**Report created by : Kaushik Prasad Dey**

**Topic :**NIDDK (National Institute of Diabetes and Digestive and Kidney Diseases) – research for the most chronic, costly, and consequential diseases and predict whether a patient has diabetes, based on certain diagnostic measurements.



**Description**

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 |

**Tools Used For this Project**

Icon

Description automatically generated A picture containing chart

Description automatically generated A picture containing text, clipart

Description automatically generated Chart, radar chart

Description automatically generated

**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:

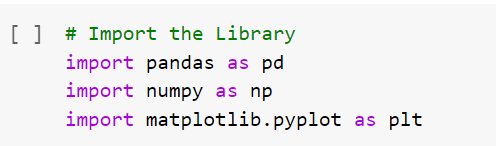
* Glucose
* BloodPressure
* Skin Thickness
* Insulin
* BMI

1. Visually explore these variables using histograms. Treat the missing values accordingly.
2. 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.
3. Check the balance of the data by plotting the count of outcomes by their value. Describe your findings and plan future course of action.
4. Create scatter charts between the pair of variables to understand the relationships. Describe your findings.
5. Perform correlation analysis. Visually explore it using a heat map.

**Methodology for Task for Week -1**

**Note: All Python code are written inside capstone.ipynb file. Attached this file set along with the report folder.**

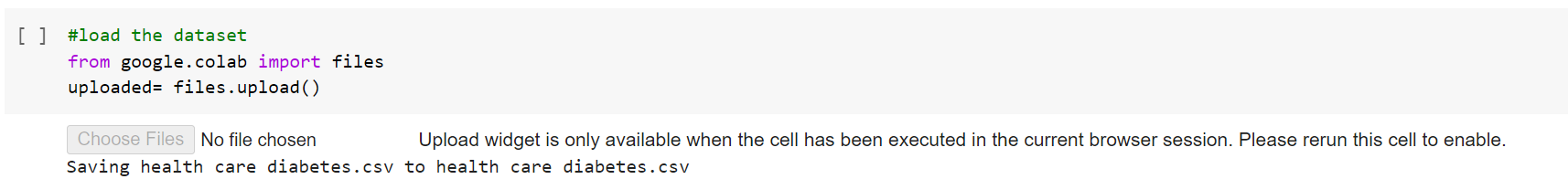
1. **Import** all necessary library for this project. The Screenshot is given below.



1. Next, we have to read the required csv file. The screen-shot is given below.

Note:

* 1. As using in google-colab note-book for this project, two extra code I have to add as a screen-shot.
  2. Data set name is **“health care diabetes.csv”**.



Graphical user interface, text

Description automatically generated with medium confidence

1. After reading csv file some following steps we have to perform for more details knowledge of this csv data set.

**3.1** Read first and last five rows from the dataset.

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated

**3.2** Now, find out the shape, five point summary, statistical information & how many null values are present inside this data-set.

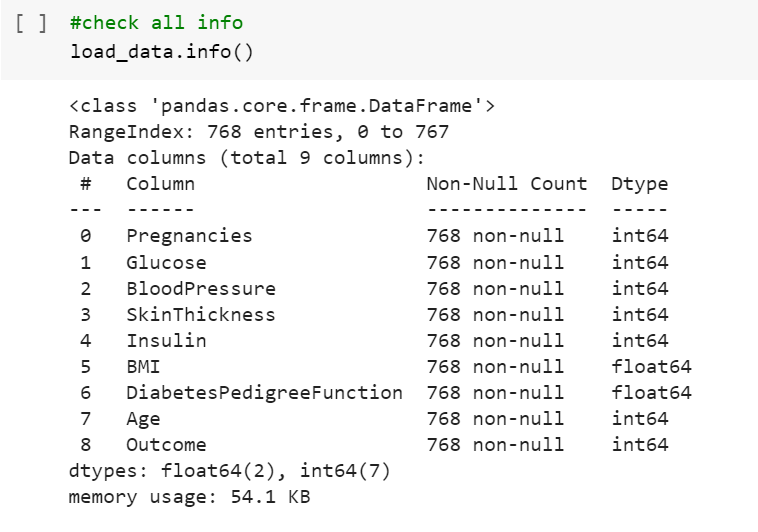
**Shape: It helps to find out rows & columns from the data set.**

Graphical user interface, text, application

Description automatically generated

So, we got to know that we have **768 rows and 9 columns** in that dataset.

**Info: It** prints information about the DataFrame. The information contains the number of columns, column labels, column data types, memory usage, range index, and the number of cells in each column (non-null values). Note: the info() method actually prints the info.



