Predicting **Diabetes**

Diabetes is a chronic, metabolic disease characterized by elevated levels of blood glucose (or blood sugar), which leads over time to serious damage to the heart, blood vessels, eyes, kidneys and nerves. The most common is type 2 diabetes, usually in adults, which occurs when the body becomes resistant to insulin or doesn't make enough insulin. In the past three decades the prevalence of type 2 diabetes has risen dramatically in countries of all income levels. Type 1 diabetes, once known as juvenile diabetes or insulin-dependent diabetes, is a chronic condition in which the pancreas produces little or no insulin by itself. For people living with diabetes, access to affordable treatment, including insulin, is critical to their survival.

Data Description :-

```
Pregnancies: Number of times pregnant
Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
BloodPressure: Diastolic blood pressure (mm Hg)
SkinThickness: Triceps skin fold thickness (mm)
Insulin: 2-Hour serum insulin (mu U/ml)
BMI: Body mass index (weight in kg/(height in m)^2)
DiabetesPedigreeFunction: Diabetes pedigree functionr
Age: Age (years)
Cabin : Cabin Number
Outcome: Class variable (0 or 1)
```

Importing numpy ,pandas ,matplotlib ,seaborn

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
%matplotlib inline
```

from sklearn package	importing modules
preprocessing	StandardScaler
model_selection	train_test_split, GridSearchCV
metrics	accuracy_score, confusion_matrix

```
data.head(10)
```

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

```
data.shape
```

```
(768, 9)
```

```
data.info
```

```
<bound method DataFrame.info of      Pregnancies  Glucose  ...  Age  Outcome
0              6      148  ...   50        1
1              1       85  ...   31        0
2              8      183  ...   32        1
3              1       89  ...   21        0
4              0      137  ...   33        1
..           ...      ...  ...  ...      ...
763           10      101  ...   63        0
764            2      122  ...   27        0
765            5      121  ...   30        0
766            1      126  ...   47        1
767            1       93  ...   23        0

[768 rows x 9 columns]>
```

Some stats about data

min	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.30000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.00000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.60000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.10000

Number of nulls in each column

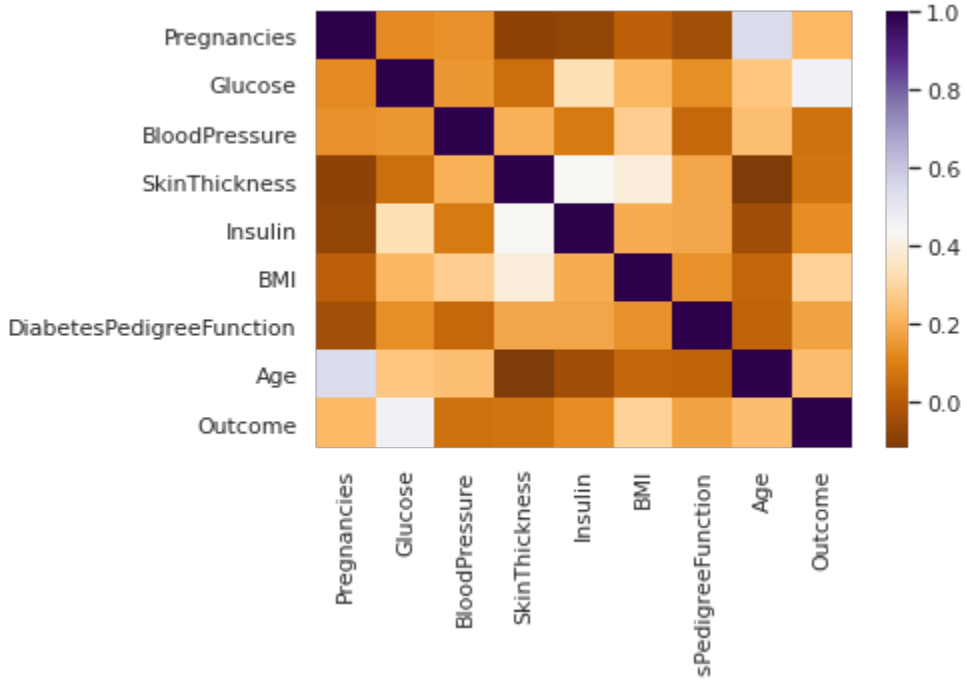
```
data.isnull().sum()
```

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

Correlation Matrix

