bias in data: The data used for training the algorithm may contain inherent bias due to various factors such as ethnicity, age, and gender, which can lead to inaccurate predictions for certain population subgroups.
- Complexity of disease: Diabetes is a complex disease that can have various causes and risk factors, including genetic, lifestyle, and environmental factors. Developing a prediction model that takes into account all these factors accurately is challenging.

- Ethical issues: The use of personal health data in a prediction model raises concerns about privacy and data security. The data used for training the algorithm must be anonymized and protected to prevent any unauthorized access.
- False positives/negatives: Even with a highly accurate prediction model, false positives and false negatives can occur. False positives can lead to unnecessary treatment, while false negatives can delay the diagnosis and treatment of the disease.
- Interpretability: The inner workings of some machine learning models are not always easily interpretable by humans, which can make it difficult to understand how the model arrived at a particular prediction.

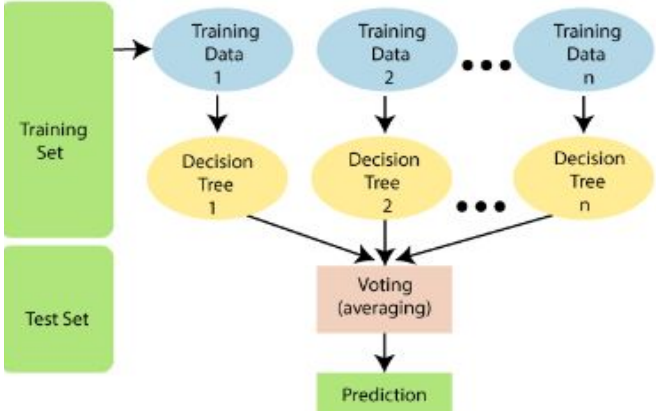
Objectives: These are mainly to resolve the limitations

- Collecting and analyzing data: The project involves collecting and analyzing large amounts of data related to the patient's medical history, lifestyle, and other factors that may influence their likelihood of developing diabetes.[6]
- Data diversity: Since diabetes is a complex disease and various factors affect the outcome, the data taken has multiple factors that change the result.[2]
- Multiple data sources: Diabetes is a personal issue, therefore data is not available to the public, to resolve this issue the data collected has been taken from multiple sources.[3]
- Mixed Data: Data from multiple ethnic and racial backgrounds has been taken to better predict the result.[2]
- Algorithms used: Since machine learning involves many different algorithms, we have done a comparative study involving random forest and logistic regression.[1]

Algorithm: Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique

- Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset
- The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting
- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.



Algorithm: SVM Classifier

Support Vector Machine (SVM) is a popular machine learning algorithm for classification and regression tasks. Here are some key points about SVM:

1. SVM is a supervised learning algorithm that can be used for both classification and regression tasks.

2. The main idea behind SVM is to find the hyperplane that maximizes the margin between two classes in the feature space.

3. SVM is a binary classifier, which means it can only classify data into two classes.

4. SVM is a powerful algorithm that can handle both linear and non-linear data.

5. SVM is widely used in many areas, including finance, biology, and computer vision.

$$\mathrm{Accuracy} = \frac{(\mathrm{TP}+\mathrm{TN})}{(\mathrm{TP}+\mathrm{TN}+\mathrm{FN}+\mathrm{FP})}$$

