Data Science Capstone Project-Healthcare

Problem Statement: *NIDDK (National Institute of Diabetes and Digestive and Kidney Diseases) research creates knowledge about and treatments for the most chronic, costly, and consequential diseases.

*The dataset used in this project is originally from NIDDK. The objective is to predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. *Build a model to accurately predict whether the patients in the dataset have diabetes or not.

Dataset Description The datasets consists of several medical predictor variables and one target variable (Outcome). Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and more.

Variables - Description

- Pregnancies - Number of times pregnant
- Glucose - Plasma glucose concentration in an oral glucose tolerance test
- BloodPressure - Diastolic blood pressure (mm Hg)
- SkinThickness - Triceps skinfold thickness (mm)
- Insulin - Two hour serum insulin
- BMI - Body Mass Index
- DiabetesPedigreeFunction - Diabetes pedigree function
- Age - Age in years
- Outcome - Class variable (either 0 or 1). 268 of 768 values are 1, and the others are 0

Project Task: Week 1:

Data Exploration:

1. Perform descriptive analysis. Understand the variables and their corresponding values. On the columns below, a value of zero does not make sense and thus indicates missing value:

    • Glucose

    • Blood Pressure

    • Skin Thickness

    • Insulin

    • BMI

```
data = pd.read_csv('health care diabetes.csv')
```

Data Preprocessing (EDA)

1. Perform descriptive analysis. Understand the variables and their corresponding values. On the columns below, a value of zero does not make sense and thus indicates missing value: ¶

