PROJECT DESIGN PHASE - 3

AI BASED DIABETES PREDICTION SYSTEM

INTRODUCTION :

A system is used to predict whether a patient has diabetes based on some of its health-related details such as BMI (Body Mass Index), blood pressure, Insulin, etc.

This system is only for females as the dataset used to make this system exclusively belongs to the females.

The accuracy level was 90% using the random forest algorithm, which is much higher when compared to other algorithms. In a recent paper [5], Mohan and Jain used the SVM algorithm to analyse and predict diabetes with the help of the Pima Indian Diabetes Dataset.

DATA PREPROCESSING :

Data preprocessing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining task.

Data preprocessing is an important step in the data mining process that involves cleaning and transforming raw data to make it suitable for analysis.

CODE :

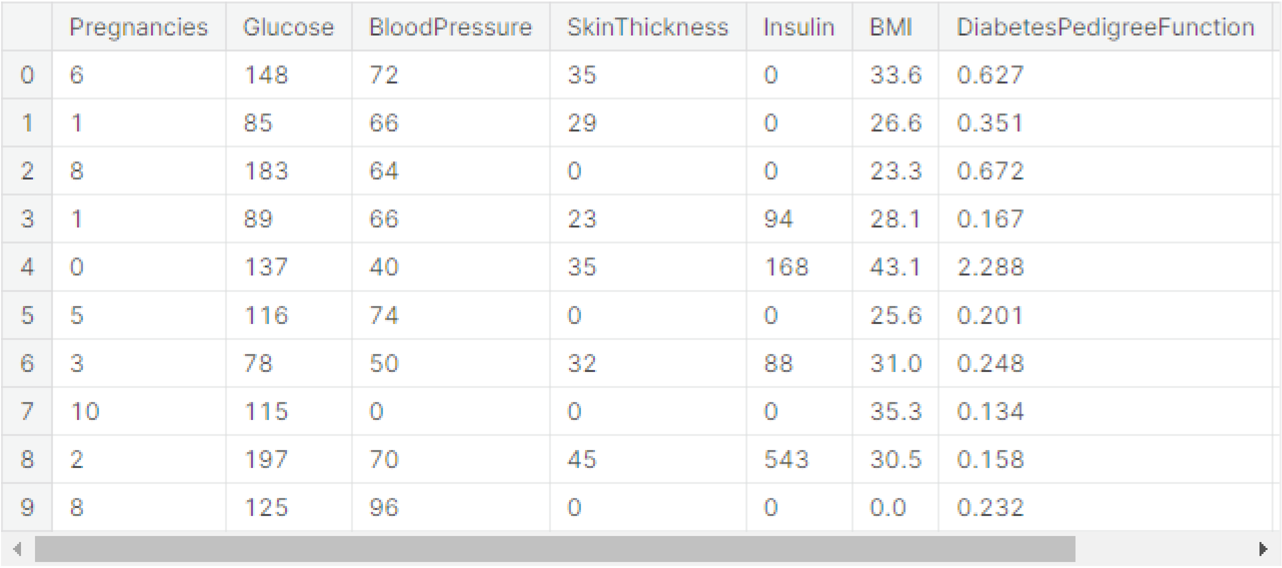
```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df=pd.read_csv('/kaggle/input/diabetes-data-set/diabetes.csv')
```

OUTPUT:



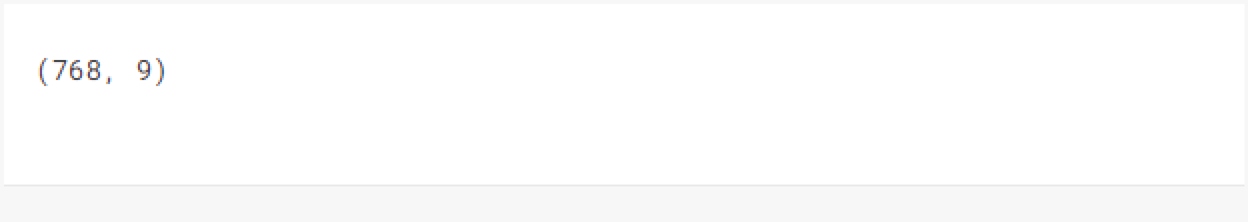
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
0	6	148	72	35	0	33.6	0.627
1	1	85	66	29	0	26.6	0.351
2	8	183	64	0	0	23.3	0.672
3	1	89	66	23	94	28.1	0.167
4	0	137	40	35	168	43.1	2.288
5	5	116	74	0	0	25.6	0.201
6	3	78	50	32	88	31.0	0.248
7	10	115	0	0	0	35.3	0.134
8	2	197	70	45	543	30.5	0.158
9	8	125	96	0	0	0.0	0.232