```
sns.heatmap(data.corr(),cbar=True,cmap='PuOr',annot=False)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb3a6996b50>
```

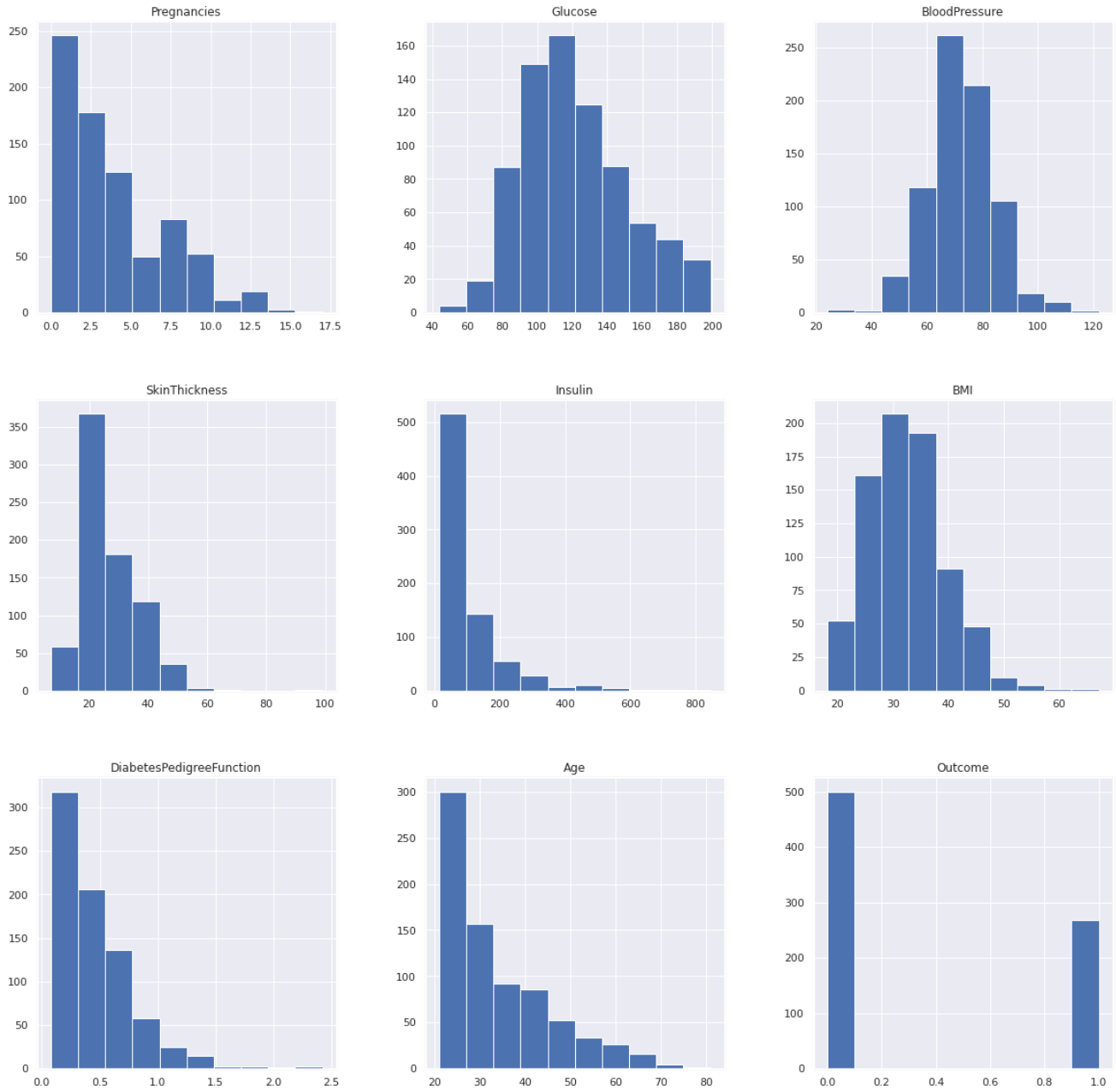


