DIABETES PREDICTION USING MACHINE LEARNING

A

Project Based Learning Report

Submitted

by

Mr. Shreyash Somvanshi	**2127062**
Mr. Sujay Shinde	**2127063**
Mr. Prajwal Rudrapwar	**2127076**
Ms. Vedanti Mane	**2127073**
Ms. Trupti Kharat	**2127032**

In partial fulfillment for the requirement of Project Based Learning-II

of

Bachelor of Artificial Intelligence and Data Science

Under the guidance of

Prof. Rajkumar Panchal

(Assistant Professor)

Department of Artificial Intelligence and Data Science

Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering and Technology

Bhigwan Road, Vidyanagari

Baramati-413133

2021-2022

Vidya Pratishthan's

Kamalnayan Bajaj Institute of Engineering and Technology, Baramati

Department of Artificial Intelligence and Data Science

Certificate

This is to certify that following students

Mr. Shreyash Somvanshi	**2127062**
Mr. Sujay Shinde	**2127063**
Mr. Prajwal Rudrapwar	**2127076**
Ms. Vedanti Mane	**2127073**
Ms. Trupti Kharat	**2127032**

have successfully completed their project work on

DIABETES PREDICTION USING MACHINE LEARNING

during the academic year **2021-2022** in the partial fulfillment towards the completion of **Project Based Learning-II** in **Artificial Intelligence and Data Science**

Project Guide
(Rajkumar Panchal)

Head, Deptt.of AI & DS
(Digambar Padulkar)

Principal
(**Dr. R. S. Bichkar**)

Internal Examiner **External Examiner**

Acknowledgments

We express our deep sense of gratitude to all those who have been instrumental in preparation of this project.We are thankful to all the faculties of Deptt.of AI and DS for their constant support, guidance and encouragement. We acknowledge the kind of support, efforts and timely guidance provided by Prof.Rajkumar Panchal sir.This project report helps in better understanding of the subject matter. We also like to express regards to the books and internet and linkedin conections who provided us with subject related insights.

Mr. Shreyash Somvanshi
Mr. Sujay Shinde
Mr. Prajwal Rudrapwar
Ms. Vedanti Mane
Ms. Trupti Kharat

ABSTRACT

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 is a disease caused due to the increase level of blood glucose. This high blood glucose produces the symptoms of frequent urination, increased thirst, and increased hunger. Diabetes is a one of the leading cause of blindness, kidney failure, amputations, heart failure and stroke. When we eat, our body turns food into sugars, or glucose. At that point, our pancreas is supposed to release insulin. Insulin serves as a key to open our cells, to allow the glucose to enter and allow us to use the glucose for energy. But with diabetes, this system does not work. Type 1 and type 2 diabetes are the most common forms of the disease, but there are also other kinds, such as gestational diabetes, which occurs during pregnancy, as well as other forms. Machine learning is an emerging scientific field in data science dealing with the ways in which machines learn from experience. The aim of this project is to develop a system which can perform early prediction of diabetes for a patient with a higher accuracy by combining the results of different machine learning techniques. The algorithms like K nearest neighbour, Logistic Regression, Random forest, Support vector machine and Decision tree are used. The accuracy of the model using each of the algorithms is calculated. Then the one with a good accuracy is taken as the model for predicting the diabetes.

Contents

1

Synopsis

1.1 Title

DIABETES PREDICTION USING MACHINE LEARNING

1.2 Technical Keywords

Machine Learning, Diabetes, Decision tree, Train-Test-Split, K nearest neighbour, Logistic Regression, Support vector Machine, Accuracy.

2

Problem Statement

Make use of new emerging technologies in the healthcare to reduce time and efforts.

2.1 Goals and Objectives

To use modern technologies to increase accuracy and automation in healthcare

2.2 Statement of Scope

This Diabetes Prediction using Machine Learning works for Healthcare domain.It has 80% accuracy in predicting diabetes based on the factors like patient's insulin, glucose level, age,etc. factors. This Diabetes Prediction Model makes use of various classification Algorithms to categorize patients into diabetic and non-diabetic.