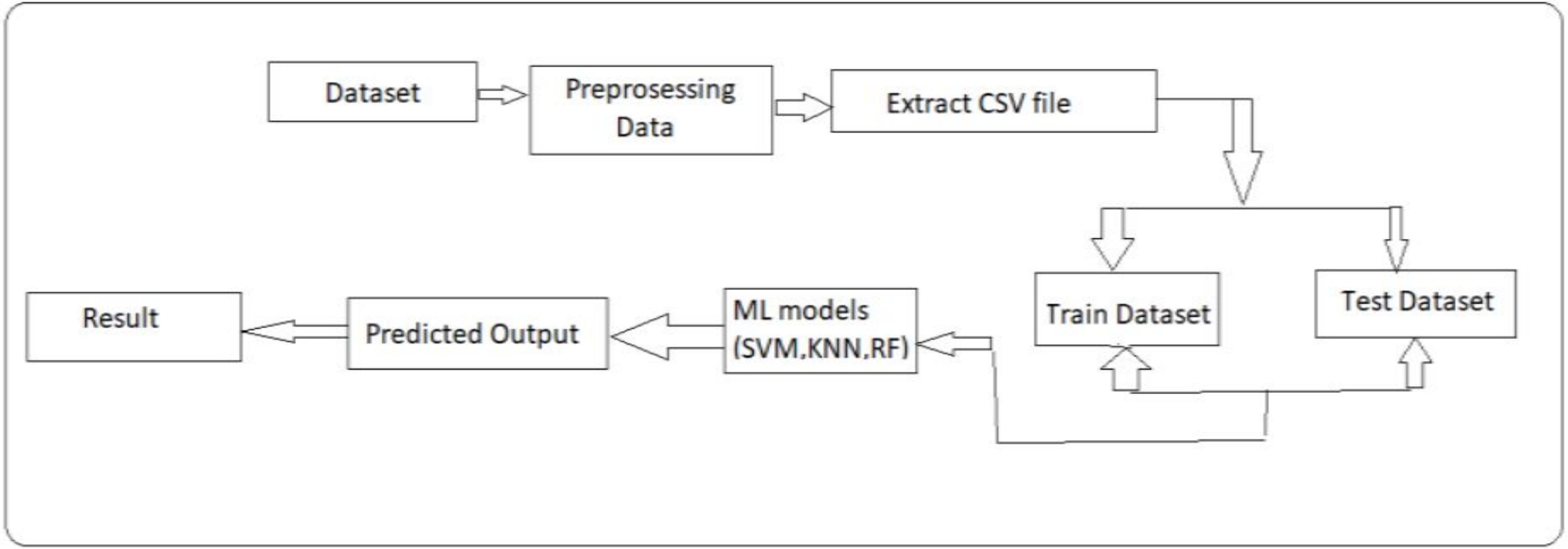
$$\mathrm{Recall} = \frac{(\mathrm{TP})}{(\mathrm{TP}+\mathrm{FN})}$$

$$\mathrm{Precision} = \frac{(\mathrm{TP})}{(\mathrm{TP}+\mathrm{FP})}$$

$$\mathrm{F} - \mathrm{measure} = \frac{2\times(\mathrm{Precision}\times\mathrm{Recall})}{(\mathrm{Precision}+\mathrm{Recall})}$$

Classification	Precision	Recall	F-measure	Accuracy
RF (K-fold)	0.744	0.750	0.746	74.69%
RF (Splitting)	0.779	0.771	0.774	77.14%
SVM (K-fold)	0.761	0.768	0.759	76.82%
SVM (Splitting)	0.774	0.777	0.775	77.71%

Architecture diagram:



Modules:

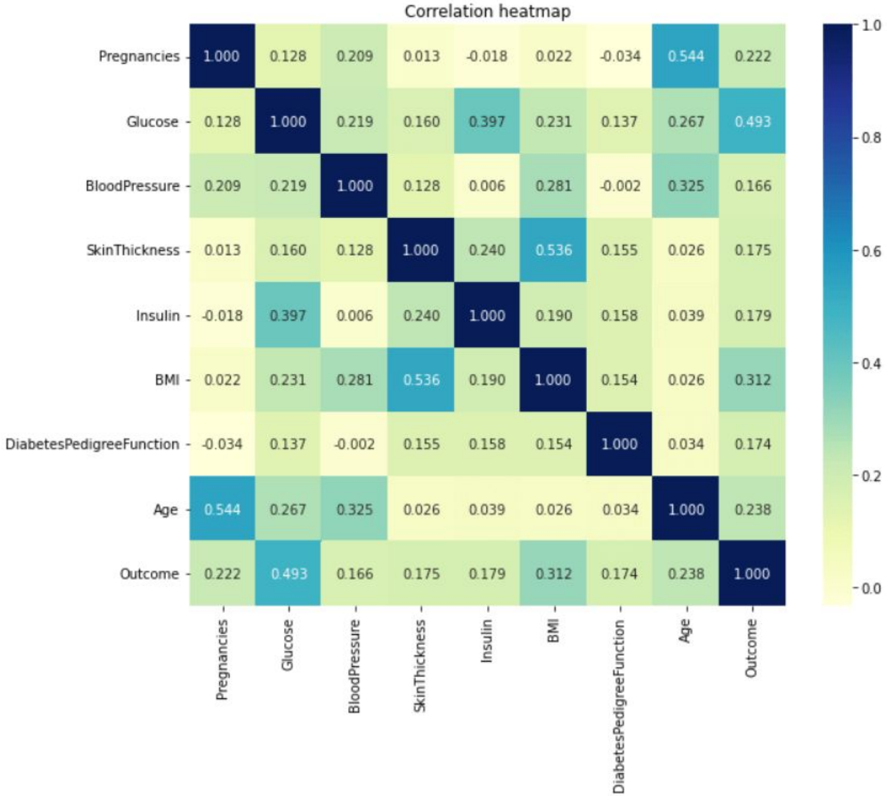
This model has five modules:

1. Collection of Data.

2. Preprocessing over Data.

3. Model Building.

4. Evaluation.

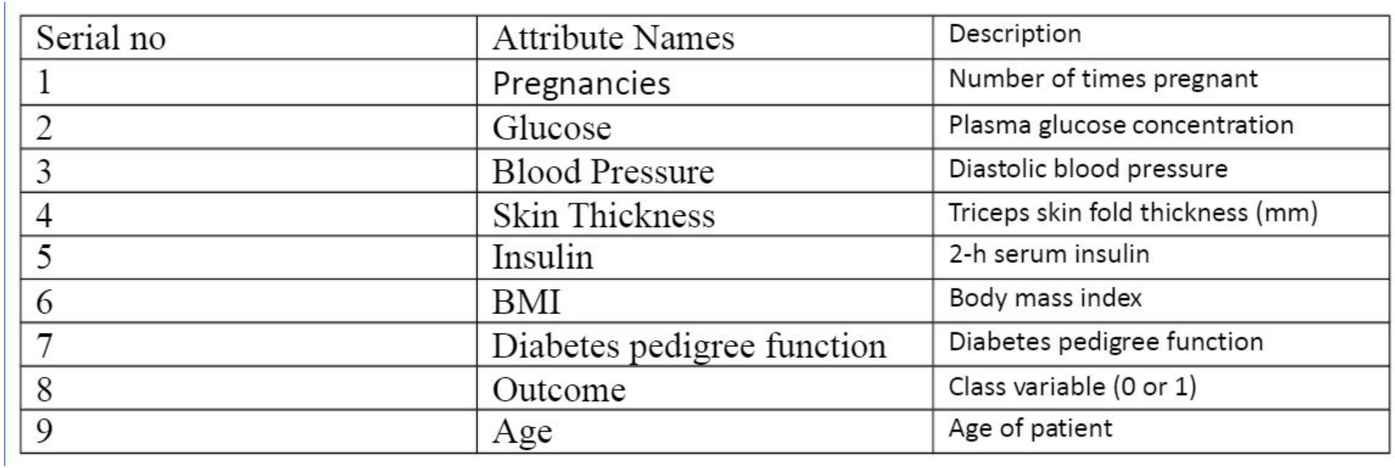


Correlation heatmap

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000	0.128	0.209	0.013	-0.018	0.022	-0.034	0.544	0.222
Glucose	0.128	1.000	0.219	0.160	0.397	0.231	0.137	0.267	0.493
BloodPressure	0.209	0.219	1.000	0.128	0.006	0.281	-0.002	0.325	0.166
SkinThickness	0.013	0.160	0.128	1.000	0.240	0.536	0.155	0.026	0.175
Insulin	-0.018	0.397	0.006	0.240	1.000	0.190	0.158	0.039	0.179
BMI	0.022	0.231	0.281	0.536	0.190	1.000	0.154	0.026	0.312
DiabetesPedigreeFunction	-0.034	0.137	-0.002	0.155	0.158	0.154	1.000	0.034	0.174
Age	0.544	0.267	0.325	0.026	0.039	0.026	0.034	1.000	0.238
Outcome	0.222	0.493	0.166	0.175	0.179	0.312	0.174	0.238	1.000