DATA CLEANING :

CODE :

```
df.shape()
```

OUTPUT:



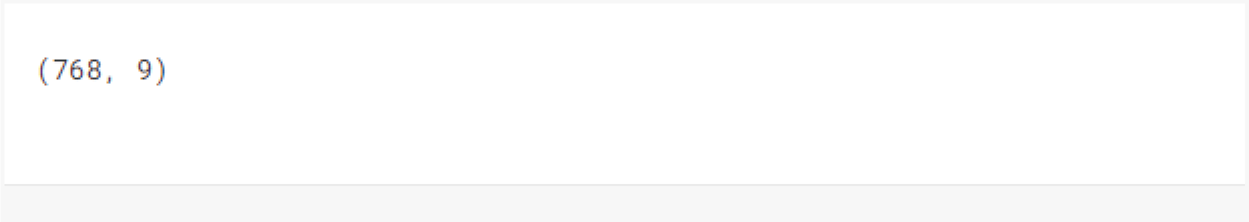
```
(768, 9)
```

CODE :

```
df=df.drop_duplicates()

df.shape()
```

OUTPUT :



```
(768, 9)
```

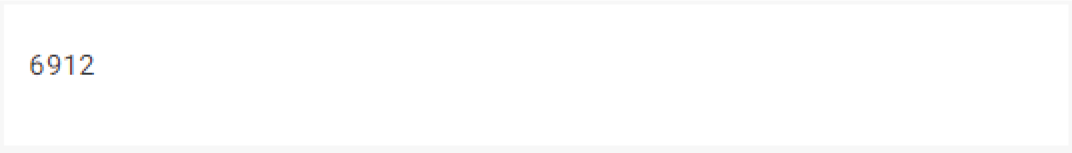
Check null Values:

CODE :

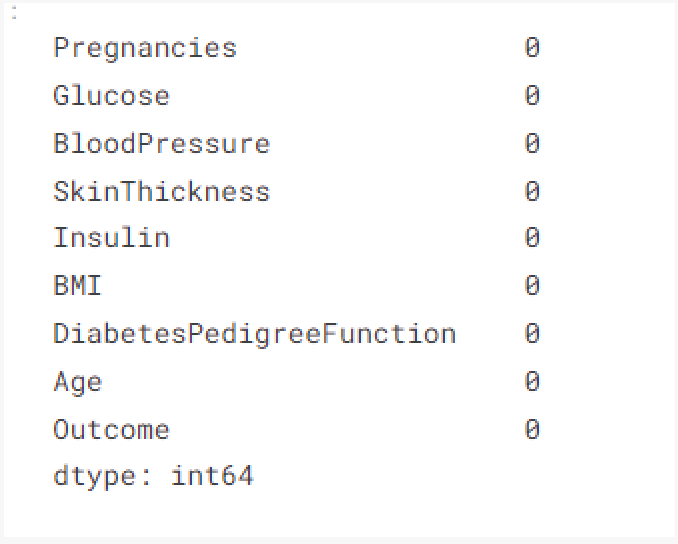
```
df.size()

df.isnull().sum()
```

OUTPUT :



```
6912
```



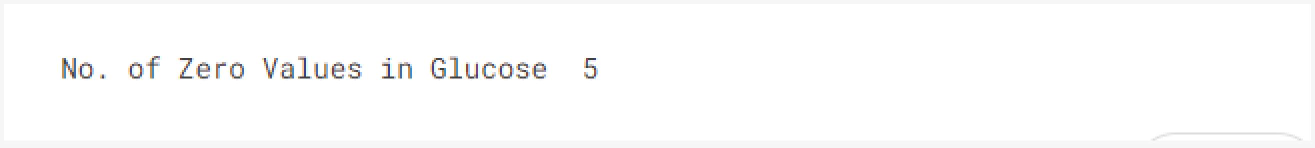
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Check the number of Zero Values in Dataset :

CODE :

```
print("No. of Zero Values in Glucose ", df[df['Glucose']==0].shape[0])
```

OUTPUT :