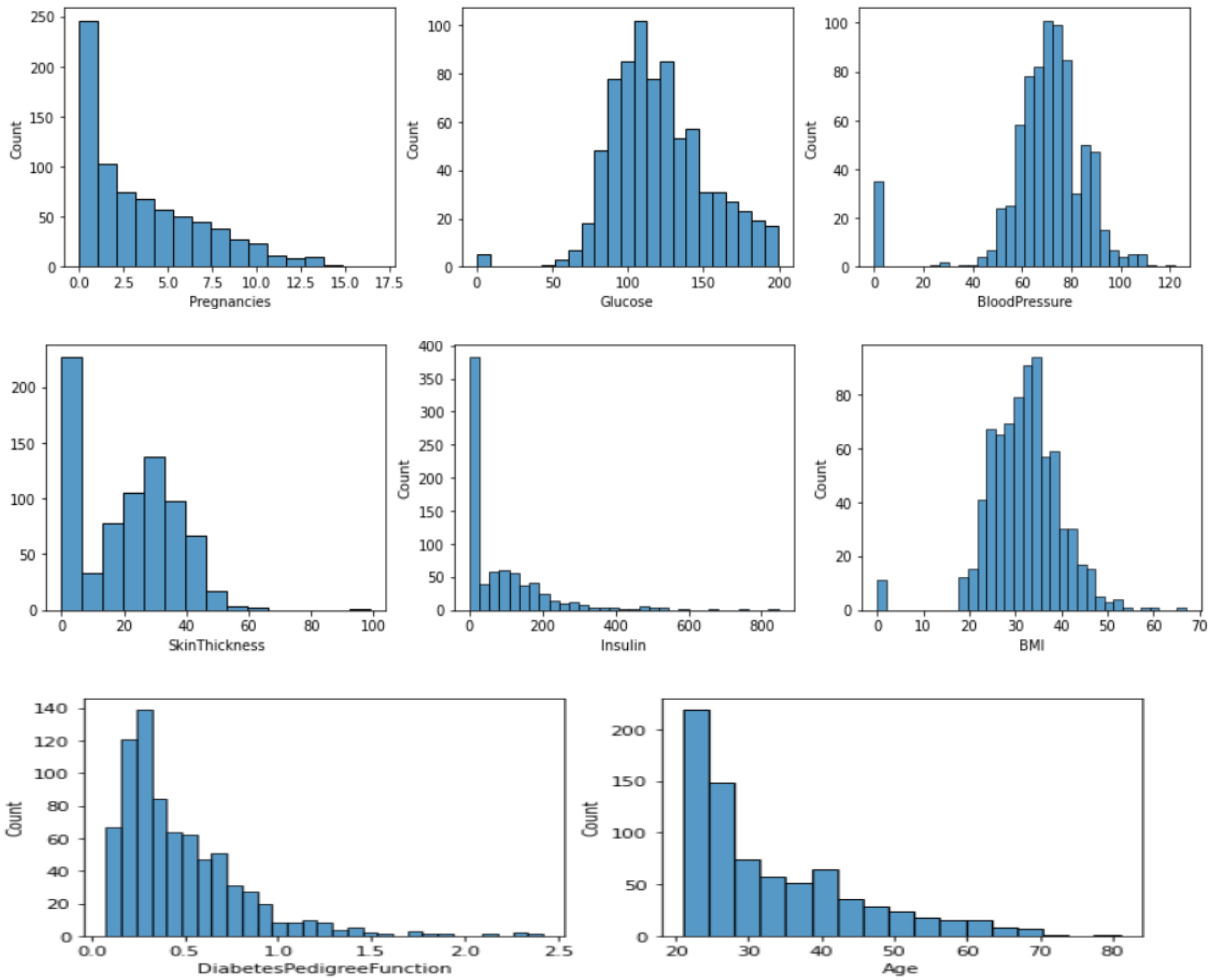
[4]:
```
data.head()
```

[4]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

2. Visually explore these variables using histograms. Treat the missing values accordingly.

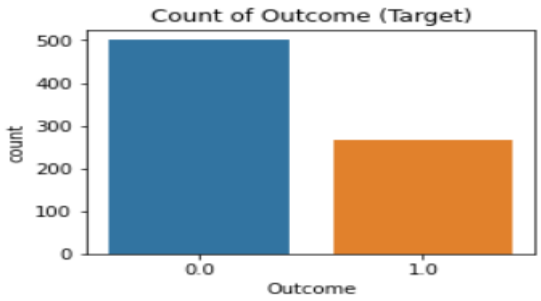
```
#Checking distribution of variables
plt.figure(figsize=(15,12))
for i in range(8):
    plt.subplot(3,3,i+1)
    sns.histplot(data.iloc[:,i])
```



3. There are integer and float data type variables in this dataset. Create a count (frequency) plot describing the data types and the count of variables.

```
[19]: data.dtypes.value_counts().plot(kind='bar', figsize=(5,3),title = 'Data types Count Plot\n', xlabel='Data types', ylabel= 'Count');
plt.show()
```

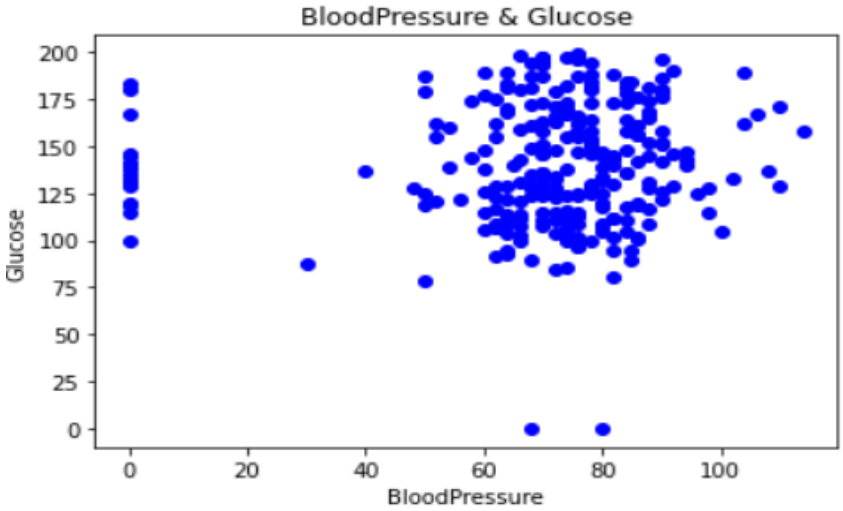
4. Check the balance of the data by plotting the count of outcomes by their value. Describe your findings and plan future course of action.



