

**EXPLORATORY DATA ANALYTICS**

REVIEW 2

PREDICTION OF DIABETES USING MACHINE LEARNING TECHNIQUES

S DEEPACHANDRAN 21MIA1078

V AMITH REDDY 21MIA1097

D SONAKSHI 21MIA1082

AATHMIKA V 21MIA1103

SANSKRITI R 21MIA1093

**ABSTRACT:**

Diabetes is one of the major diseases affected by most of the people in the world. Actually, it is a chronic disease that affects the body's ability to regulate blood sugar levels.

Nearly one out of six people in the world are been affected by diabetes mellitus and nearly 7 lakh people died because of Diabetes in 2020.

So our aim of our project is to predict diabetes use of factors such as age, glucose, pregnancy, blood pressure, skin thickness, insulin, BMI etc. These factors can give a clear prediction of diabetes.

**INTRODUCTION:**

Diabetes is a chronic (long-lasting) health condition that affects how your body turns food into energy.

Your body breaks down most of the food you eat into sugar (glucose) and releases it into your bloodstream. When your blood sugar goes up, it signals your pancreas to release insulin. Insulin acts like a key to let the blood sugar into your body’s cells for use as energy.

With diabetes, your body doesn’t make enough insulin or can’t use it as well as it should. When there isn’t enough insulin or cells stop responding to insulin, too much blood sugar stays in your bloodstream. Over time, that can cause serious health problems, such as heart disease, vision loss, and kidney disease. There are three main types of diabetes: type 1, type 2, and gestational diabetes (diabetes while pregnant)

**Type 1 Diabetes**

Type 1 diabetes is thought to be caused by an autoimmune reaction (the body attacks itself by mistake). This reaction stops your body from making insulin. Approximately 5-10% of the people who have diabetes have type 1. Symptoms of type 1 diabetes often develop quickly. It’s usually diagnosed in children, teens, and young adults. If you have type 1 diabetes, you’ll need to take insulin every day to survive. Currently, no one knows how to prevent type 1 diabetes.

**Type 2 Diabetes**

With type 2 diabetes, your body doesn’t use insulin well and can’t keep blood sugar at normal levels. About 90-95% of people with diabetes have type 2. It develops over many years and is usually diagnosed in adults (but more and more in children, teens, and young adults). You may not notice any symptoms, so it’s important to get your blood sugar tested if you’re at [risk](https://www.cdc.gov/diabetes/basics/risk-factors.html). Type 2 diabetes can be prevented or delayed with healthy lifestyle changes, such as:

* Losing weight.
* Eating healthy food.
* Being active.

**Gestational Diabetes**

Gestational diabetes develops in pregnant women who have never had diabetes. If you have gestational diabetes, your baby could be at higher risk for health problems. Gestational diabetes usually goes away after your baby is born. However, it increases your risk for type 2 diabetes later in life. Your baby is more likely to have obesity as a child or teen and develop type 2 diabetes later in life.

**LITERATURE SURVEY:**

|  |  |  |
| --- | --- | --- |
| Published on | Paper Name | Description |
| 09/2017 | A Survey on Prediction of  Diabetes Using Data  Mining Technique  K.Priyadarshini &  Dr.I.Lakshmi | Diabetes can be predicted by using the dataset of the patient . This paper is about how data mining is going to help in predition of diabetes .The techniques are Classification, Clustering, Decision Tree, Naive Bayes methods in order to predict the diabetes disease. This paper predicts that overall population affected by diabetes will also double the rate of diabetes of the population by the upcoming year . The main objective of this paper is that early prediction saves the life of a human . By using effective algorithm methods prediction can be done . After analysis decision Tree is the best data mining technique to predict Diabetes . |
| 02/2019 | Review of Predictive  Analysis Techniques for  Analysis Diabetes Risk  -Sonali Vyas , Rajeev  Ranjan , Navdeep Singh ,  Arohan Mathur | We use predictive analysis technique to detect fraud, reducing risk improve operation, diagnosis diseases improve marketing . Predictive analysis techniques are Machine Learning , Supervised  Learning , Unsupervised Learning . There is another algorithm named NATURE BASED ALGORITHM . There are 2 types of this algorithm :- Evolutionary and Swarm based algorithm. Evolutionary is a generic algorithm whereas Swarm is an example of PSO algorithm . |
| 06/2015 | A Data Mining Approach for the Diagnosis of Diabetes Mellitus using  Random Forest Classifier  -Kumari, Sonu, and  Archana Singh | Pima Indian diabetic database (PIDD) at the UCI Machine Learning Lab has been tested data mining algorithms to predict their accuracy in diabetic status . A random forest classifier is the assembly of tree-structured classifiers . This algorithm supplements the objects from array of input to every tree of forest. The elements of the unit vector are individually voted for classification by every single tree. The forest filters the most voted classifications out of the forest . This Random Forest Classifier performs with an accuracy of 99.7 % . |