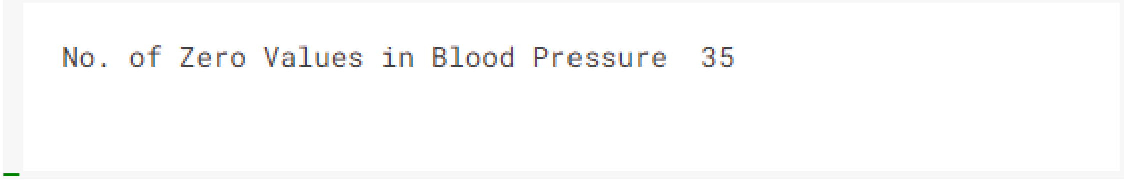


```
No. of Zero Values in Glucose  5
```

CODE :

```
print("No. of Zero Values in Blood Pressure df[df['BloodPressure']==0].shape[0])
```

OUTPUT :

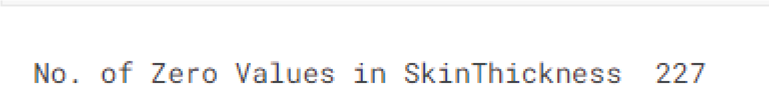


```
No. of Zero Values in Blood Pressure  35
```

CODE :

```
print("No. of Zero Values in SkinThickness ", df[df['SkinThickness']==0].shape[0])
```

OUTPUT :

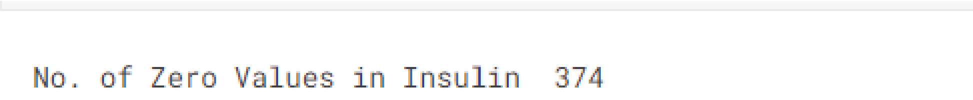


```
No. of Zero Values in SkinThickness  227
```

CODE :

```
print("No. of Zero Values in Insulin ", df[df['Insulin']==0].shape[0])
```

OUTPUT :

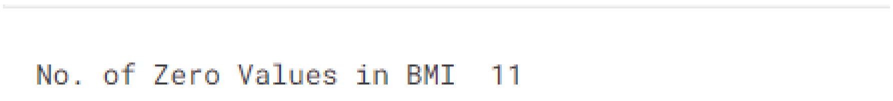


```
No. of Zero Values in Insulin  374
```

CODE :

```
print("No. of Zero Values in BMI ", df[df['BMI']==0].shape[0])
```

OUTPUT :



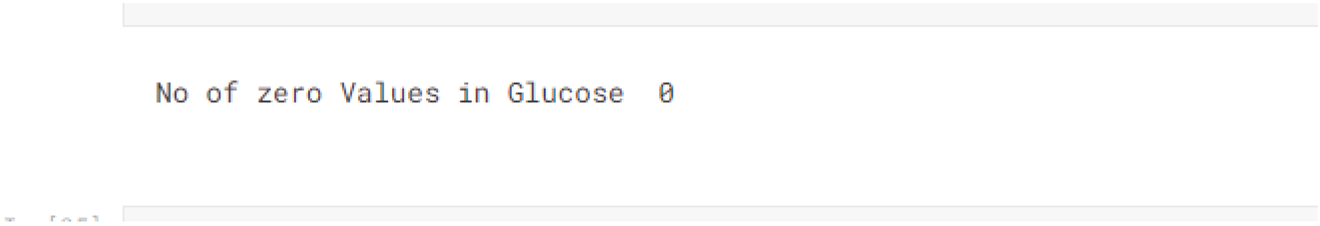
```
No. of Zero Values in BMI  11
```

Replace zeroes with mean of that Columns :

CODE :

```
df['Glucose']=df['Glucose'].replace(0, df['Glucose'].mean())
print('No of zero Values in Glucose ', df[df['Glucose']==0].shape[0])
```

OUTPUT :



```
No of zero Values in Glucose  0
```

CODE :