```
col=['Glucose' ,'BloodPressure' ,'SkinThickness', 'Insulin' ,'BMI']
```

Replacing missing data with 0.

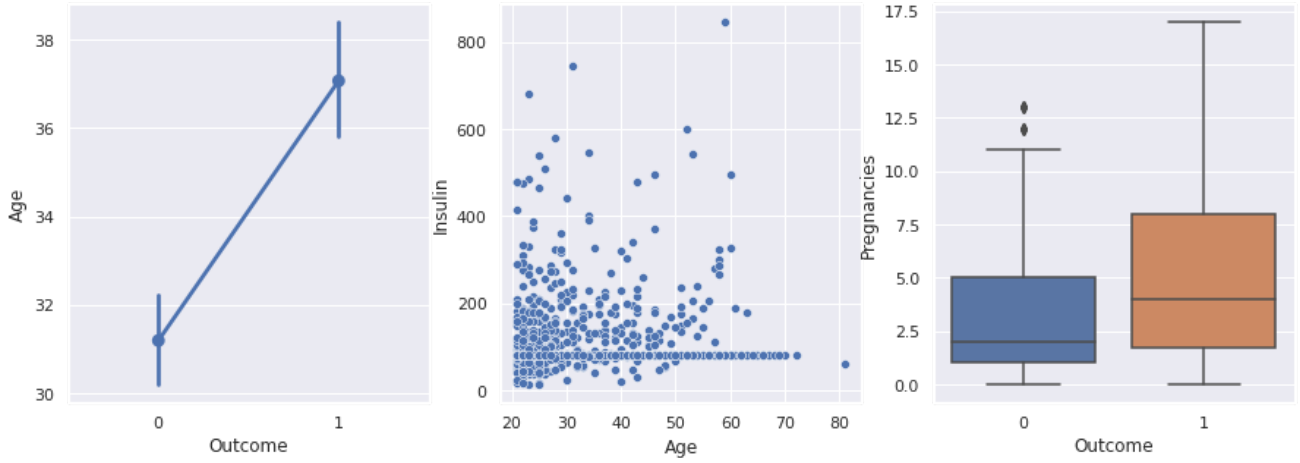
```
for i in col:
  data[i].replace(0,data[i].mean(),inplace=True)
```

```
p=data.hist(figsize = (20,20))
```



```
fig, axes = plt.subplots(1, 3,figsize=(15,5))
sns.pointplot(ax= axes[0],x='Outcome', y= 'Age', data=data)
sns.scatterplot(ax = axes[1],x='Age',y='Insulin',data=data)
sns.boxplot(ax = axes[2],x='Outcome',y='Pregnancies',data=data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb3a031c350>
```



```
data.var()
```

```
Pregnancies                  11.354056
Glucose                     926.351048
BloodPressure               146.795798
SkinThickness                92.760798
Insulin                    8663.952981
BMI                          47.270761
DiabetesPedigreeFunction      0.109779
Age                         138.303046
Outcome                       0.227483
dtype: float64
```

```
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X =  pd.DataFrame(sc_X.fit_transform(data.drop(["Outcome"],axis = 1),),
        columns=['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insul
```

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=9)                #knn classifier
knn.fit(X_train,Y_train)

knn_acc = accuracy_score(Y_test,knn.predict(X_test))


print("Train Set Accuracy:"+str(accuracy_score(Y_train,knn.predict(X_train))*100))
print("Test Set Accuracy:"+str(accuracy_score(Y_test,knn.predict(X_test))*100))
```

```
Train Set Accuracy:82.12290502793296
Test Set Accuracy:71.42857142857143
```

```
gbc = GradientBoostingClassifier()
gbc.fit(X_train, Y_train)


gbc_acc=accuracy_score(Y_test,gbc.predict(X_test))
```