**Describe: It** returns description of the data in the DataFrame. If the DataFrame contains numerical data, the description contains these information for each column: count - The number of not-empty values. mean - The average (mean) value. std - The standard deviation.

Table

Description automatically generated

**Isnull & sum: It** returns how many null values are present inside this dataset & its total count.

Table

Description automatically generated

**Perform Descriptive Analysis**

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

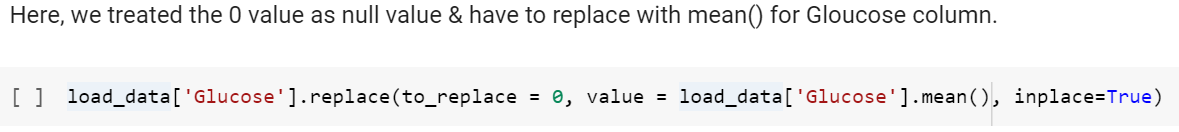
**Case for Variable Glucose**

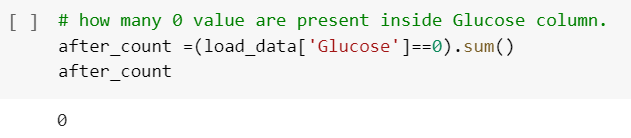
**Before treating missing value( in my case 0) .**

Text

Description automatically generated with medium confidence

**Handle the missing value( in my case 0)**





**If, we compare the Histogram before & after the missing value treatment, it shows like that**

Chart, histogram

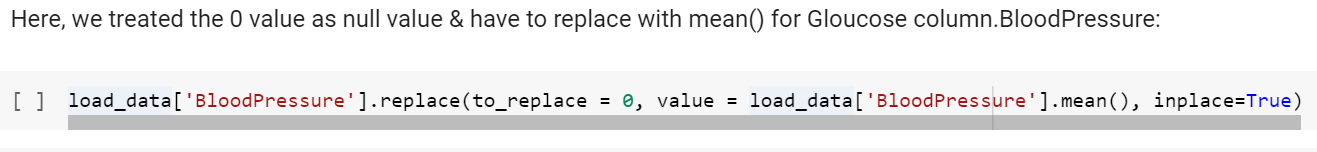
Description automatically generatedChart, histogram

Description automatically generated **Before After**

**Case for Variable BloodPressure**

**Before treating missing value( in my case 0)**

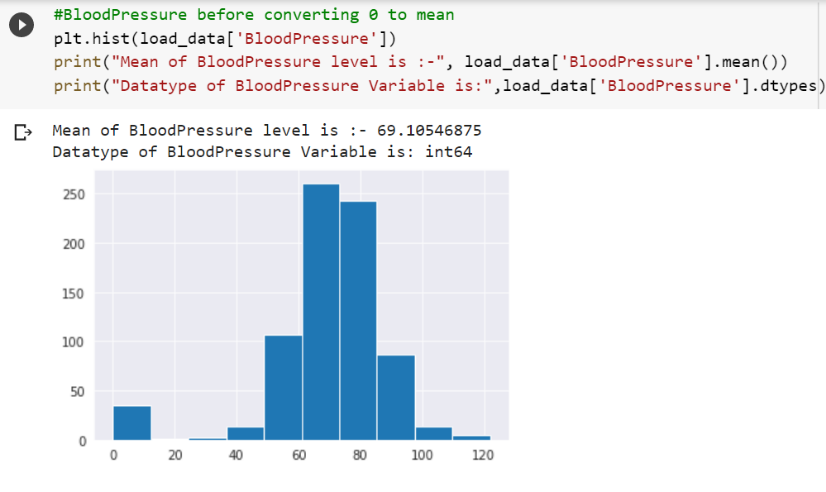
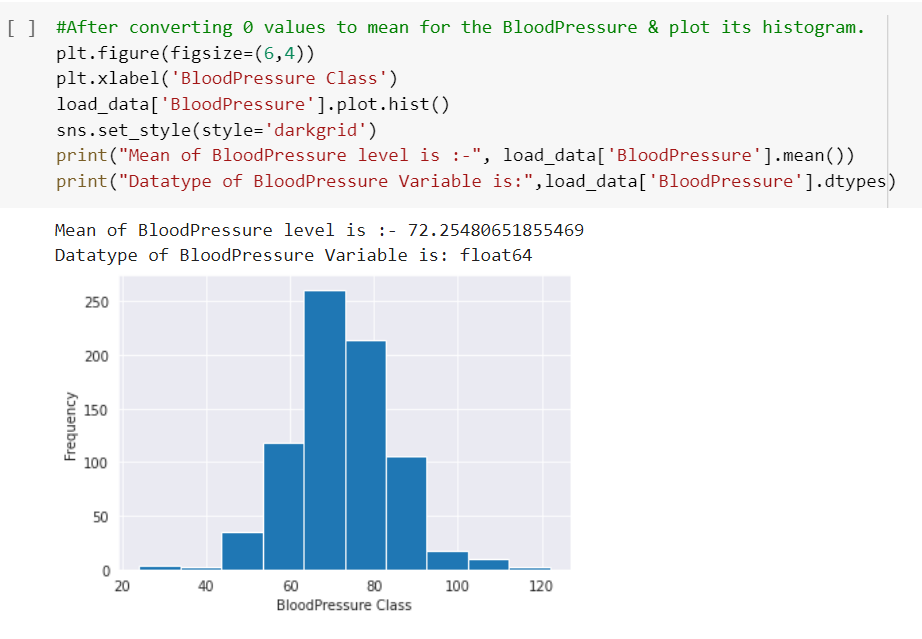