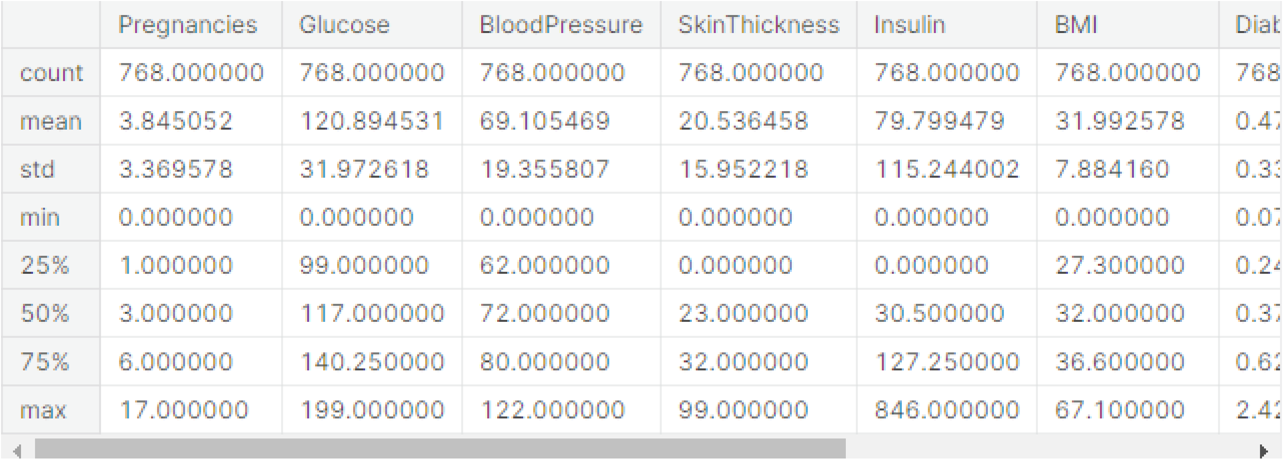
```
df['BloodPressure']=df['BloodPressure'].replace(0,    df['BloodPressure'].mean())
df['SkinThickness']=df['SkinThickness'].replace(0, df['SkinThickness'].mean())
df['Insulin']=df['Insulin'].replace(0, df['Insulin'].mean())
df['BMI']=df['BMI'].replace(0, df['BMI'].mean())
```

Validate the Zero Values :

CODE :

```
df.describe()
```

OUTPUT :



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diat
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.47
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.33
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.07
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.24
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.37
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.62
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.42

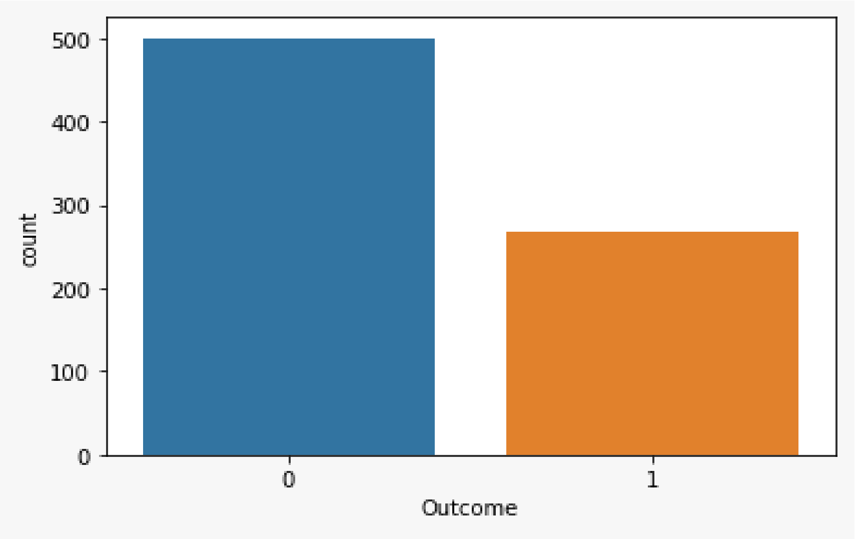
Data Visualization :

COUNT PLOT

CODE:

```
sns.countplot('Outcome',data=df)
```

OUTPUT :