3

Area Project

This Diabetes Prediction using Machine Learning works for Healthcare domain.It has 80% accuracy in predicting diabetes based on the factors like patient's insulin, glucose level, age,etc. factors.

4

Methodology of Problem Solving

1. Import the dataset with various patient records on parameters like age, insulin, glucose, blood pressure.
2. Clean the dataset i.e. remove the unwanted constraints (Preprocessing)
3. Perform train-test-split on processed dataset
4. Use the preffered algorithm
5. Compare and check accuracy

5

Introduction

5.1 Motivation of the Project

Motive of this project is to use the Artificial Intelligence technologies like Machine Learning in healthcare sector to increase efficiency and accuracy of results

5.2 Literature Survey

This Diabetes Prediction Model makes use of various classification Algorithms to categorize patients into diabetic and non-diabetic.
Currently we have achieved 80% accuracy but it can be made more precise with help of appropriate data processing.

6

Dataset Description

1. 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.

2. Dataset used for this model is located on "https://raw.githubusercontent.com/Shreya

3. It consists the detailed records of patients.The attributes in datasets are Age, Insulin, Pregnancies, Glucose, DiabetesPedigreeFunction, BMI, Blood pressure,etc

4. Dataset consists of 768 rows and 9 columns

7

Libraries and Functions used

Libraries:

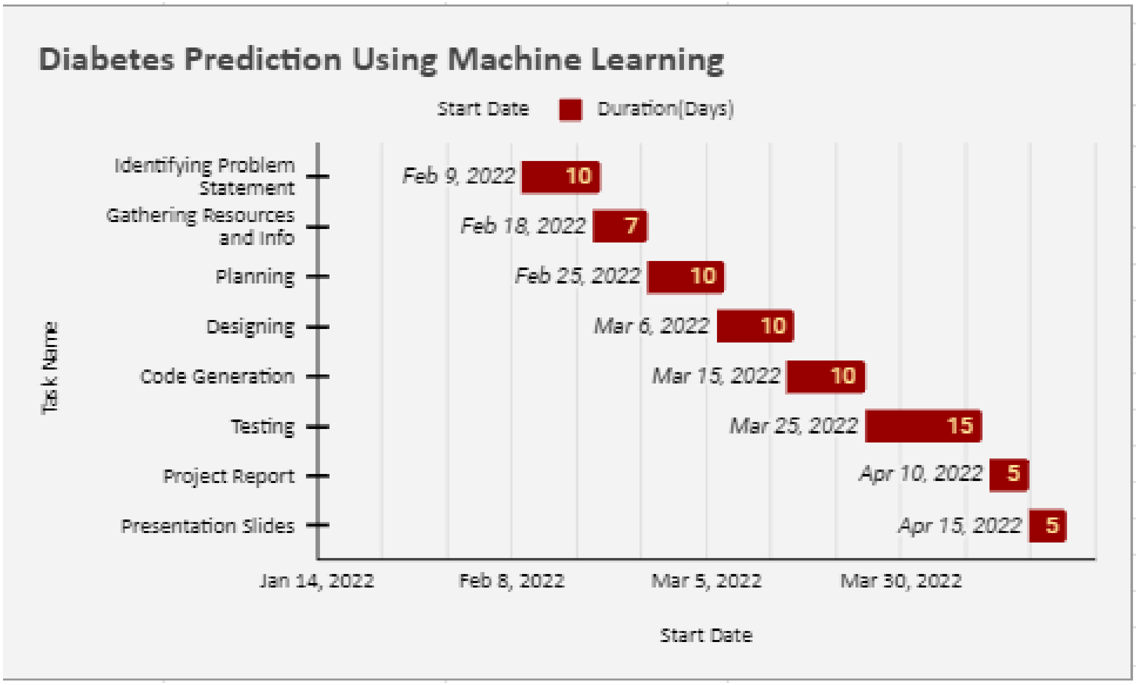
1. pandas
2. numpy
3. sklearn

Functions/Modules used from sklearn:

1. sklearn.model_selection import train_test_split
2. sklearn.metrics import mean_absolute_error
3. sklearn.linear_model import LogisticRegression
4. sklearn.linear_model import LinearRegression
5. sklearn.metrics import accuracy_score
6. sklearn.neighbors import KNeighborsClassifier
7. sklearn.naive_bayes import GaussianNB
8. sklearn.svm import SVC
9. sklearn.tree import DecisionTreeClassifier

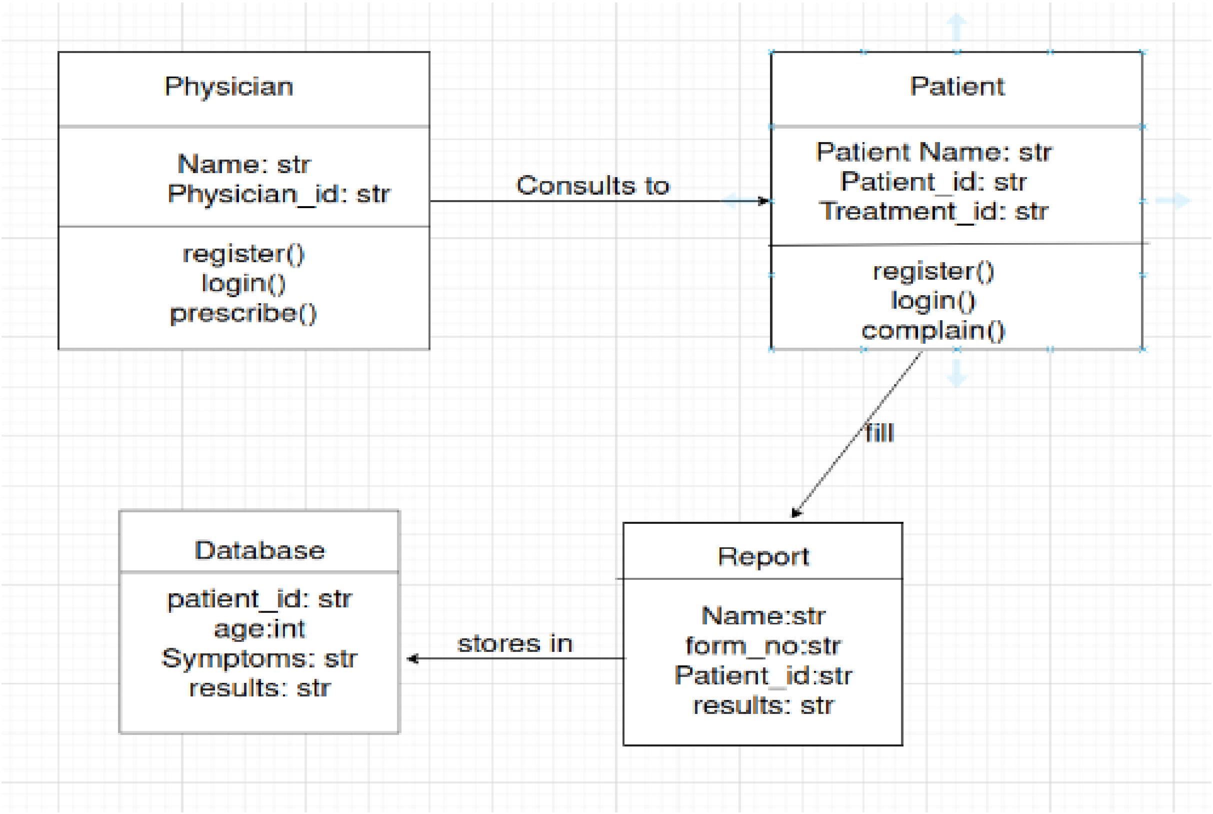
8

Project Planning



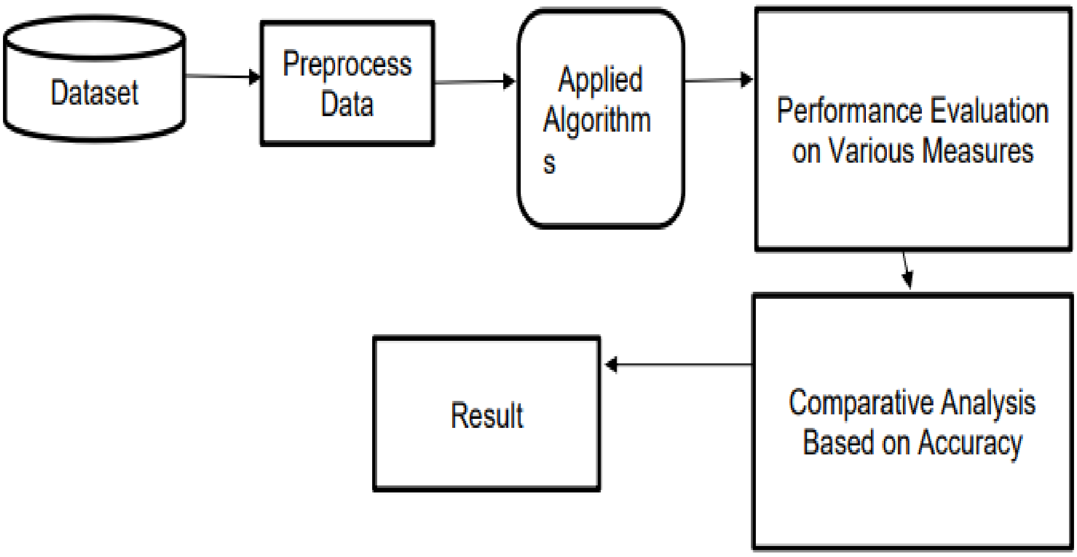
9

Class Diagram



10

Architecture


Dataset
Preprocess Data
Applied Algorithm s
Performance Evaluation on Various Measures
Comparative Analysis Based on Accuracy
Result

Proposed Model Diagram

11

Code

```
# -*- coding: utf-8 -*-
"""PBL2DiabetesPredictionUsingML.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1IpgIYr-Vhc95CbLqk0EK78A8P8njEYoE
"""

import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
import warnings
warnings.filterwarnings('ignore')

"""https://raw.githubusercontent.com/ShreyashSomvanshi/Datasets/main/Diabetes.csv

"""

dataset =
    pd.read_csv(r'https://raw.githubusercontent.com/ShreyashSomvanshi/Datasets/main/Diabetes.csv')

"""https://github.com/Yantra-Byte/Dataset/raw/main/Diabetes.csv"""

dataset.head()

dataset.tail()
dataset.shape