1. Collection of Data.

This module includes collection of data and understanding the data to study trends which helps in prediction and access the results, Dataset description is given below.



Serial no	Attribute Names	Description
1	Pregnancies	Number of times pregnant
2	Glucose	Plasma glucose concentration
3	Blood Pressure	Diastolic blood pressure
4	Skin Thickness	Triceps skin fold thickness (mm)
5	Insulin	2-h serum insulin
6	BMI	Body mass index
7	Diabetes pedigree function	Diabetes pedigree function
8	Outcome	Class variable (0 or 1)
9	Age	Age of patient

2. Preprocessing over Data.

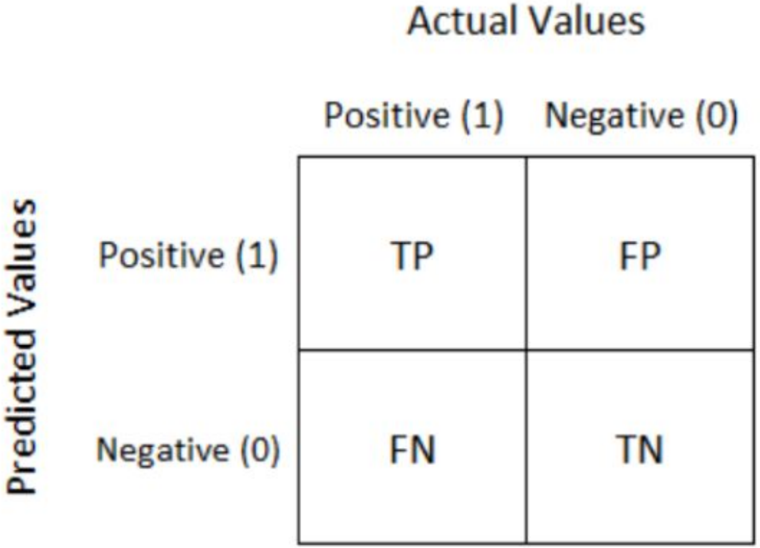
This module consists of process of preparing the raw data and making it suitable for a machine learning model. (i.e. cleaning and organizing) to make it suitable for a building and training ML model.