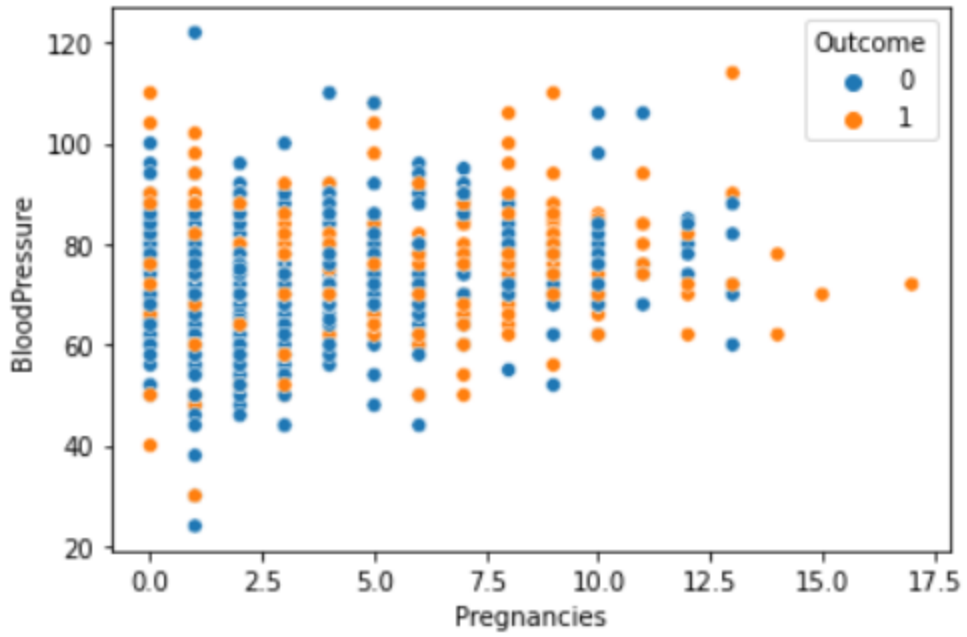
Observations:

Approximately 35% of patients in the dataset have diabetes, while 65% are non-diabetic.

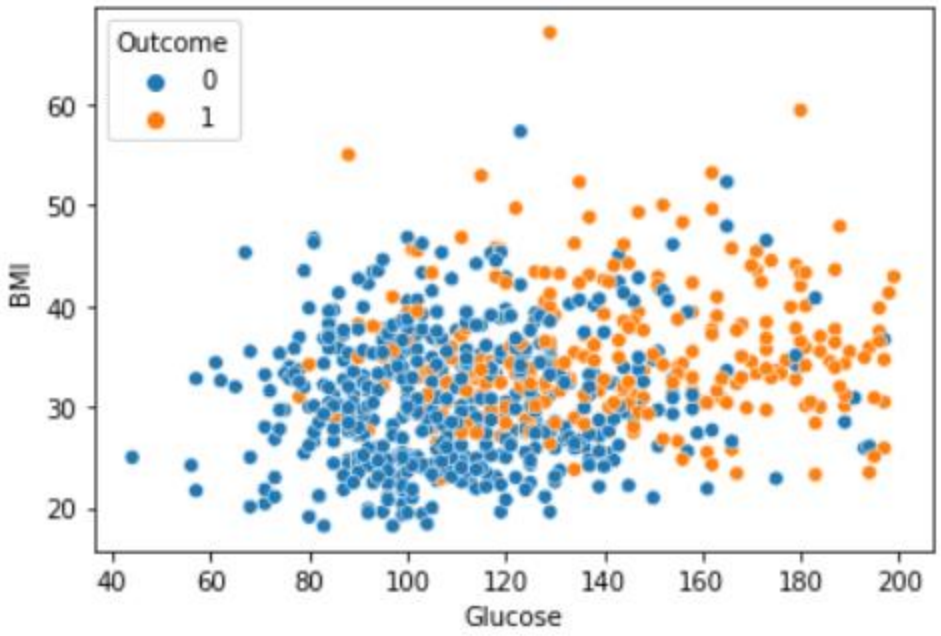
5. Create scatter charts between the pair of variables to understand the relationships.



