Project-Analysis_on_Diabetes_Patients

In [1]:
```
'''
Hello Everyone
I am Umang
I am going to Analyse this data on Diabetes Patients as my Project
Submitted to: Meriskill
'''
```

Out[1]:
```
'\nHello Everyone\nI am Umang\nI am going to Analyse this data on Diabetes Patients as my Pro
ject\nSubmitted to: Meriskill\n'
```

In [2]:
```
'''
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 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. In particular, all patients here are females at
least 21 years old of Pima Indian heritage.
'''
```

Out[2]:
```
'\nThis dataset is originally from the National Institute of Diabetes \nand Digestive and Kid
ney Diseases. The objective of the dataset is \nto diagnostically predict whether a patient h
as diabetes based on \ncertain diagnostic measurements included in the dataset. Several \ncon
straints were placed on the selection of these instances from a \nlarger database. In particu
lar, all patients here are females at \nleast 21 years old of Pima Indian heritage.\n'
```

In [3]:
```
#importing the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
```

In [4]:
```
# Loading the Dataset
patient_data = pd.read_csv(r"C:\Users\umang\Desktop\meriskill intern\project 2\diabetes.csv")
```

Exploring the Dataset

In [5]:
```
print(patient_data)
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
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
..           ...      ...            ...            ...      ...   ...
763           10      101             76             48      180  32.9
764            2      122             70             27        0  36.8
765            5      121             72             23      112  26.2
766            1      126             60              0        0  30.1
767            1       93             70             31        0  30.4

     DiabetesPedigreeFunction  Age  Outcome
0                       0.627   50        1
1                       0.351   31        0
2                       0.672   32        1
3                       0.167   21        0
4                       2.288   33        1
..                        ...  ...      ...
763                     0.171   63        0
764                     0.340   27        0
765                     0.245   30        0
766                     0.349   47        1
767                     0.315   23        0

[768 rows x 9 columns]
```

In [6]:
```
patient_data.columns
```

Out[6]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [7]:
```
patient_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [8]:
```
print(patient_data.describe())
```

```
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000
mean      3.845052  120.894531      69.105469      20.536458   79.799479
std       3.369578   31.972618      19.355807      15.952218  115.244002
min       0.000000    0.000000       0.000000       0.000000    0.000000
25%       1.000000   99.000000      62.000000       0.000000    0.000000
50%       3.000000  117.000000      72.000000      23.000000   30.500000
75%       6.000000  140.250000      80.000000      32.000000  127.250000
max      17.000000  199.000000     122.000000      99.000000  846.000000