**Handle the missing value ( in my case 0)**

A picture containing text

Description automatically generated

**If, we compare the Histogram before & after the missing value treatment, it shows like that**

**Before After**

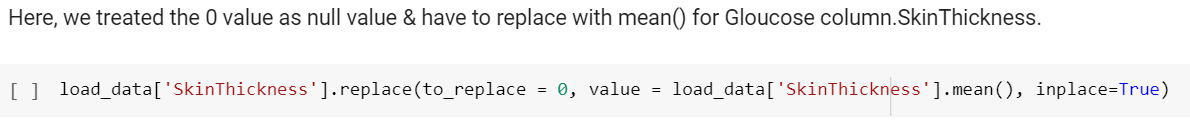
**Case for Variable Skin-Thickness**

**Before treating missing value ( in my case 0 )**

A picture containing text

Description automatically generated

**Handle the missing value ( in my case 0)**



Text

Description automatically generated with medium confidence

**If, we compare the Histogram before & after the missing value treatment, it shows like that**

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated**Before After**

**Case for Variable Insulin**

**Before treating missing value ( in my case 0 )**

Text

Description automatically generated with low confidence

**Handle the missing value ( in my case 0)**

A picture containing text

Description automatically generated

Graphical user interface, text

Description automatically generated with medium confidence

**If, we compare the Histogram before & after the missing value treatment, it shows like that**

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated**Before After**

**Case for Variable BMI**

**Before treating missing value ( in my case 0 )**

A picture containing graphical user interface

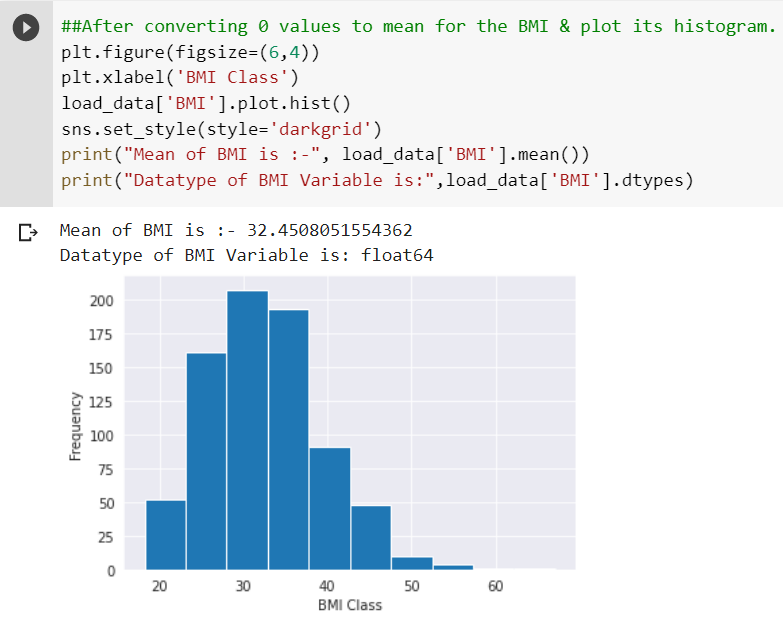
Description automatically generated

**Handle the missing value ( in my case 0)**

A picture containing text

Description automatically generated

**If, we compare the Histogram before & after the missing value treatment, it shows like that.**

Chart, histogram

Description automatically generated**Before After**

**So, finally all histogram are looks like in below**

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated**Histogram for Glucose Histogram for Blood-Pressure**