|  |  |  |
| --- | --- | --- |
| 02/2010 | A Novel Classification Method for Diagnosis of Diabetes Mellitus using  Artificial Neural Networks  -T.Jayalakshmi and  Dr.A.Santhakumaran | Artificial Neural networks require complete set of data for an accurate classification . This paper works on missing value techniques to improve classification accuracy . This method predicts the possibility of diabetes with an accuracy of 92% .  This technique works with the data which has incomplete information . It is having a  fault tolerance capability . The loss of performance depends on how important the missing information is . |
| 2017 | Predicting Diabetes  Mellitus using Data  Mining Techniques  Comparative analysis of  Data Mining Classification  Algorithms  -J.Steffi,  Dr.R.Balasubramanian | Design of prediction models for diabetes diagnosis has been an active research area for the past decade. Most of the models found in literature are based on Classification Algorithms and artificial neural Networks . The objective of this paper was to diagnose by using ANN and demonstrated the need for pre processing and replacing missing values in the dataset that is considered . This paper specifies the techniques that is used for Diabetes Prediction and thereby bring a comparative analysis based on their accuracy , precision , sensitivity , specificity and F1 score . After the comparision we conclude that the Logistic Regression and C5.0 are equally good based of their accuracy measures , then comes the Naive Bayes and finally the ANN with the lowest accuracy . |
| 06/2017 | Application of data mining methods in diabetes prediction  -Komi.M , Li. J. Zhai,  Zhang.X | This paper specifies on 5 data mining techniques that includes GMM , SVM , Logistic Regression , ELM and ANN . As per the research done this paper concludes that ANN has the highest accuracy . The main advantage of using Data Mining Technique in prediction is that it has the ability to extract hidden knowledge from a huge amount of diabetes related data . |
| 10/2017 | Analysis and prediction of diabetes diseases using machine learning algorithm: Ensemble approach  -Rahul Joshi, Minyechil  Alehegn | Machine learning methods have different power in different data set. There are few machine learning techniques such as Hybrid Model , J48 , Naive Bayes , Random forest , KNN etc. Among these this paper concludes that Random forest model is the best because it accurately predict whether the patients have diabetes or not and along with that few insights can be drawn from the data via data analysis and visualization . |
| 03/2016 | Performance Analysis of Data Mining Classification | This paper specifies on techniques for predictive data mining tasks . Few of them |
|  | Techniques to Predict  Diabetes  -Perveen . S , Shahbaz. M,  Guergachi.A & Keshavjee  . K | are J48 decision tree , bagging ,  Adaboost . After the comparision we can conclude that Decision tree is one of the most powerful and widely applied  techniques for classification and prediction  . Evaluation of results indicates that adaboost ensemble method outperforms than bagging as well as standalone J48 decision tree . Furthermore, diverse individual techniques like Naïve Bayes, SVM and neural networks etc. can be incorporated as base learners in ensemble framework. |
| 04/2017 | An efficient instance selection algorithm for k nearest neighbor regression  -Song.Y, Liang.J, Lu. J &  Zhao | This paper is based on the KNN algorithm . The factor population becomes a major attribute that affects the efficiency of this algorithm . Large population results in slow execution speed and also involves large memory requirements .  In this paper an efficient instance selection algorithm DISKER is proposed .Firstly, the proposed algorithm removes the outlier instances. Secondly, it sorts the left instances by the difference between their true and predicted output provided by their neighbors. Finally, DISKER removes the instances with less effect on the regressor one by one. After performing the experiments we can conclude that DISKR has a more consistent performance but with a lower storage ratio . |
| 12/2016 | Predictive analysis of diabetes using J48 algorithm of classification  techniques -Pradeep.KR &  Naveen.NC | J48 decision tree algorithm is based on a machine learning decision tree classification which is based on Iterative Dichotomiser . It is helpful in examining the data categorically and continuously . In this algorithm less amount of data preparation is only needed during pre processing . This neither requires normalization of data nor scaling of data . The missing values also does not affect this process . |

**Keywords:**

Diabetes, Data Analysis, Exploratory Data Analytics, Model Building, SVM.

**Something New:**

Not only the predictions, we are going to visualize the predictions in the form of various visualizations as well as we are going to draw some insights from the data analysis visualisation.

**Proposed Method:**

The methods involved in Machine Learning Model:

1. Dataset collection
2. Exploratory Data Analysis
3. Data Pre-processing
4. Model Training
5. Making a Predictive System
6. **Dataset Collection**

This process involves collection of data from census or survey websites which helps in prediction and evaluation of results.

Totally there are 768 records and 9 attributes.

**Information about dataset attributes -**

Pregnancies: Number of pregnancies

Glucose: The Glucose level in blood

Blood Pressure: The Blood pressure measurement

Skin Thickness: The thickness of the skin

Insulin: The Insulin level in blood

BMI: The Body mass index

Diabetes Pedigree Function: The Diabetes percentage

Age: The age of the person

Outcome: The final result - 1 is Yes and 0 is No

|  |  |
| --- | --- |
| **ATTRIBUTES** | **DATA TYPES** |
| Pregnancies | Integer |
| Glucose | Integer |
| Blood Pressure | Integer |
| Skin Thickness | Integer |
| Insulin | Integer |
| BMI | Integer |
| Diabetes Pedigree Function | Integer |
| Age | Integer |
| Outcome | Boolean |

1. **Exploratory Data Analysis**

Second step is used to summarize and analyse the data using visualization techniques.

So now we need to summarize and analyse the diabetes dataset using Python.

**Required modules and functions**

import numpy as np

import pandas as pd

import statsmodels.api as sm

from lightgbm import LGBMClassifier

from sklearn.model\_selection import KFold

import warnings

warnings.simplefilter(action = "ignore")

import statsmodels.api as sm

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.neighbors import KNeighborsClassifier

from sklearn.svm import SVC

from sklearn.neural\_network import MLPClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier,GradientBoostingClassifier

from sklearn.preprocessing import scale, StandardScaler

from sklearn.model\_selection import train\_test\_split, GridSearchCV, cross\_val\_score

from sklearn.metrics import confusion\_matrix, accuracy\_score, mean\_squared\_error, r2\_score, roc\_auc\_score, roc\_curve, classification\_report

from sklearn.model\_selection import train\_test\_split

from sklearn import svm

from sklearn.metrics import accuracy\_score

from sklearn.linear\_model import LogisticRegression

from sklearn.neighbors import LocalOutlierFactor

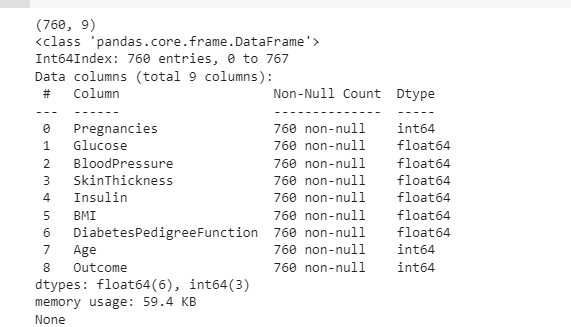
**Input the database into program**

df = pd.read\_csv("C:\\Users\\Asus\\OneDrive - vit.ac.in\\Desktop\\diabetes.csv")