dataset.info()

dataset.isnull().sum()

dataset.describe()

"""Visualizing Data"""

# Now we will be imputing the mean value of the column to each missing value of that
    particular column
dataset['Glucose'].fillna(dataset['Glucose'].mean(), inplace = True)
dataset['BloodPressure'].fillna(dataset['BloodPressure'].mean(), inplace = True)
dataset['SkinThickness'].fillna(dataset['SkinThickness'].median(), inplace = True)
dataset['Insulin'].fillna(dataset['Insulin'].median(), inplace = True)
dataset['BMI'].fillna(dataset['BMI'].median(), inplace = True)

# Now, let's check that how well our outcome column is balanced
color_wheel = {1: "#0392cf", 2: "#7bc043"}
colors = dataset["Outcome"].map(lambda x: color_wheel.get(x + 1))
print(dataset.Outcome.value_counts())
p=dataset.Outcome.value_counts().plot(kind="pie")

dataset['Outcome'].value_counts()

# Plotting the distributions after removing the NAN values
p = dataset.hist(figsize = (20,20))

"""Correlation Matrix

"""

# Correlation between all the features before cleaning
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12,10))
p = sns.heatmap(dataset.corr(), annot=True,cmap ='Reds') # seaborn has an easy method
    to showcase heatmap

X = dataset.drop(columns='Outcome',axis=1)

y = dataset['Outcome']

X
y
```