              BMI  DiabetesPedigreeFunction          Age     Outcome
count  768.000000                768.000000  768.000000  768.000000
mean    31.992578                  0.471876   33.240885    0.348958
std      7.884160                  0.331329   11.760232    0.476951
min      0.000000                  0.078000   21.000000    0.000000
25%     27.300000                  0.243750   24.000000    0.000000
50%     32.000000                  0.372500   29.000000    0.000000
75%     36.600000                  0.626250   41.000000    1.000000
max     67.100000                  2.420000   81.000000    1.000000
```

In [9]:
```
# We have data of 768 patients
# The numerical columns of Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin,
# BMI, DiabetesPredigreeFunction, and Age are independent variables
# The outcome column is a dependent variable. It shows whether the person has diabetes.
# In outcome column 1 ='Person has diabetes' and 0='Person not diabetic'
```

Cleaning the dataset

In [10]:
```
patient_data.isnull().sum()
```

Out[10]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [11]:
```
patient_data.duplicated().sum()
```

Out[11]:
```
0
```

In [12]:
```
# No Null values in our data
# No duplicate values in our data
```

Correlation Matrix

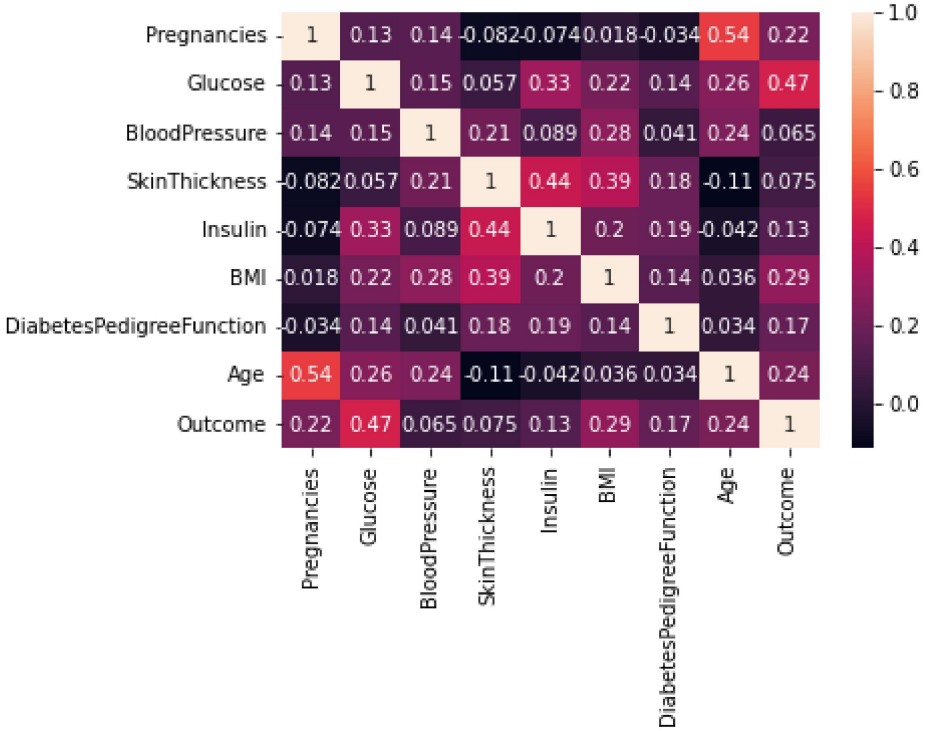
In [13]:
```
correlation = patient_data.corr()
print(correlation)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	\
Pregnancies	1.000000	0.129459	0.141282	-0.081672	
Glucose	0.129459	1.000000	0.152590	0.057328	
BloodPressure	0.141282	0.152590	1.000000	0.207371	
SkinThickness	-0.081672	0.057328	0.207371	1.000000	
Insulin	-0.073535	0.331357	0.088933	0.436783	
BMI	0.017683	0.221071	0.281805	0.392573	
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	
Age	0.544341	0.263514	0.239528	-0.113970	
Outcome	0.221898	0.466581	0.065068	0.074752	

	Insulin	BMI	DiabetesPedigreeFunction	\
Pregnancies	-0.073535	0.017683	-0.033523	
Glucose	0.331357	0.221071	0.137337	
BloodPressure	0.088933	0.281805	0.041265	
SkinThickness	0.436783	0.392573	0.183928	
Insulin	1.000000	0.197859	0.185071	
BMI	0.197859	1.000000	0.140647	
DiabetesPedigreeFunction	0.185071	0.140647	1.000000	
Age	-0.042163	0.036242	0.033561	
Outcome	0.130548	0.292695	0.173844	

	Age	Outcome
Pregnancies	0.544341	0.221898
Glucose	0.263514	0.466581
BloodPressure	0.239528	0.065068
SkinThickness	-0.113970	0.074752
Insulin	-0.042163	0.130548
BMI	0.036242	0.292695
DiabetesPedigreeFunction	0.033561	0.173844
Age	1.000000	0.238356
Outcome	0.238356	1.000000

In [14]:
```
#creating a correlation heatmap
sns.heatmap(correlation, xticklabels = correlation.columns,
            yticklabels = correlation.columns, annot = True)
plt.show()
```



In [15]:
```
# We notice that there is 0.54 positive correlation between Pregnancies and Age
# Showing that higher the Age, more the pregnancies.
# We also notice that all the factors affect the outcome of being Diabetic Positively
# Though their effect varies but more of these factors more chances of being diabetic
# Glucose and Diabetes are positively correlated with 0.47
                                                    Age are correlated with range 0.2 to 0.3
```

Training and Predicting the Model

In [16]:
```
x = patient_data.drop("Outcome", axis =1)
y = patient_data['Outcome']
X_train, X_test, Y_train, Y_test = train_test_split(x,y,test_size=0.2)
```

In [17]:
```
model = LogisticRegression(max_iter=1000)
model.fit(X_train, Y_train)
```

Out[17]:
```
LogisticRegression(max_iter=1000)
```

In [18]:
```
# Predicting the model
prediction = model.predict(X_test)
print(prediction)
```

```
[0 0 0 0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1 0 0
 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
 0 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
 0 0 0 1 0 0]
```

In [19]:
```
# Checking the accuracy
accuracy = accuracy_score(prediction,Y_test)
print(accuracy)
```

```
0.7207792207792207
```

In [20]:
```
# We can say that our predictive model created is 81.16% accurate
```