**Size of dataset and summary**

print(df.shape)

print(df.info())



**Descriptive statistics of the dataset**

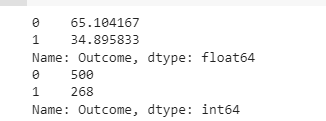
print(df.describe([0.1,0.25,0.5,0.75,0.9,0.99])).T



**Outcome analysis**

print(df["Outcome"].value\_counts()\*100/len(df))

print(df.Outcome.value\_counts())

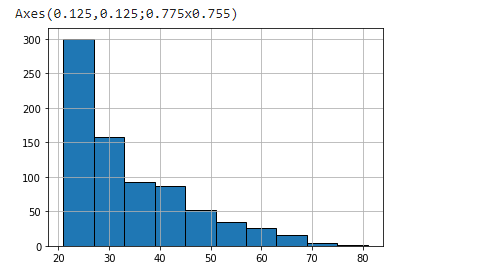
****

print("Max Age: ",str(df["Age"].max()))

print("Min Age: ",str(df["Age"].min()))



print(df[‘age’].hist())



**Histogram of all variables in dataset**

# Histogram and density graphs of all variables were accessed.

fig, ax = plt.subplots(4,2, figsize=(16,16))

sns.distplot(df.Age, bins = 20, ax=ax[0,0])

sns.distplot(df.Pregnancies, bins = 20, ax=ax[0,1])

sns.distplot(df.Glucose, bins = 20, ax=ax[1,0])

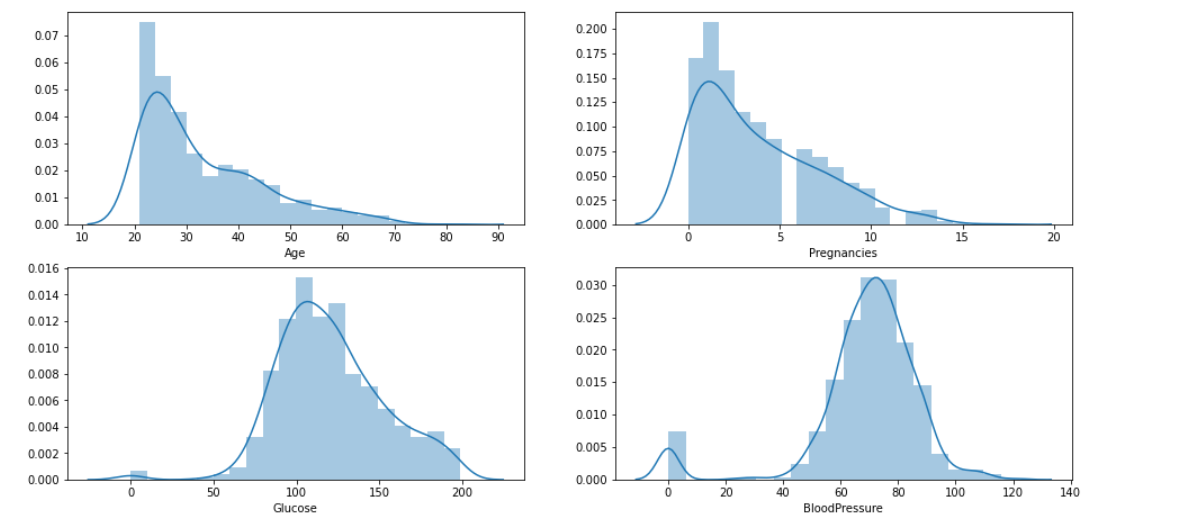
sns.distplot(df.BloodPressure, bins = 20, ax=ax[1,1])

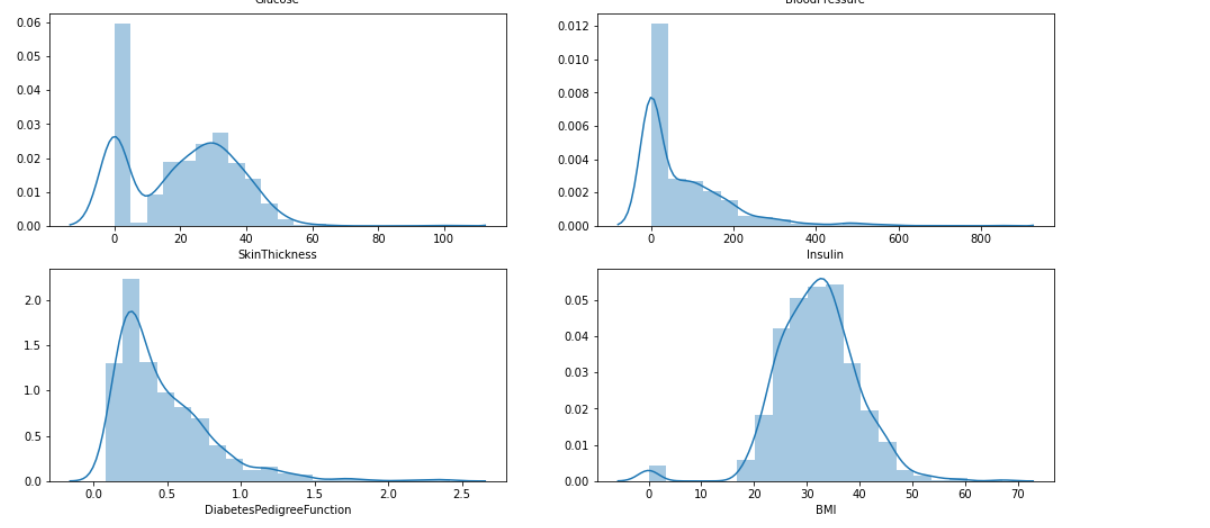
sns.distplot(df.SkinThickness, bins = 20, ax=ax[2,0])