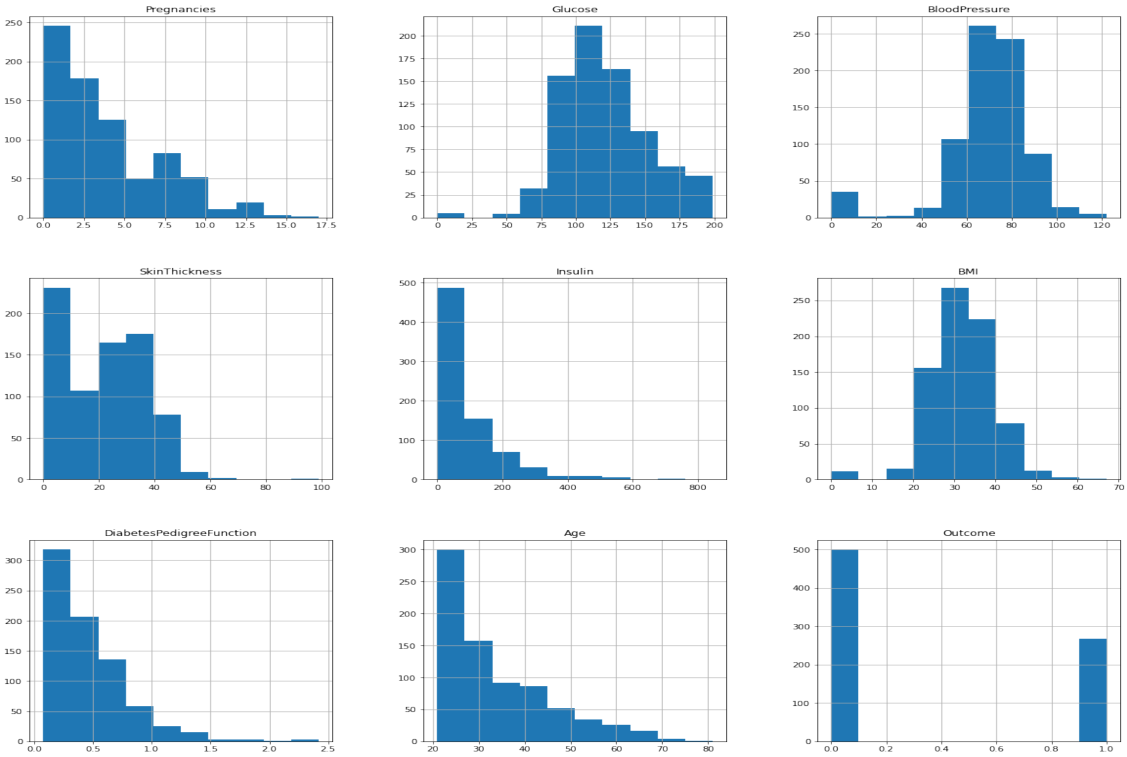
```
"""#Declaring Training and Testing Data"""

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state
    = 0)

X_train.shape, X_test.shape, y_train.shape, y_test.shape

#from sklearn.linear_model import LogisticRegression
#from sklearn.neighbors import KNeighborsClassifier
#from sklearn.naive_bayes import GaussianNB
#from sklearn.svm import SVC
#from sklearn.tree import DecisionTreeClassifier
#from sklearn.ensemble import RandomForestClassifier

#model = LogisticRegression()
#model = KNeighborsClassifier()
#model = GaussianNB()
#model = SVC()
#model = DecisionTreeClassifier()
#model = RandomForestClassifier()

"""#Logistic Regression"""

modl = LogisticRegression()
modelLR = modl.fit(X_train, y_train)
X_train_pred = modelLR.predict(X_train)
trainingData_accuracy = accuracy_score(X_train_pred, y_train)
print("Accuracy on Training Data : ", trainingData_accuracy)
X_test_pred = modelLR.predict(X_test)
testData_accuracy = accuracy_score(X_test_pred, y_test)
print("Accuracy on Test Data : ", testData_accuracy)

# from sklearn.metrics import classification_report, confusion_matrix

# print(confusion_matrix(y_test, X_test_pred))
# print(classification_report(y_test, X_test_pred))

"""#K Neighbors Classifier"""

modl = KNeighborsClassifier()
modelKNN = modl.fit(X_train, y_train)
X_train_pred = modelKNN.predict(X_train)
trainingData_accuracy = accuracy_score(X_train_pred, y_train)
print("Accuracy on Training Data : ", trainingData_accuracy)
X_test_pred = modelKNN.predict(X_test)
testData_accuracy = accuracy_score(X_test_pred, y_test)
print("Accuracy on Test Data : ", testData_accuracy)
```

```
"""#Support Vector Machine"""

model = SVC()
modelSVM = modl.fit(X_train, y_train)
X_train_pred = modelSVM.predict(X_train)
trainingData_accuracy = accuracy_score(X_train_pred, y_train)
print("Accuracy on Training Data : ", trainingData_accuracy)
X_test_pred = modelSVM.predict(X_test)
testData_accuracy = accuracy_score(X_test_pred, y_test)
print("Accuracy on Test Data : ", testData_accuracy)

"""#Decision Tree"""

modl = DecisionTreeClassifier()
modelDT = modl.fit(X_train, y_train)
X_train_pred = modelDT.predict(X_train)
trainingData_accuracy = accuracy_score(X_train_pred, y_train)
print("Accuracy on Training Data : ", trainingData_accuracy)