3. Model Building.

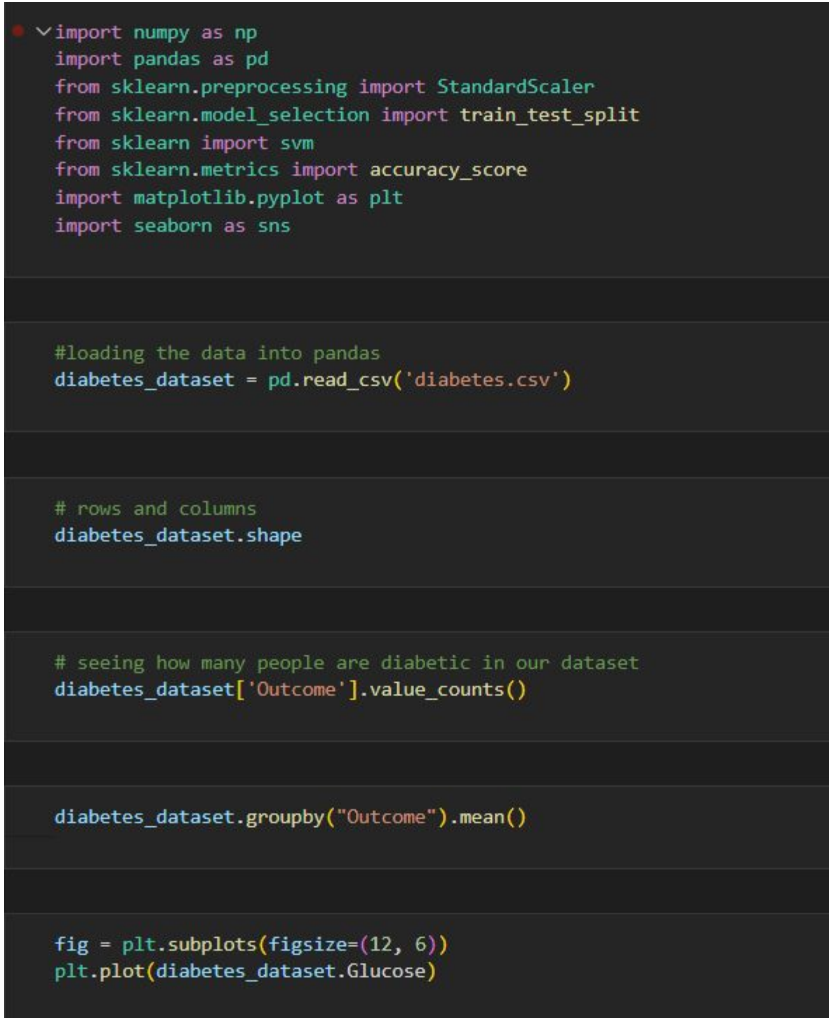
This phase includes building model for prediction of diabetes we have implementing various machine learning algorithms. This algorithm includes support vector classifier, Random Forest, K-nearest neighbor

4. Evaluation.

This final step of prediction model here we evaluate the prediction results using various evaluations metrics like classification, accuracy, confusion matrix.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN



```python
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns
```

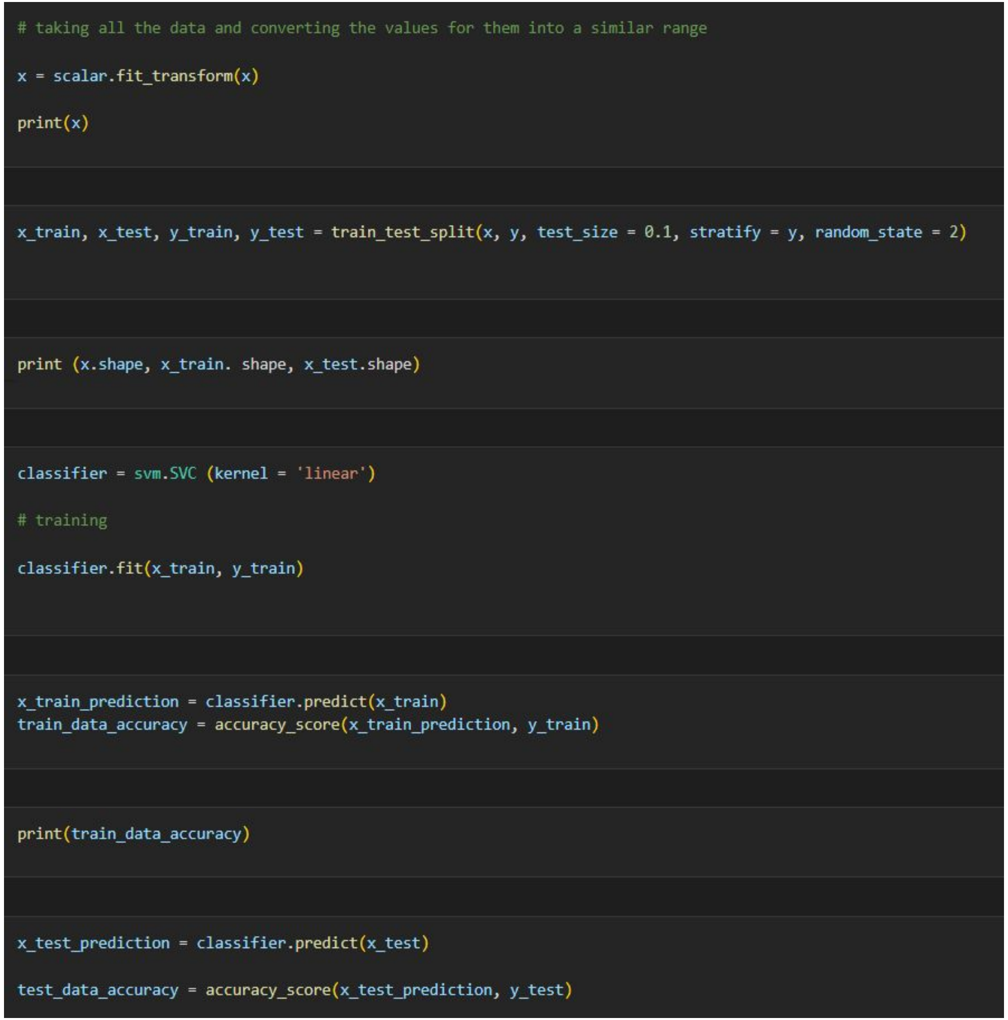
```python
#loading the data into pandas
diabetes_dataset = pd.read_csv('diabetes.csv')
```

```python
# rows and columns
diabetes_dataset.shape
```

```python
# seeing how many people are diabetic in our dataset
diabetes_dataset['Outcome'].value_counts()
```

```python
diabetes_dataset.groupby("Outcome").mean()
```

```python
fig = plt.subplots(figsize=(12, 6))
plt.plot(diabetes_dataset.Glucose)
```



```python
# taking all the data and converting the values for them into a similar range

x = scalar.fit_transform(x)

print(x)
```

```python
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.1, stratify = y, random_state = 2)
```

```python
print (x.shape, x_train. shape, x_test.shape)
```

```python
classifier = svm.SVC (kernel = 'linear')

# training

classifier.fit(x_train, y_train)
```