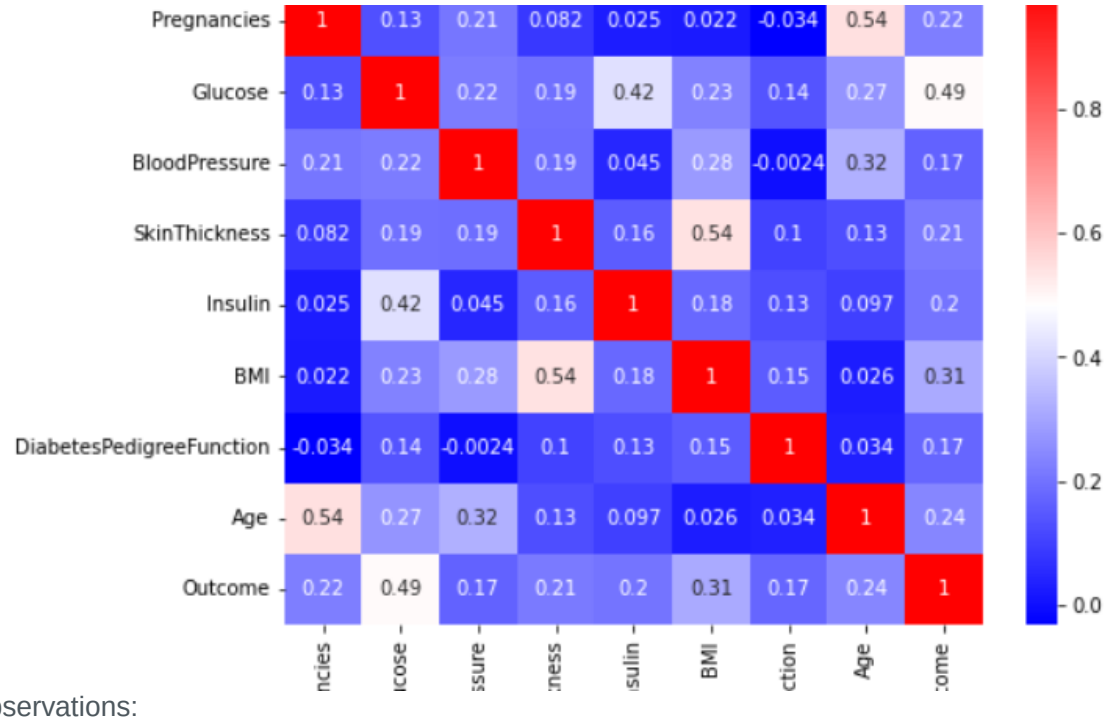
```
S =sns.scatterplot(x= "Pregnancies" ,y= "BloodPressure",
                   hue="Outcome",
                   data=data);
```




```
G =sns.scatterplot(x= "Glucose" ,y= "BMI",
                   hue="Outcome",
                   data=data);
```



6. Perform correlation analysis. Visually explore it using a heat map.



Observations:

Our analysis of the Scatter plot and Heatmap found some noteworthy correlations worth mentioning. "BMI" and "Skin Thickness" are positively correlated, as are "Age" and "Pregnancies," with "Age" also slightly correlated to "Blood Pressure." "Insulin" and "Glucose" are moderately correlated, but "Outcome" shows the most significant correlation with "Glucose." There is also a slight correlation between "Outcome" and "BMI." No pairs of variables have negative correlations.

Project Task: Week 2

Data Modeling:

1. Devise strategies for model building. It is important to decide the right validation framework. Express your thought process.

To build a reliable machine learning classification model for a Diabetes dataset, we need to evaluate its performance using appropriate validation techniques. We can split the dataset into training and testing sets or use cross-validation technique, which partitions the data into subsets and tests the model on each of them. We can also use grid search to find the optimal set of hyperparameters for the best performance.

Evaluating the model's performance using parameters such as sensitivity, specificity, and AUC (ROC curve) can help us identify its accuracy in predicting positive and negative cases. Sensitivity measures the proportion of true positive cases to actual positive cases, specificity measures the proportion of true negative cases to actual negative cases, and AUC measures the model's ability to distinguish between positive and negative cases. By calculating these parameters, we can make informed decisions on how to improve our model's performance.