Chart, histogram

Description automatically generatedChart, histogram

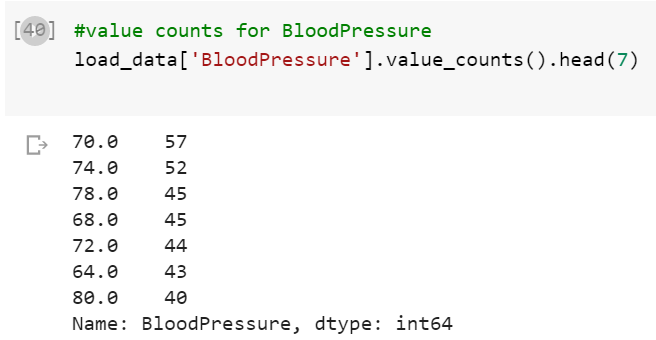
Description automatically generated**Histogram for Skin-ThickNess Histogram for Insulin**

**Histogram for BMI**

Chart, histogram

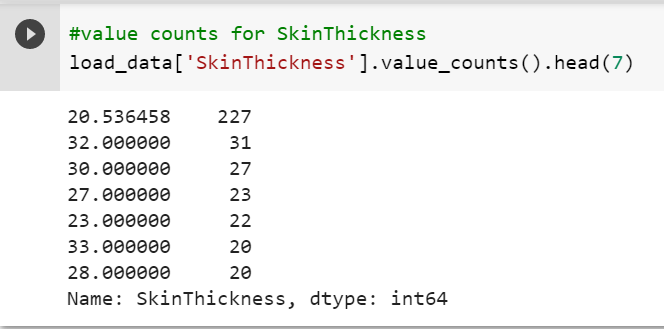
Description automatically generated

**Create a count(frequency) plot describing the data types & the count of variables**

Text

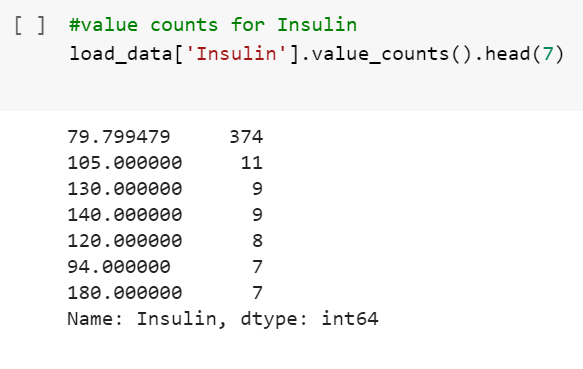
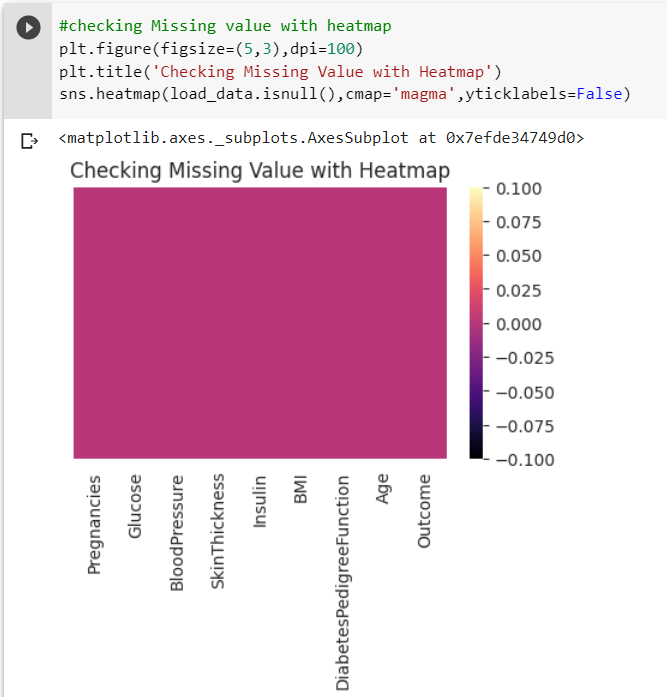
Description automatically generated with medium confidence**Count Table for Variable Glucose Count Table for Variable Blood Pressure**