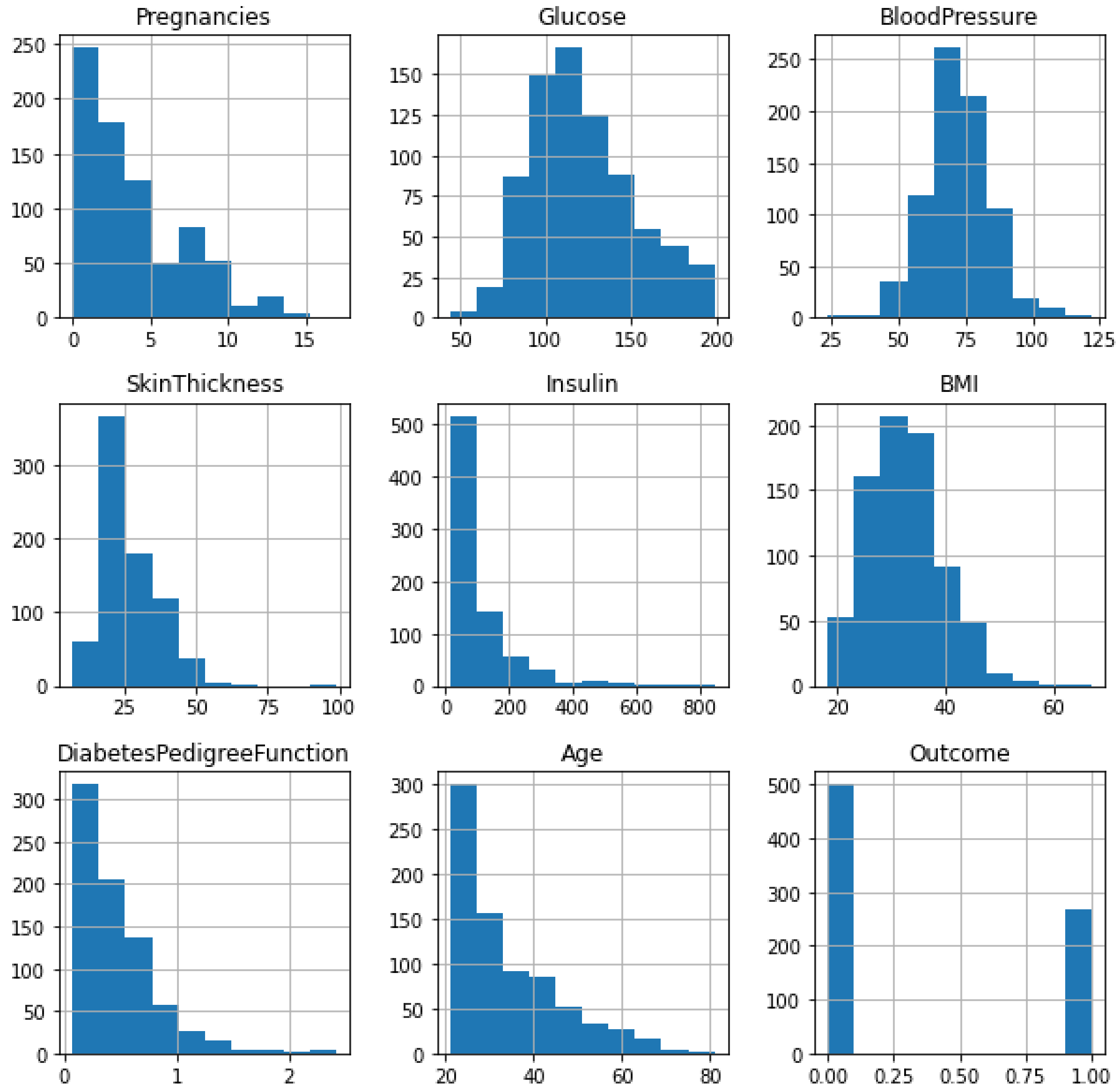
HISTOGRAM :