2. Apply an appropriate classification algorithm to build a model.

LogisticRegression

[63]:
```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
```

[64]:
```
model.fit(x_train_norm,y_train_norm)
```

[64]:
```
▾ LogisticRegression
LogisticRegression()
```

[65]:
```
y_pred = model.predict(x_test_norm)
y_pred
```

[65]:
```
array([1., 0., 0., 1., 0., 0., 1., 1., 0., 0., 1., 1., 0., 0., 0., 0., 1.,
       0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0.,
       0., 1., 0., 0., 0., 1., 0., 0., 0., 1., 1., 0., 0., 0., 0., 0., 0.,
       0., 1., 0., 0., 0., 0., 1., 0., 0., 1., 1., 0., 0., 1., 1., 1., 0.,
       0., 0., 0., 0., 0., 1., 1., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 1., 0., 0., 0., 0., 0., 1., 0., 0., 1., 1., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 1., 0., 1., 1., 0., 1., 0.,
       1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0.,
       0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0.,
       0.])
```

```
y_pred = model.predict(x_test_norm)
y_pred
```

```
array([1., 0., 0., 1., 0., 0., 1., 1., 0., 0., 1., 1., 0., 0., 0., 0., 1.,
       0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0.,
       0., 1., 0., 0., 0., 1., 0., 0., 0., 1., 1., 0., 0., 0., 0., 0., 0.,
       0., 1., 0., 0., 0., 0., 1., 0., 0., 1., 1., 0., 0., 1., 1., 1., 0.,
       0., 0., 0., 0., 0., 1., 1., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 1., 0., 0., 0., 0., 0., 1., 0., 0., 1., 1., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 1., 0., 1., 1., 0., 1., 0.,
       1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0.,
       0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0.,
       0.])
```

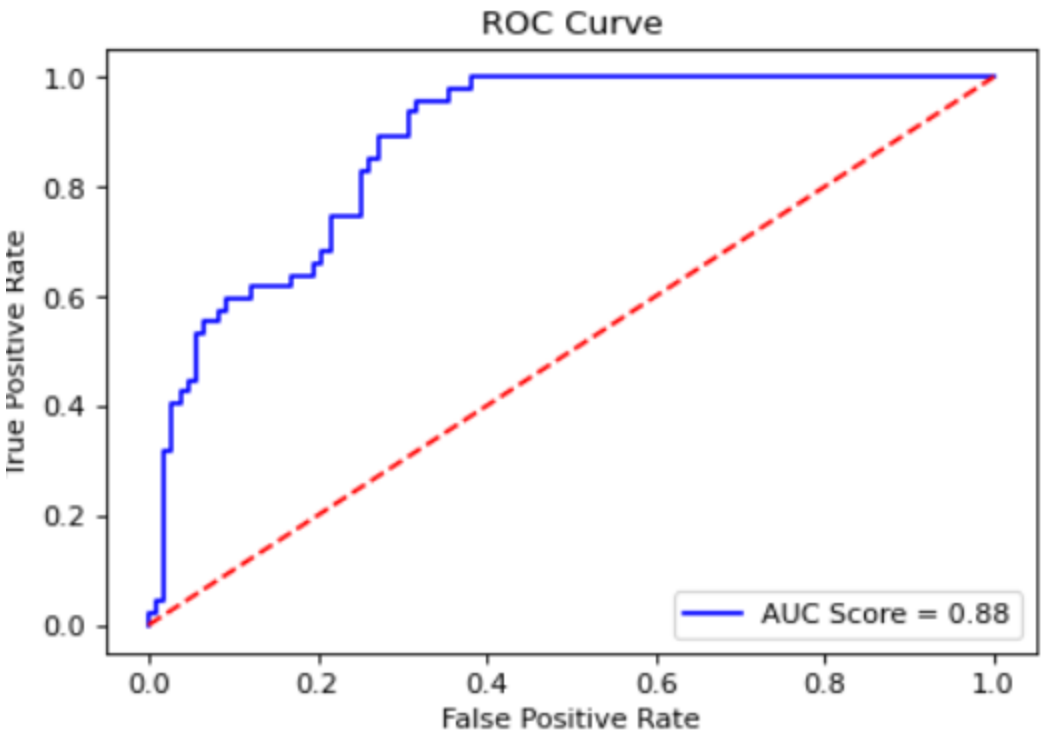
```
from sklearn import metrics
matrix = metrics.confusion_matrix(y_test_norm, model.predict(x_test_norm))
print(matrix)
```

```
[[98  9]
 [20 27]]
```

```
Model Validation ==>

Accuracy Score of Logistic Regression Model::
0.8116883116883117

 Classification Report::
              precision    recall  f1-score   support

         0.0       0.83      0.92      0.87       107
         1.0       0.75      0.57      0.65        47

    accuracy                           0.81       154
   macro avg       0.79      0.75      0.76       154
weighted avg       0.81      0.81      0.80       154
```