X_test_pred = modelDT.predict(X_test)
testData_accuracy = accuracy_score(X_test_pred, y_test)
print("Accuracy on Test Data : ", testData_accuracy)

"""#Naive Bayes"""

modl = GaussianNB()
modelNB = modl.fit(X_train, y_train)
X_train_pred = modelNB.predict(X_train)
trainingData_accuracy = accuracy_score(X_train_pred, y_train)
print("Accuracy on Training Data : ", trainingData_accuracy)
X_test_pred = modelNB.predict(X_test)
testData_accuracy = accuracy_score(X_test_pred, y_test)
print("Accuracy on Test Data : ", testData_accuracy)

"""#Conclusion from Training and Testing Accuracies of different algorithms used:

% <table>
%   <tr>
%     <th> Algorithm </th>
%     <th> Testing Accuracy </th>
%     <th> Training Accuracy </th>
%   </tr>
%   <tr>
%     <td>Logistic Regression</td>
%     <td>0.7792</td>
%     <td>0.7636</td>
%   </tr>
%   <tr>
%     <td>Naive Bayes</td>
%     <td>0.7619</td>
%     <td>0.7673</td>
%   </tr>
%   <tr>
%     <td>Decision Tree</td>
%     <td>0.7533</td>
%     <td>1.0</td>
%   </tr>
%   <tr>
%     <td>SVM</td>
%     <td>0.7489</td>
%     <td>0.7896</td>
%   </tr>
%   <tr>
%     <td>KNN Classifier</td>
%     <td>0.7489</td>
%     <td>0.7896</td>
%   </tr>
% </table>

1.In Logistic Regression this accuracy is not valid as its testing accuracy is less
    than training accuracy.Training accuracy must always be greater than Testing
    Accuracy.

2.Naive Bayes accuracy we got is valid, and we may use it.

3.Decision Tree gave us the Training accuracy of 100% but this is not valid because
    we are using a part of training data for testing. At the time of training,
    decision tree gained knowledge about that data, and now if you give same data to
    predict it will give exactly same value. That's why decision tree producing
    correct results every time.

4.Support Vector Machine has given descent accuracy but not as much as Naive Bayes.

5.KNN classifier has also given same accuracy as of SVM.

#Building a Prediction System

So from above conclusions we may either use SVM or KNN. Here we will be using SVM
"""
# (pd.Series(modl.feature_importances_, index=X.columns)
#    .plot(kind='barh'))

dataset.tail()

inputData=(5,121,72,23,112,26.2,0.245,30)
#inputData=(1,126,60,0,0,30.1,0.349,47) for this data at 766 LR,NB,gave wrong outputs.
```