sns.distplot(df.Insulin, bins = 20, ax=ax[2,1])

sns.distplot(df.DiabetesPedigreeFunction, bins = 20, ax=ax[3,0])

sns.distplot(df.BMI, bins = 20, ax=ax[3,1])





print(df.groupby("Outcome").agg({"Pregnancies":"mean"}))

print(df.groupby("Outcome").agg({"Age":"mean"}))

print(df.groupby("Outcome").agg({"Age":"max"}))

print(df.groupby("Outcome").agg({"Insulin":"mean"}))

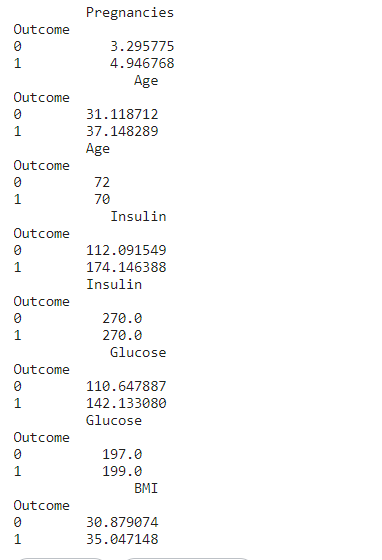
print(df.groupby("Outcome").agg({"Insulin":"max"}))

print(df.groupby("Outcome").agg({"Glucose":"mean"}))

print(df.groupby("Outcome").agg({"Glucose":"max"}))

print(df.groupby("Outcome").agg({"BMI":"mean"}))

​



f,ax=plt.subplots(1,2,figsize=(18,8))

df['Outcome'].value\_counts().plot.pie(explode=[0,0.1],autopct='%1.1f%%',ax=ax[0],shadow=True)

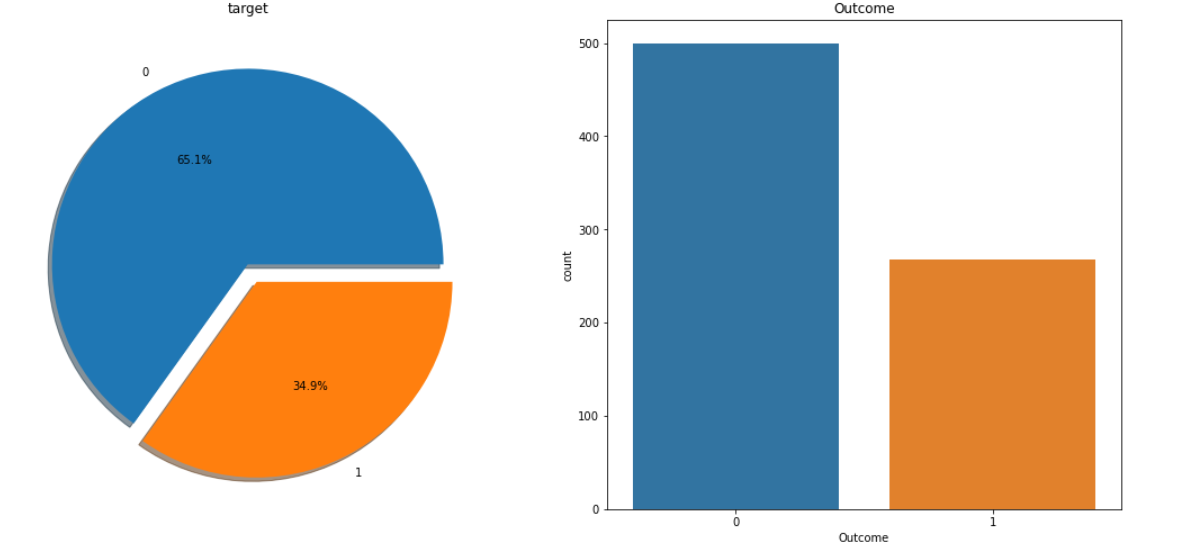
ax[0].set\_title('target')

ax[0].set\_ylabel('')

sns.countplot('Outcome',data=df,ax=ax[1])

ax[1].set\_title('Outcome')

plt.show()