ROC Curve

```
<matplotlib.legend.Legend at 0x7fd29582f220>
```

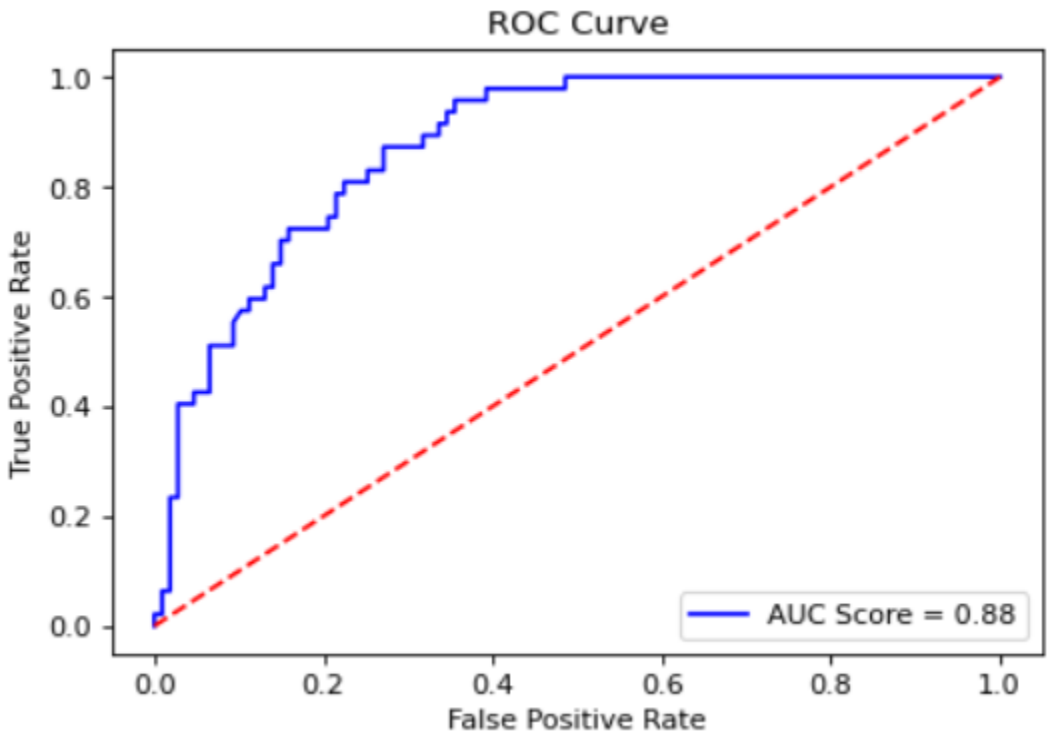


3. Compare various models with the results from KNN algorithm.

```
Model Validation ==>

Accuracy Score of KNN Model::
0.8181818181818182

 Classification Report::
              precision    recall  f1-score   support

         0.0       0.85      0.90      0.87       107
         1.0       0.73      0.64      0.68        47

    accuracy                           0.82       154
   macro avg       0.79      0.77      0.78       154
weighted avg       0.81      0.82      0.81       154


 Model Validation ==>

 Accuracy Score of Logistic Regression Model::
 0.8116883116883117

  Classification Report::
              precision    recall  f1-score   support

         0.0       0.83      0.92      0.87       107
         1.0       0.75      0.57      0.65        47

    accuracy                           0.81       154
   macro avg       0.79      0.75      0.76       154
weighted avg       0.81      0.81      0.80       154

 Model Validation ==>

 Accuracy Score of Decision Tree Model::
 0.7467532467532467

  Classification Report::
              precision    recall  f1-score   support

         0.0       0.83      0.79      0.81       107
         1.0       0.58      0.64      0.61        47

    accuracy                           0.75       154
   macro avg       0.71      0.72      0.71       154