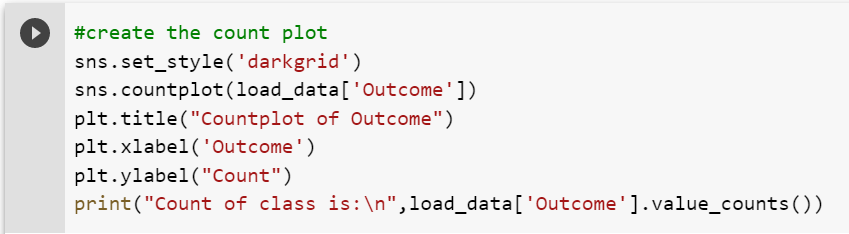
**Count Table for Variable BMI Count Table for Variable SkinThickness**

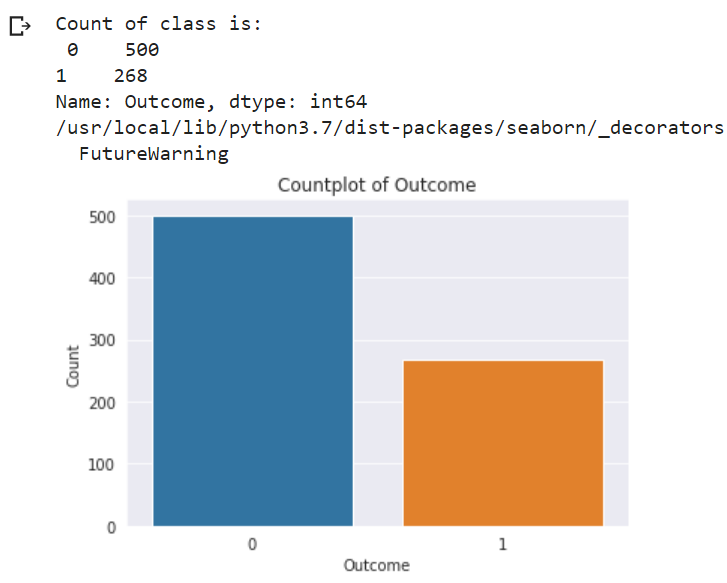
Table

Description automatically generated

**Count Table for Insulin Checking Missing Values with Heat-Map**



**Plotting the count of outcomes by their values.**



**Describe the finding**

We can see that both class is balanced so we need not to perform any sampling method to maintain the balance between both classes. Therefor directly using this data in training and testing purpose without performing any sampling method. Meanwhile during Model Validation , we also need not worry about ROC Curve because data is not imbalanced, but as this is a medical data so I will be using ROC curve to make sure TYPE 2 ERROR will not be there.

**Scatter plot**

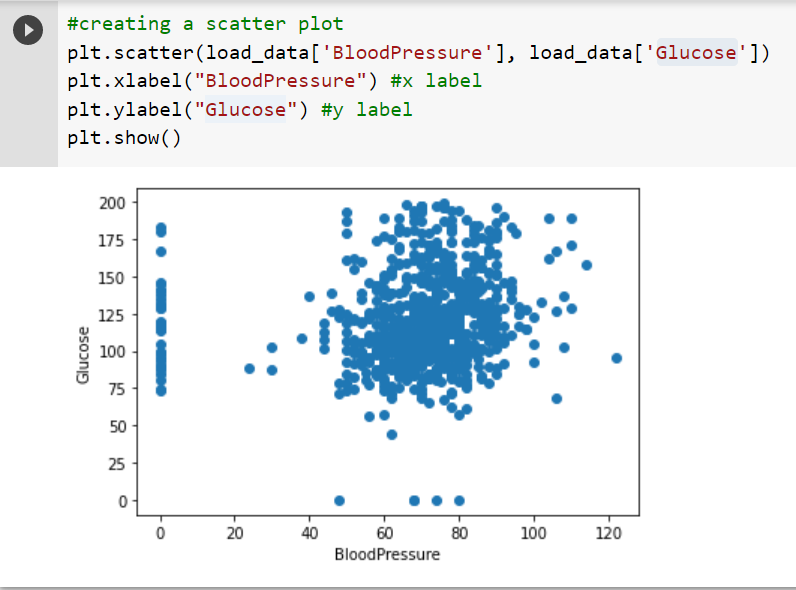
Create scatter charts between the pair of variables to understand the relationships. Describe your findings.