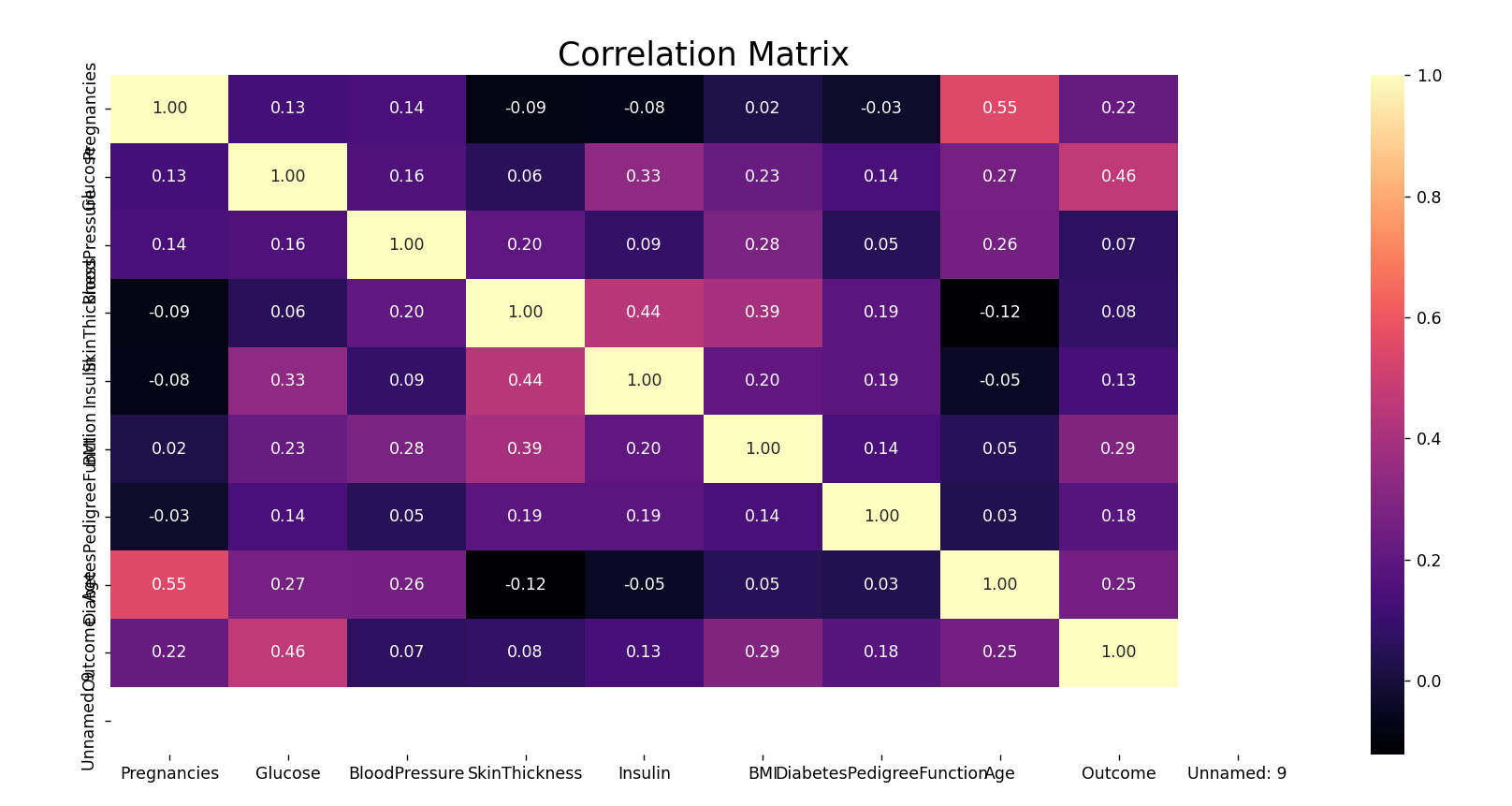
**Correlation Analysis:**

f, ax = plt.subplots(figsize= [20,15])

sns.heatmap(df.corr(), annot=True, fmt=".2f", ax=ax, cmap = "magma" )

ax.set\_title("Correlation Matrix", fontsize=20)

plt.show()



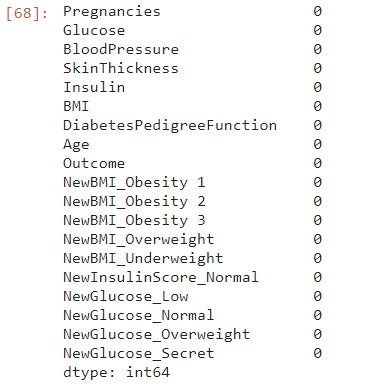
1. **Data Preprocessing**

**Missing Observation Analysis**

Replacing zero or null values with NaN

df[['Glucose','BloodPressure','SkinThickness','Insulin','BMI']] = df[['Glucose','BloodPressure','SkinThickness','Insulin','BMI']].replace(0,np.NaN)

df.isnull().sum()



import missingno as msno

msno.bar(df);

def median\_target(var):

temp = df[df[var].notnull()]

temp = temp[[var, 'Outcome']].groupby(['Outcome'])[[var]].median().reset\_index()

return temp

columns = df.columns

columns = columns.drop("Outcome")

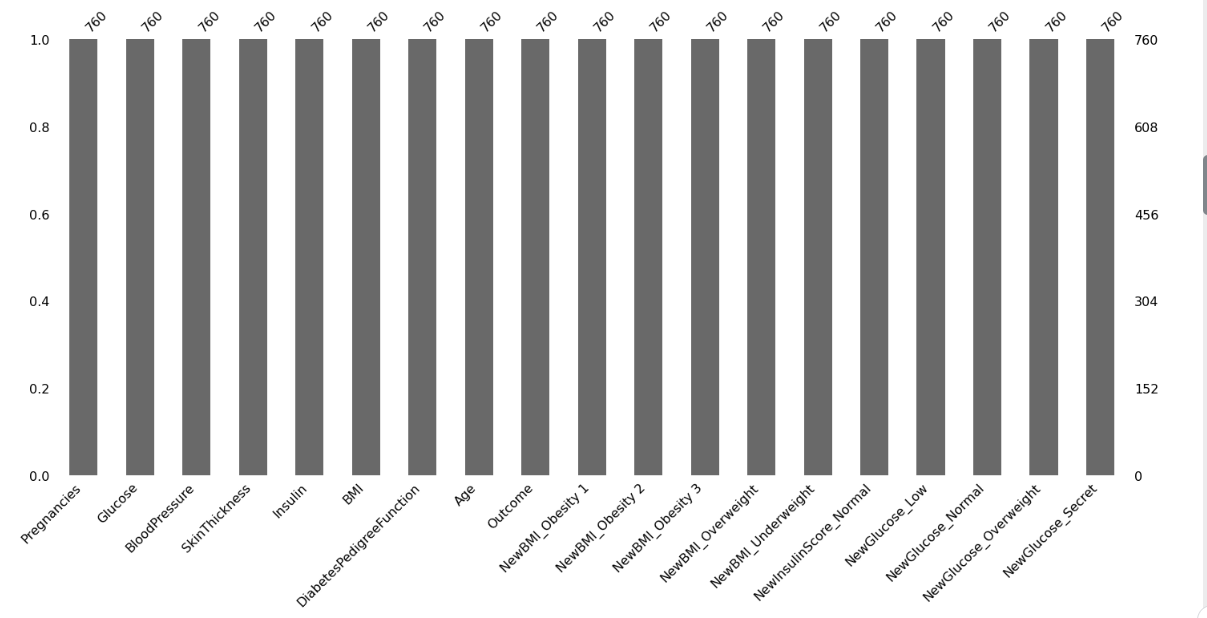
for i in columns:

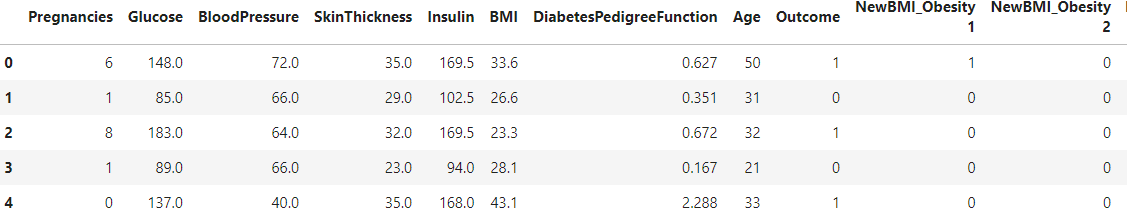
median\_target(i)

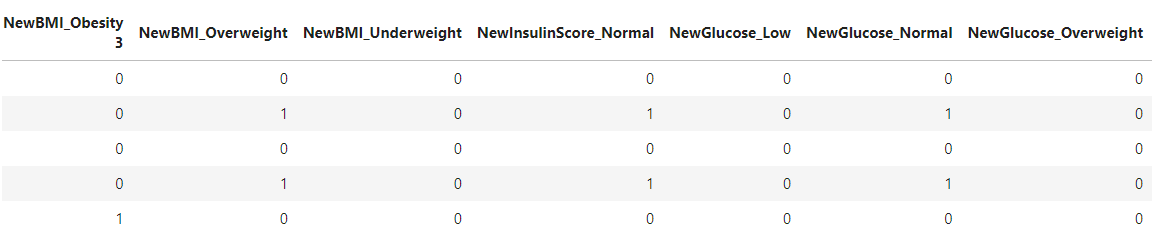
df.loc[(df['Outcome'] == 0 ) & (df[i].isnull()), i] = median\_target(i)[i][0]

df.loc[(df['Outcome'] == 1 ) & (df[i].isnull()), i] = median\_target(i)[i][1]

print(df.head())



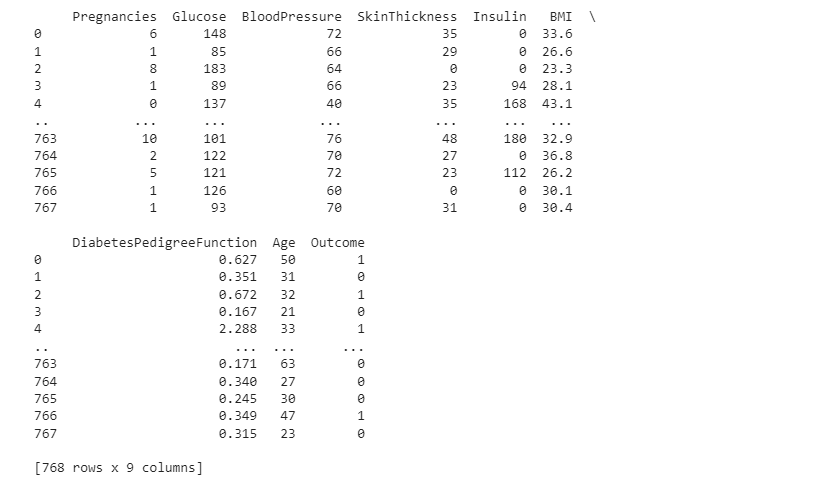






**Filling NaN values with Median values:**

print(df.head()) #Before filling NaN



def median\_target(var):

temp = df[df[var].notnull()]

temp = temp[[var, 'Outcome']].groupby(['Outcome'])[[var]].median().reset\_index()

return temp

columns = df.columns

columns = columns.drop("Outcome")

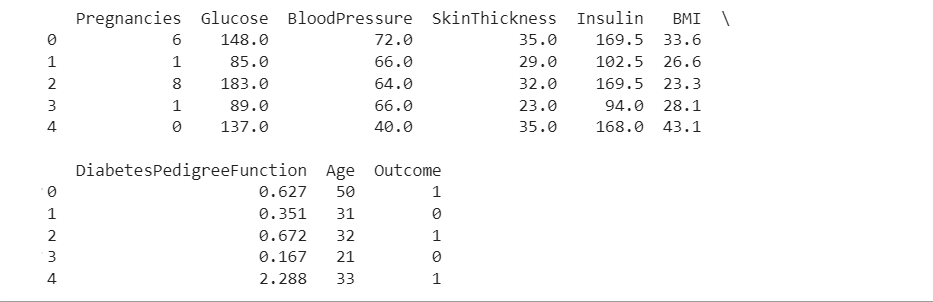
for i in columns:

median\_target(i)

df.loc[(df['Outcome'] == 0 ) & (df[i].isnull()), i] = median\_target(i)[i][0]