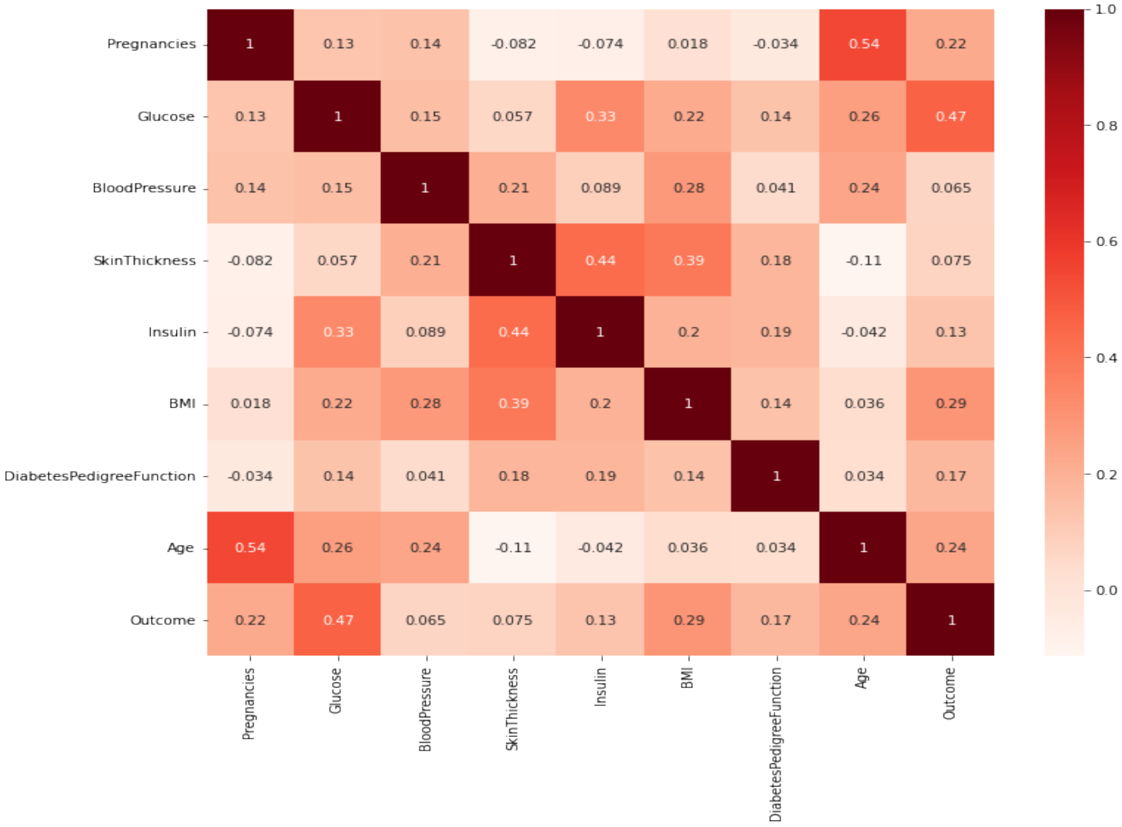
```
"""Convert input data to numpy array"""

inputDataNumpyArray = np.asarray(inputData)

"""Reshape the numpyy array as we are predicting for only one instance"""

inputDataReshaped = inputDataNumpyArray.reshape(1,-1)

prediction = modelSVM.predict(inputDataReshaped)
print(prediction)

if (prediction[0]==0):
  print("The patient doesn’t have Diabetes. ")
else:
  print("The patient is Diabetic. ")
```