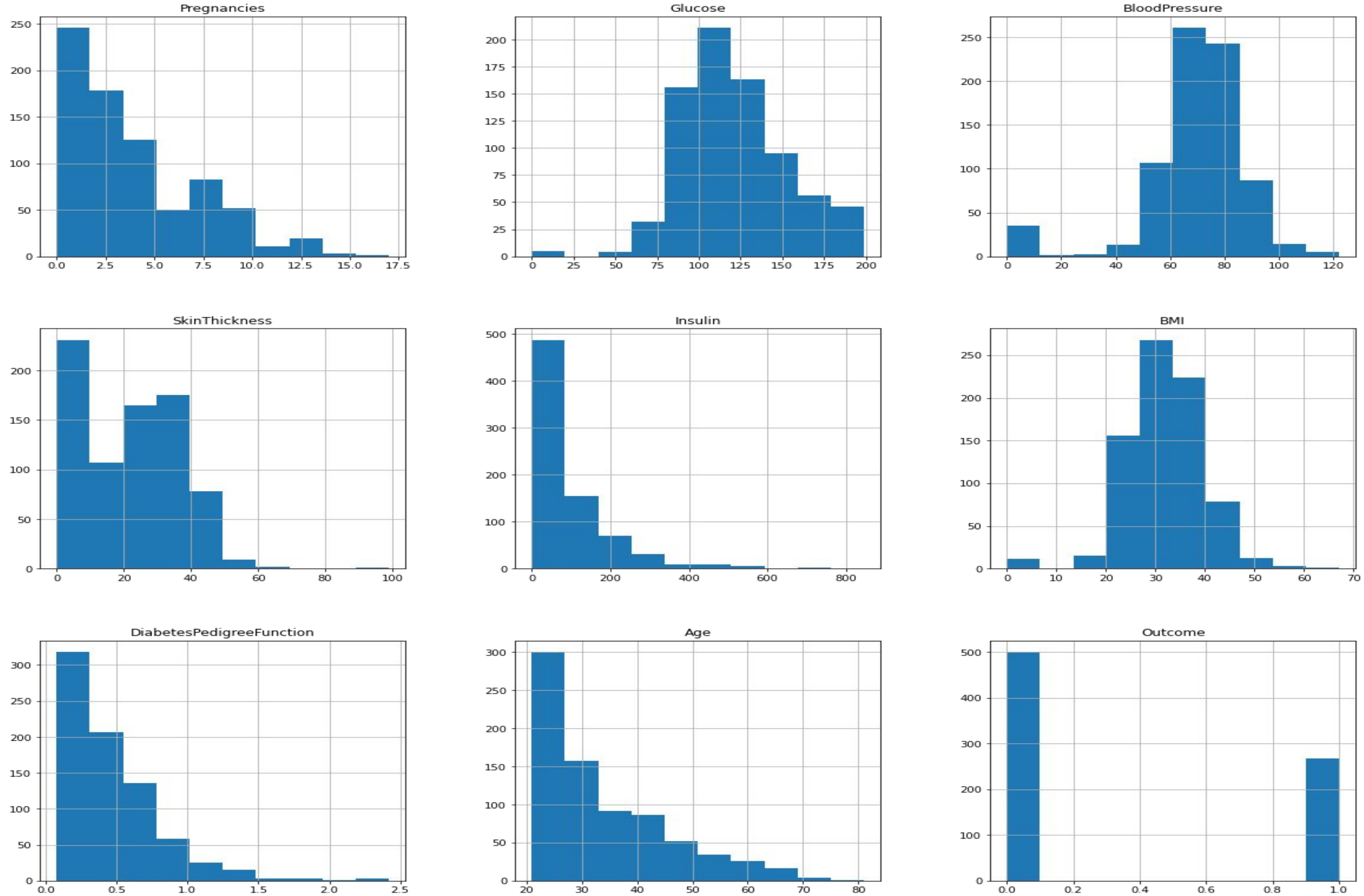
```python
x_train_prediction = classifier.predict(x_train)
train_data_accuracy = accuracy_score(x_train_prediction, y_train)
```

```python
print(train_data_accuracy)
```