CODE:

```
df.hist(bins=10,figsize=(10,10))

plt.show()
```

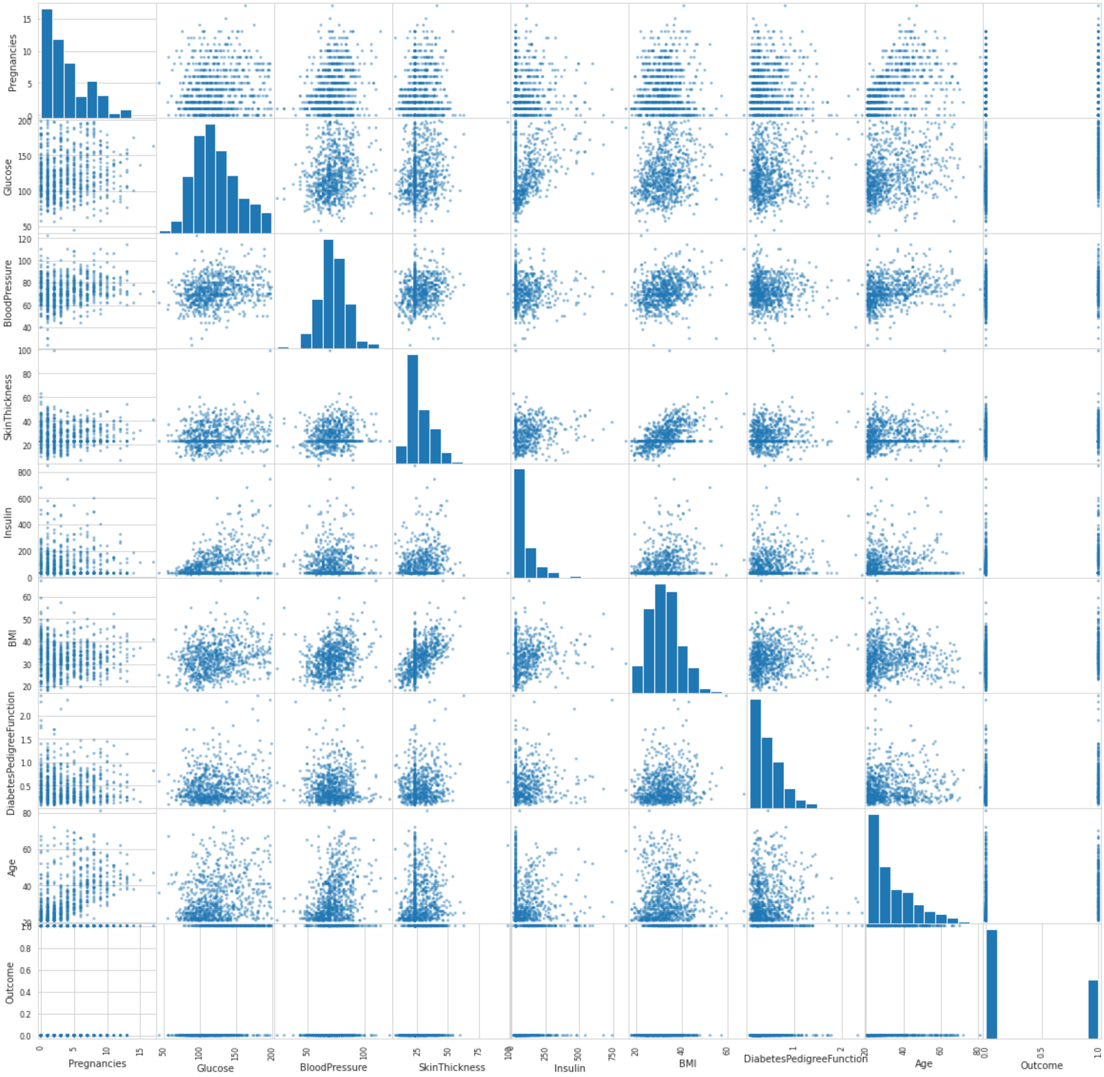
OUTPUT:



CODE:

```
from pandas.plotting import scatter_matrix
scatter_matrix(df,figsize=(20,20));
```

OUTPUT:

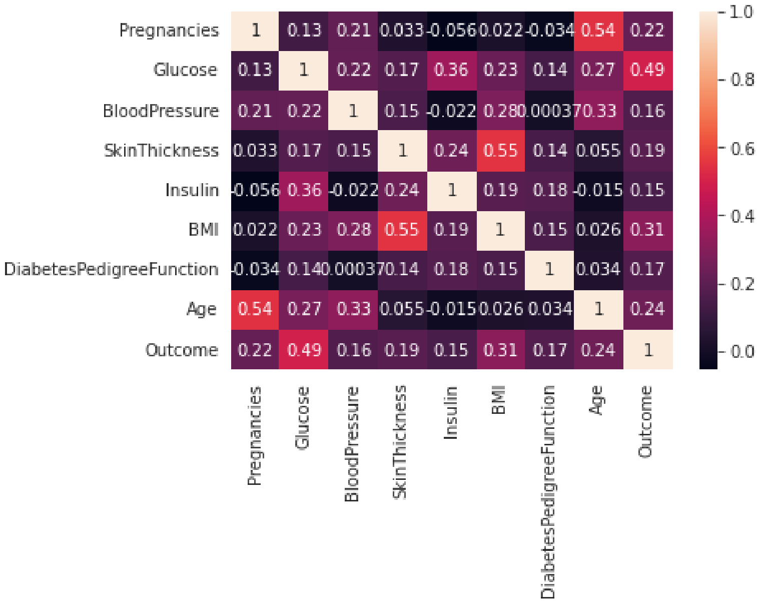


Feature Selection :

CODE:

```
corrmat=df.corr()
sns.heatmap(corrmat, annot=True)
```

OUTPUT:



CONCLUSION:

In this project we have imported the required libraries .Followed by libraries we have imported the dataset .We have done data cleaning,change ,replace the null values and then data visualization to perform preprocessing the dataset.