12

Result Analysis

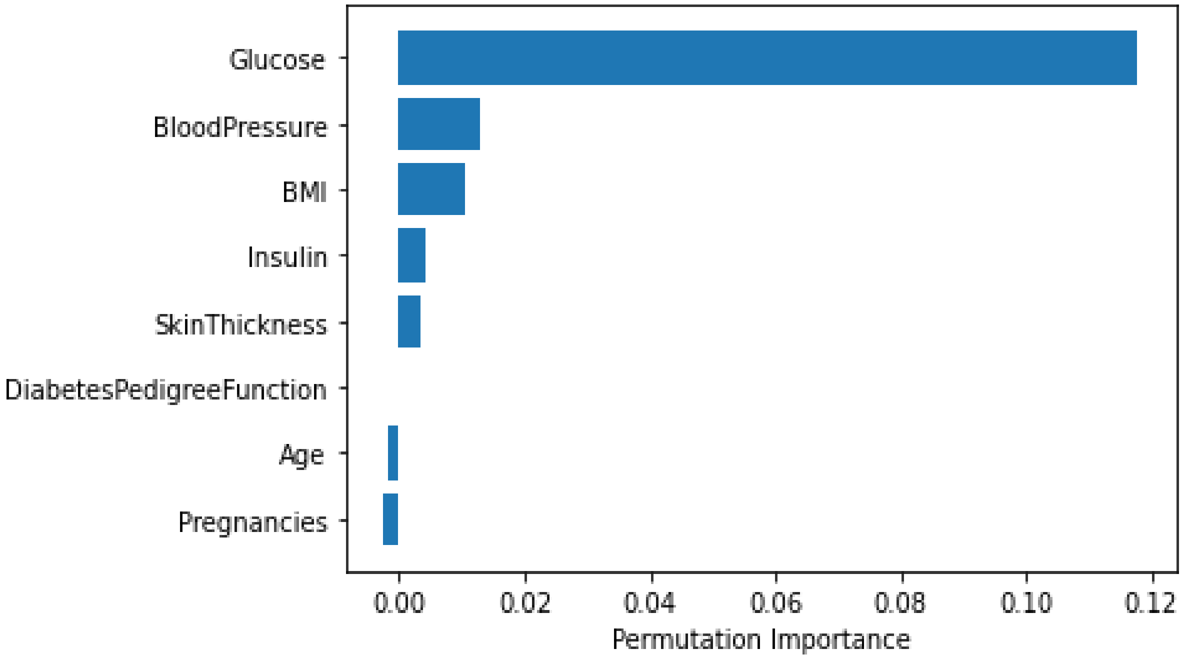


12.1 Correlation Matrix



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1	0.13	0.14	-0.082	-0.074	0.018	-0.034	0.54	0.22
Glucose	0.13	1	0.15	0.057	0.33	0.22	0.14	0.26	0.47
BloodPressure	0.14	0.15	1	0.21	0.089	0.28	0.041	0.24	0.065
SkinThickness	-0.082	0.057	0.21	1	0.44	0.39	0.18	-0.11	0.075
Insulin	-0.074	0.33	0.089	0.44	1	0.2	0.19	-0.042	0.13
BMI	0.018	0.22	0.28	0.39	0.2	1	0.14	0.036	0.29
DiabetesPedigreeFunction	-0.034	0.14	0.041	0.18	0.19	0.14	1	0.034	0.17
Age	0.54	0.26	0.24	-0.11	-0.042	0.036	0.034	1	0.24
Outcome	0.22	0.47	0.065	0.075	0.13	0.29	0.17	0.24	1

12.2 Dependent Attribute



13

Application

As the technology in each domain is progressing rapidly, this Machine Learning based Diabetes Prediction Model can also come in normal usage.This can also be updated with the future technological trends.It can adapt the changes without much complications.Now-a-days the covid self testing kits are available in the market similarly this can also be used for testing the Diabetic diseases without going to the hospitals.

14

Constraints

*Medical tests are required to get the accurate values of insulin , glucose , blood pressure of patients.
*Results may change as we are testing on human body not a machine.

15

Software/Hardware Resources

*Software Resources:

1. Python 3.9
2. Jupyter notebook
3. Github

*Hardware Resources:

1. 4GB RAM
2. 1GB Storage space
3. Intel core i3 processor

16

Conclusion

One of the important real-world medical problems is the detection of diabetes at its early stage. In this study, systematic efforts are made in designing a system which results in the prediction of diabetes. During this work, five machine learning classification algorithms are studied and evaluated on various measures. Experiments are performed on john Diabetes Database. Experimental results determine the adequacy of the designed system with an achieved accuracy of 80% Using Decision Tree Algorithm. In future, the designed system with the used machine learning classification algorithms can be used to predict or diagnose other diseases. The work can be extended and improved for the automation of diabetes analysis including some other machine learning algorithms.