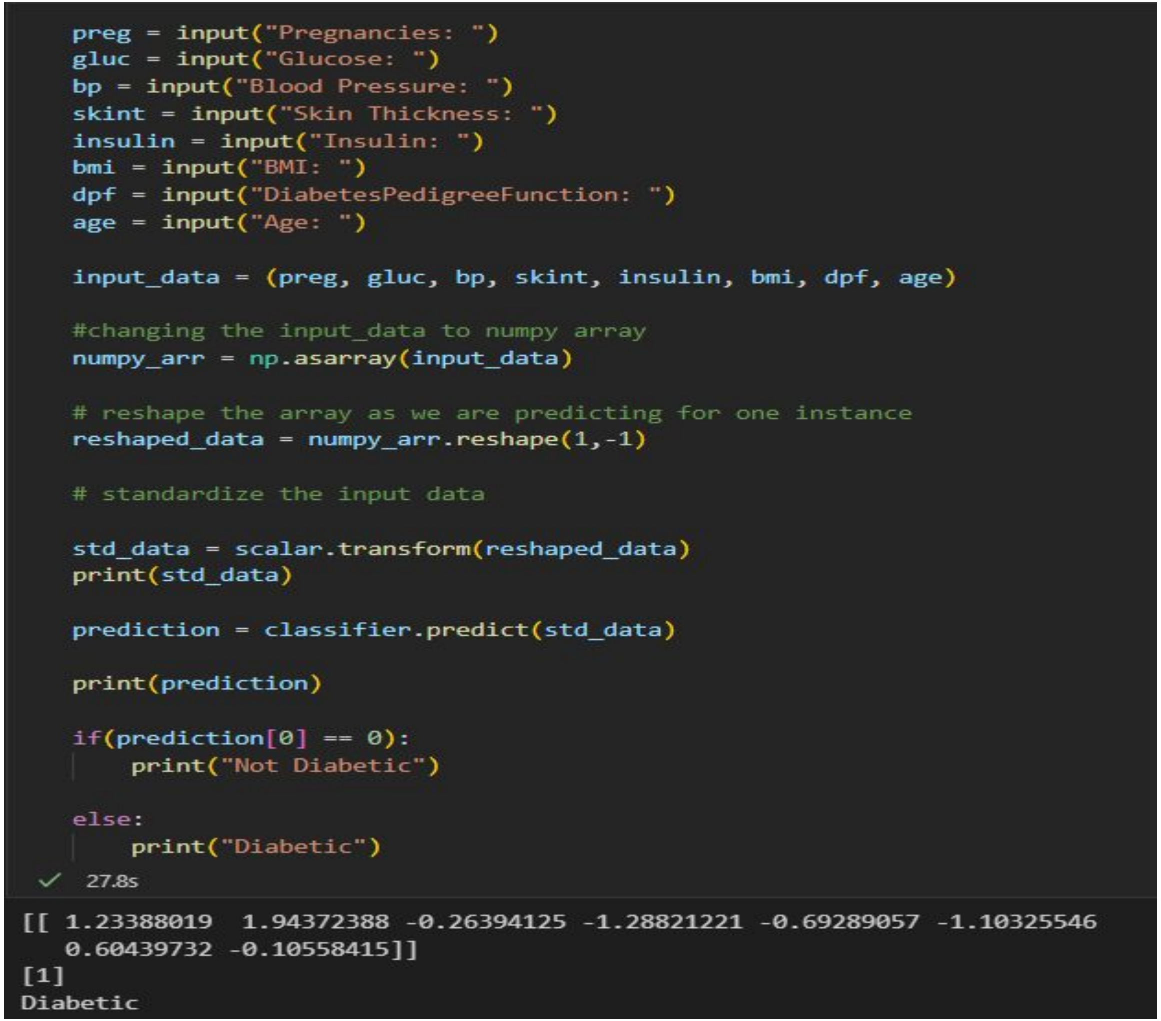
```python
x_test_prediction = classifier.predict(x_test)

test_data_accuracy = accuracy_score(x_test_prediction, y_test)
```