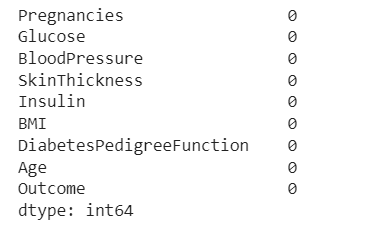
df.loc[(df['Outcome'] == 1 ) & (df[i].isnull()), i] = median\_target(i)[i][1]

print(df.head())



#Check of missing values

df.isnull().sum()



**Outlier Observation Analysis**

for feature in df:

Q1 = df[feature].quantile(0.25)

Q3 = df[feature].quantile(0.75)

IQR = Q3-Q1

lower = Q1- 1.5\*IQR

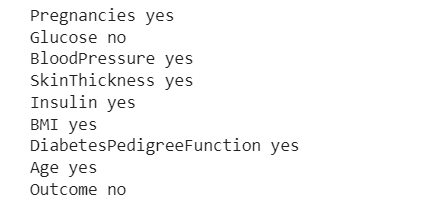
upper = Q3 + 1.5\*IQR

if df[(df[feature] > upper)].any(axis=None):

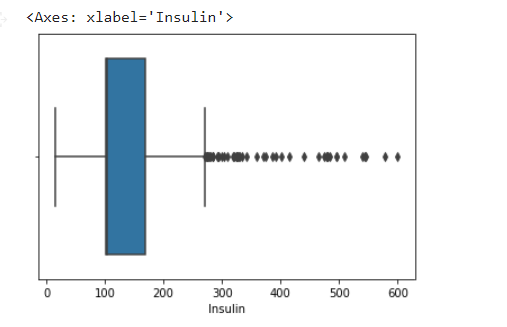
print(feature,"yes")

else:

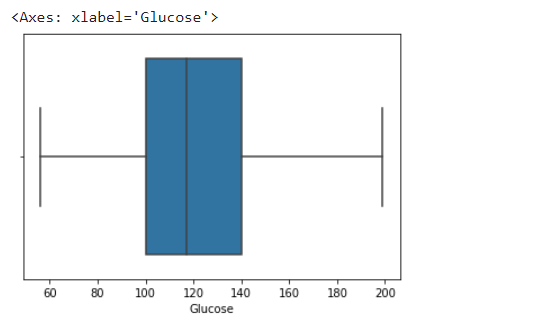
print(feature, "no")



# Check for outliers in SkinThickness



#Check for Outliers in Glucose



**Local Outlier Factor**

from sklearn.neighbors import LocalOutlierFactor

lof =LocalOutlierFactor(n\_neighbors= 10)

print(lof.fit\_predict(df))

df\_scores = lof.negative\_outlier\_factor\_

np.sort(df\_scores)[0:30]

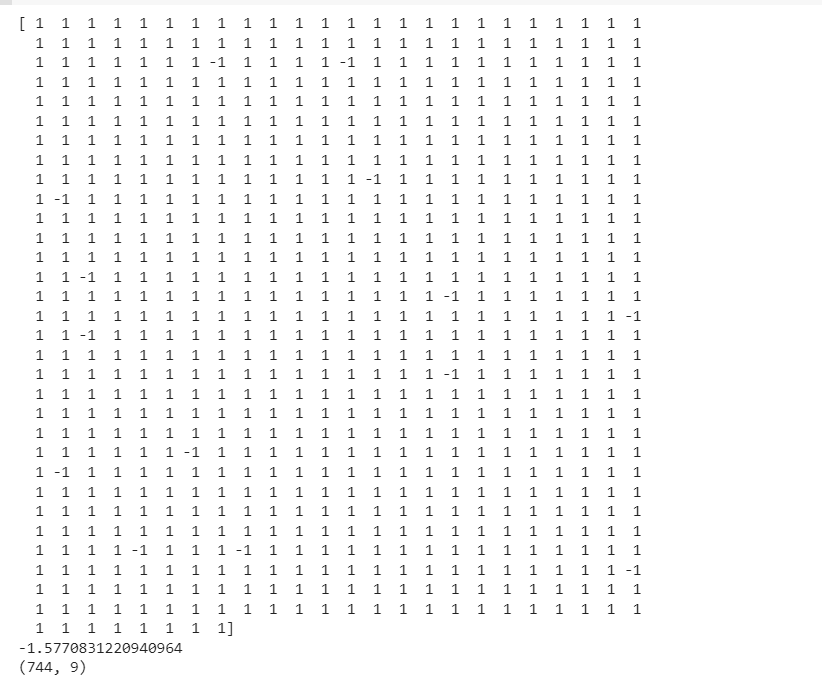
threshold = np.sort(df\_scores)[7]

print(threshold)

outlier = df\_scores > threshold

df = df[outlier]

print(df.shape)



**Data Standardization**

Since the ranges of each variable are different, it will be difficult for the machine learning to understand the data. So, standardizing the data will makes machine learning easy for prediction. In this process we need to use StandardScaler function from sklearn.

scale=StandardScaler()