**Describe the observation**

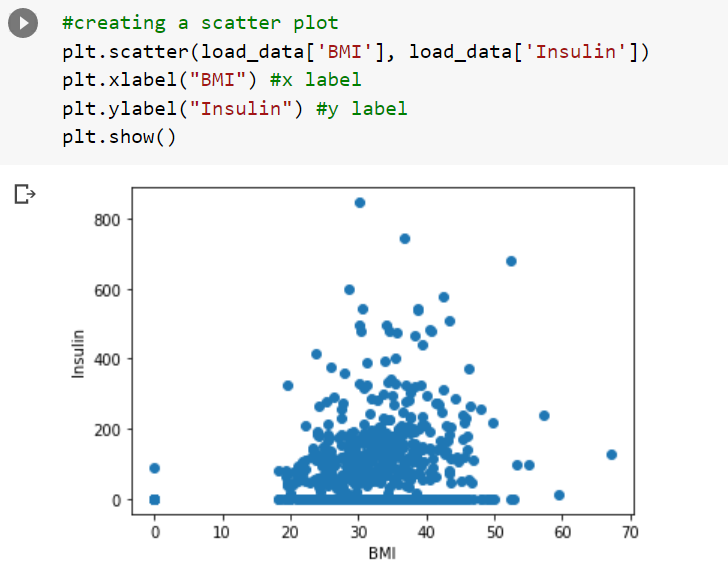
**We can see from scatter plot that there is no strong multicolinearity among features, but between skin thickness and BMI, Pregnancies and age it looks like there is small chance of positive correlation.**

**Also create scatter plot to identify the co relation between variables observation.**

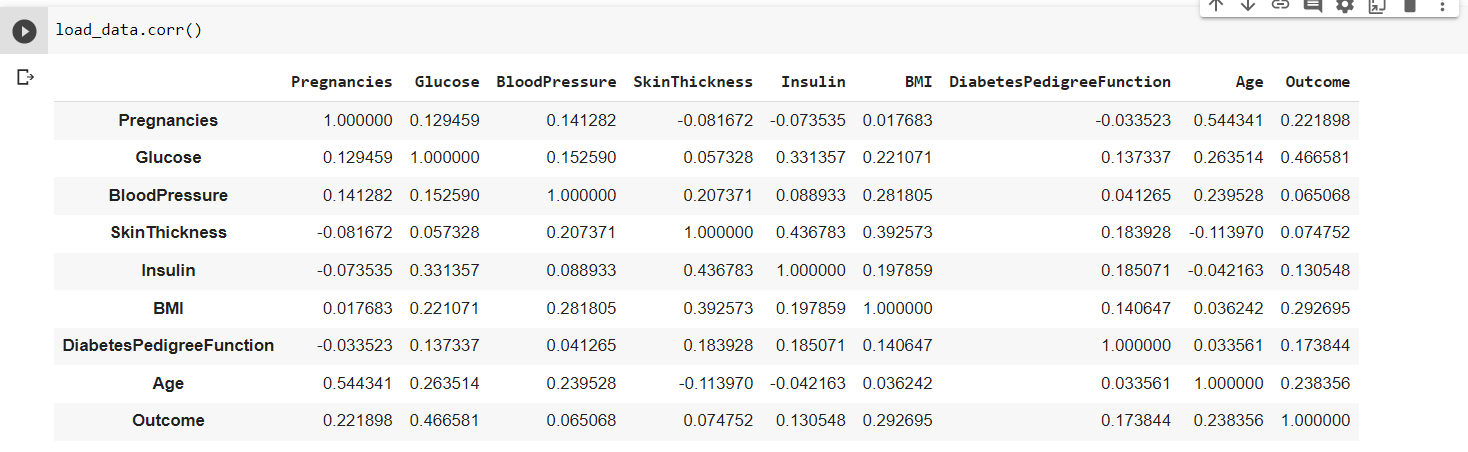
**Scatter plot with Glucose and Blood-Pressure**