weighted avg       0.76      0.75      0.75       154
```

```
Model Validation ==>

Accuracy Score of Random Forest Model::
0.7987012987012987

 Classification Report::
               precision    recall  f1-score   support

          0.0       0.85      0.87      0.86       107
          1.0       0.68      0.64      0.66        47

     accuracy                           0.80       154
    macro avg       0.76      0.75      0.76       154
 weighted avg       0.80      0.80      0.80       154
```

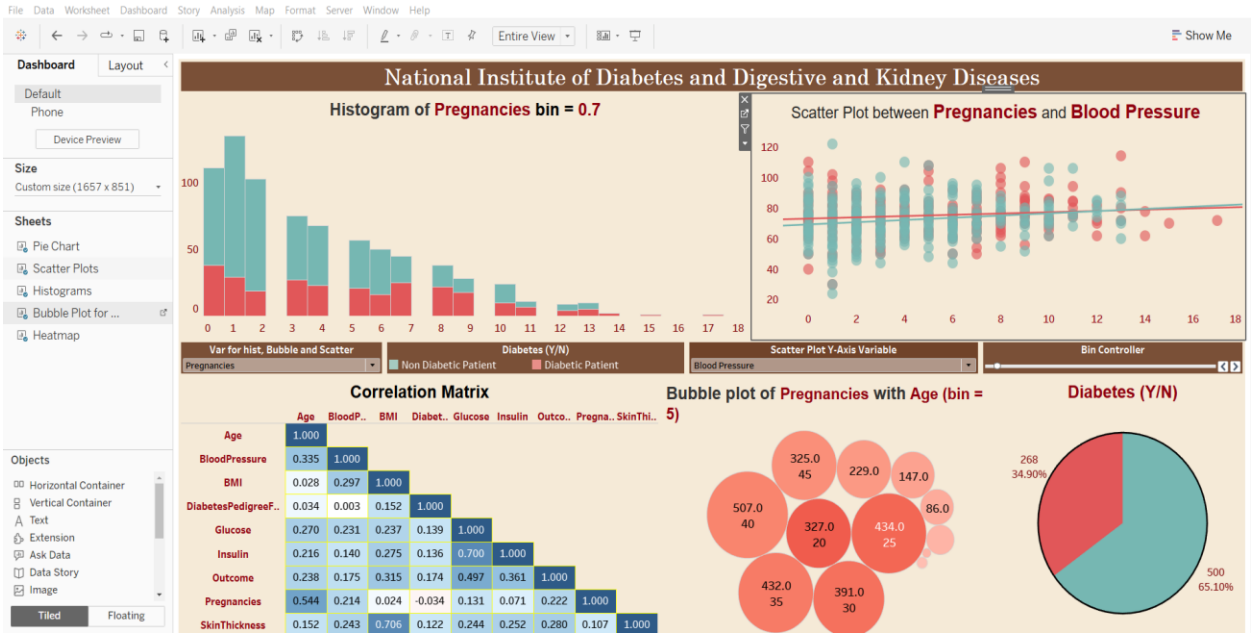
4. Create a classification report by analysing sensitivity, specificity, AUC (ROC curve), etc. Please be descriptive to explain what values of these parameter you have used.



5. Create a dashboard in tableau by choosing appropriate chart types and metrics useful for the business. The dashboard must entail the following:

- Pie chart to describe the diabetic or non-diabetic population
- Scatter charts between relevant variables to analyze the relationships
- Histogram or frequency charts to analyze the distribution of the data
- Heatmap of correlation analysis among the relevant variables

- Create bins of these age values: 20-25, 25-30, 30-35, etc. Analyze different variables for these age brackets using a bubble chart.



https://public.tableau.com/app/profile/bhavana.rahangdale/viz/DSCapstoneproject-Healthcare/Dashboard-Diabetes