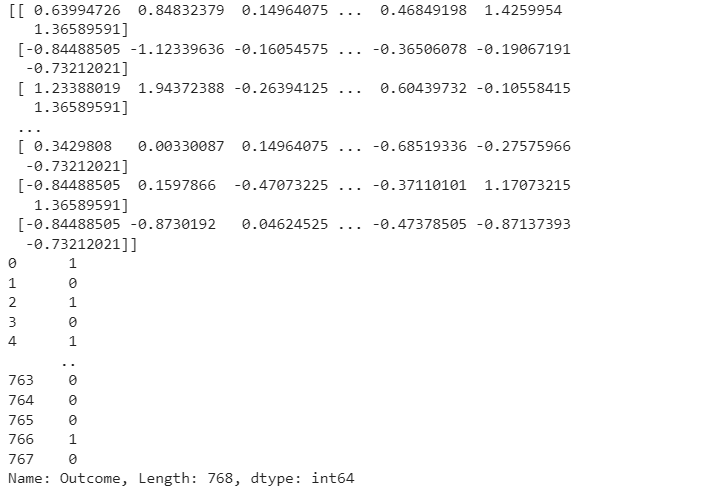
scale.fit(df)

x=scale.transform(df)

print(x) #x as data

y=df["Outcome"]

print(y) #y as label



**Train Test Split**

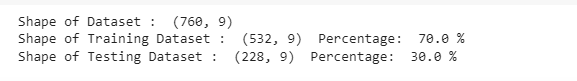
Splitting the data into Training Data and Test Data such that 70% of data is used for training and 30% of data is used for testing.

XTrain,XTest,YTrain,YTest=train\_test\_split(x,y,test\_size=0.3,stratify=y,random\_state=2)

print("Shape of Dataset : ",x.shape)

print("Shape of Training Dataset : ",XTrain.shape," Percentage: ",XTrain.shape[0]\*100/x.shape[0],"%")

print("Shape of Testing Dataset : ",XTest.shape," Percentage: ",XTest.shape[0]\*100/x.shape[0],"%")



1. **Model Training**

classify=svm.SVC(kernel="linear")

classify.fit(XTrain,YTrain)

XTrainPrediction=classify.predict(XTrain)

TrainAccuracy=accuracy\_score(XTrainPrediction,YTrain)

print("Accuracy Score of the Training Data: ",TrainAccuracy)

XTestPrediction=classify.predict(XTest)

TestAccuracy=accuracy\_score(XTestPrediction,YTest)

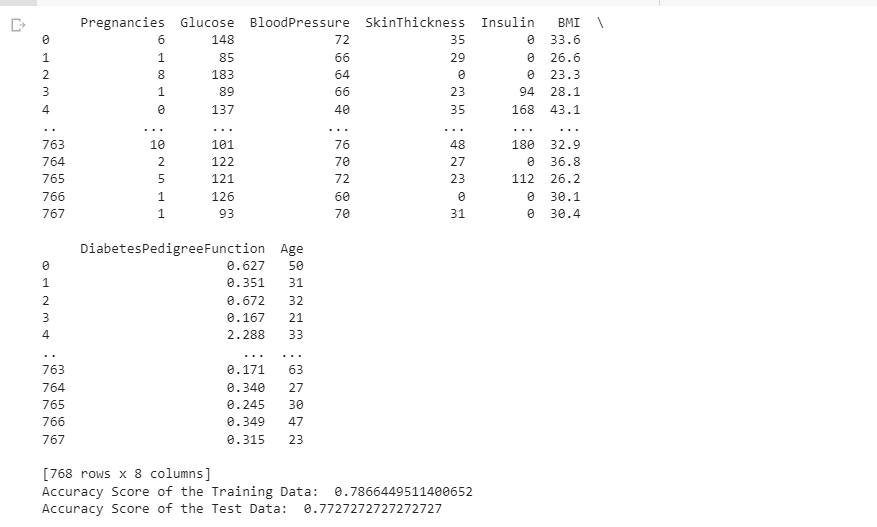
print("Accuracy Score of the Test Data: ",TestAccuracy)



Our model gives better accuracy because of Exploratory Data Analysis and

Data Pre-processing

The model accuracy without EDA and Data Pre-processing



There is no Overfitting of Data i.e., Train data accuracy is not so high and Test data accuracy is not very low. Both the accuracy scores are well balanced and hence it is a better fit.

1. **Making a Predictive System**

After Model Construction we have dome a predictive system, so that if we input the required data to the model it tells whether the person is diabetic or not

a=int(input("Enter Pregnancies: "))

b=int(input("Enter Glucose Level: "))

c=int(input("Enter Blood Pressure: "))

d=int(input("Enter Skin Thickness: "))

e=int(input("Enter Insulin Level: "))

f=float(input("Enter BMI: "))

g=float(input("Enter Diabetes Pedigree Function: "))

h=int(input("Enter Age: "))

i=(a,b,c,d,e,f,g,h)

nparray=np.asarray(i)

reshapedArray=nparray.reshape(1,-1)

stdData=scale.transform(reshapedArray)

predict=classify.predict(stdData)

if predict[0]==0:

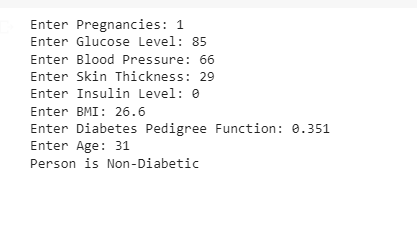
print("Person is Non-Diabetic")

else:

print("Person is Diabetic")

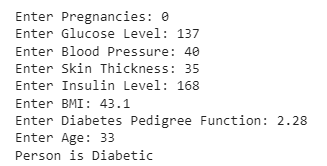
**Output for Non Diabetic Person taken from the Dataset**

1,85,66,29,0,26.6,0.351,31,0



**Output for Diabetic Person**

0,137,40,35,168,43.1,2.288,33,1

****

**Dataset link**

https://www.kaggle.com/datasets/deepachandrans/edapro

**Colab link**

https://colab.research.google.com/drive/1NrLcudBxWtoMAlGUprhffDajXtDxOFWI?usp=sharing