Graph:



Pregnancies
0 50 100 150 200 250
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
Glucose
0 25 50 75 100 125 150 175 200
0 25 50 75 100 125 150 175 200
BloodPressure
0 50 100 150 200 250
0 20 40 60 80 100 120
SkinThickness
0 50 100 150 200
0 20 40 60 80 100
Insulin
0 100 200 300 400 500
0 200 400 600 800
BMI
0 50 100 150 200 250
0 10 20 30 40 50 60 70
DiabetesPedigreeFunction
0 50 100 150 200 250 300
0.0 0.5 1.0 1.5 2.0 2.5
Age
0 50 100 150 200 250 300
20 30 40 50 60 70 80
Outcome
0 100 200 300 400 500
0.0 0.2 0.4 0.6 0.8 1.0



```python
preg = input("Pregnancies: ")
gluc = input("Glucose: ")
bp = input("Blood Pressure: ")
skint = input("Skin Thickness: ")
insulin = input("Insulin: ")
bmi = input("BMI: ")
dpf = input("DiabetesPedigreeFunction: ")
age = input("Age: ")

input_data = (preg, gluc, bp, skint, insulin, bmi, dpf, age)

#changing the input_data to numpy array
numpy_arr = np.asarray(input_data)

# reshape the array as we are predicting for one instance
reshaped_data = numpy_arr.reshape(1,-1)

# standardize the input data

std_data = scalar.transform(reshaped_data)
print(std_data)

prediction = classifier.predict(std_data)

print(prediction)

if(prediction[0] == 0):
    print("Not Diabetic")

else:
    print("Diabetic")
```

✓ 27.8s

```
[[ 1.23388019  1.94372388 -0.26394125 -1.28821221 -0.69289057 -1.10325546
   0.60439732 -0.10558415]]
[1]
Diabetic
```

THANK YOU