**So, from this observation we can seen that the positive (+) co-relation is happened between BloodPressure & Glucose.**

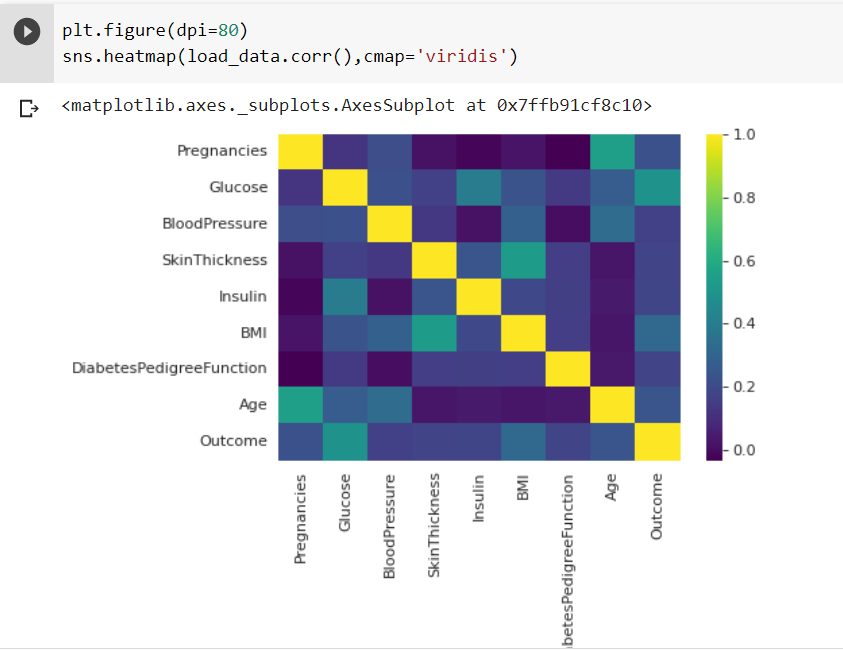
**Scatter plot with BMI and Insulin**

**So, from this observation we can seen that the positive (+) co-relation is happened between BMI & Insulin.**

**Perform Co-relation Analysis with heat map**



Timeline

Description automatically generated with medium confidence**Heat map with Theme( YlGnBu ) Heat map with Theme( virdis )**

**Project Task: Week 2**

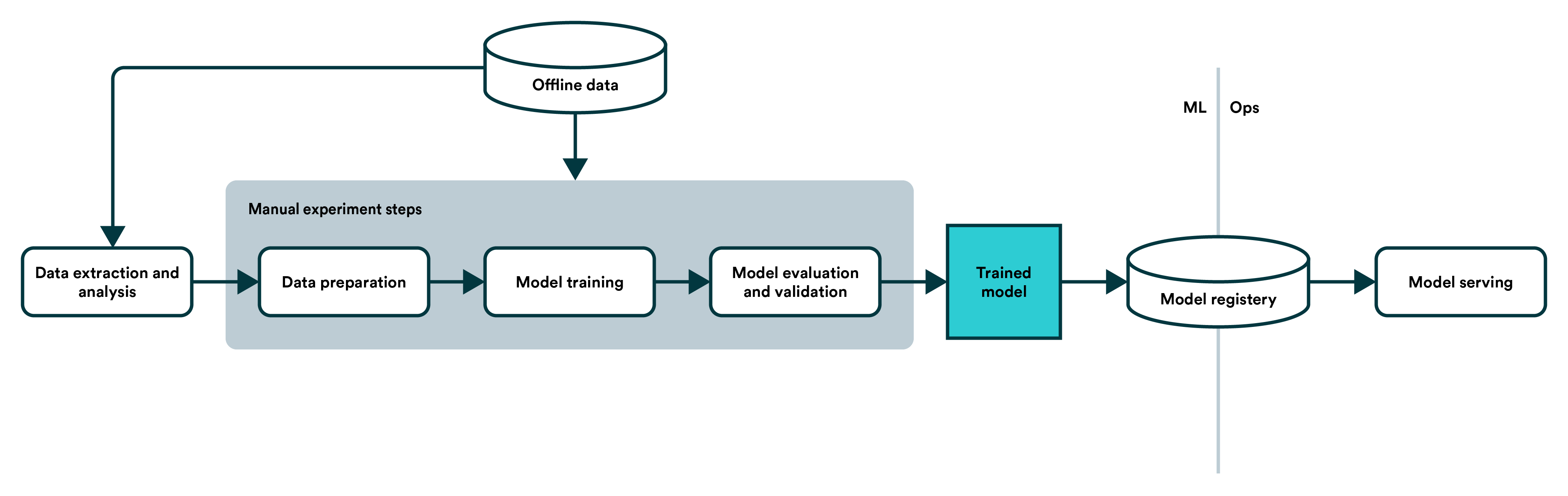
**Data Modeling**

1. Devise strategies for model building. It is important to decide the right validation framework. Express your thought process.
2. Apply an appropriate classification algorithm to build a model.
3. Compare various models with the results from KNN algorithm.
4. Create a classification report by analyzing sensitivity, specificity, AUC (ROC curve), etc.

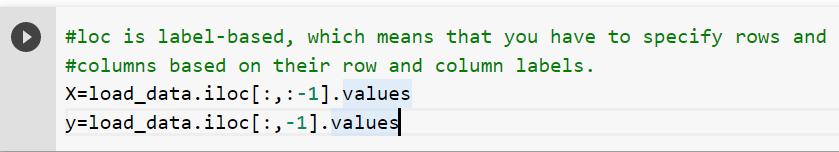
Please be descriptive to explain what values of these parameter you have used.

**Methodology for Task for Week -2**

**Note: All Python code are written inside capstone.ipynb file. Attached this file set along with the report folder.**

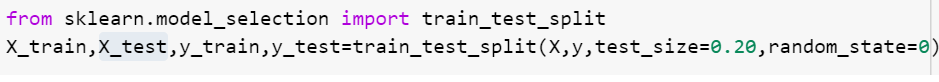
**So, from the below data pipeline we have to reach that training model stage but currently we are in data processing stage.**

**Step -1 :**

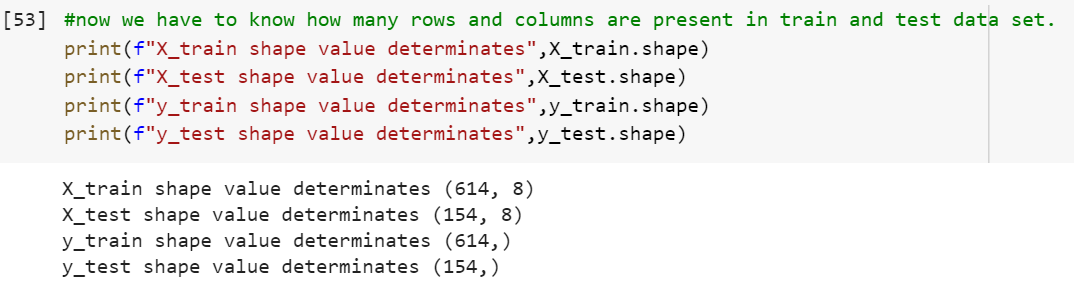


**Step -2 ( Splitting the data-set into training and testing part )**

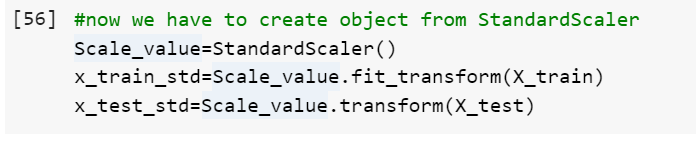
now we have import train test split from sklearn. model\_selection. now split our dataset into 80% training and 20% testing & random state is treated as hyperparameter which control the shuffing process. With random\_state=0 , we get the same train and test sets across different executions.

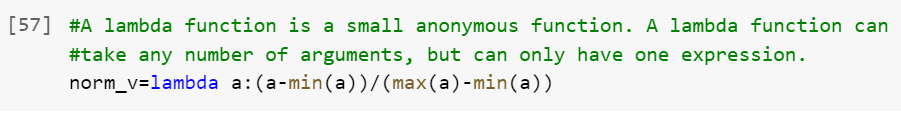


**Step -3 ( To identify how many rows and columns are present in X\_train, X\_test, y\_train and y\_test.**



**Step -4 : Now we have to import StandardScaler which** use to standardize the data values into a standard format



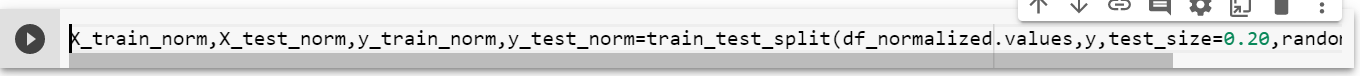


**Step -5 : we again validated with the shape value of X\_train\_form, X\_test\_form, y\_train\_form & y\_test\_form.**

A picture containing logo

Description automatically generated

Text

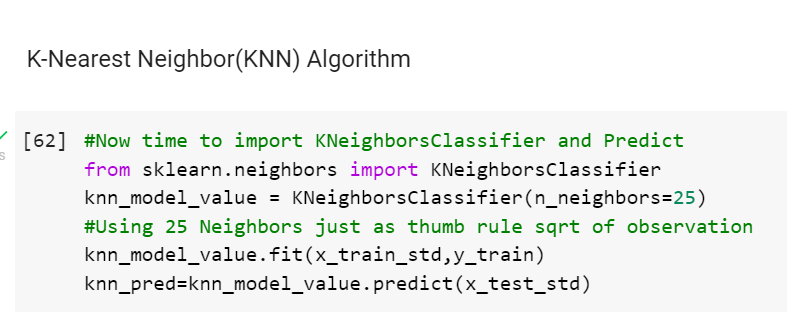
Description automatically generated

**Now Starting Different Model Building Process (KNN)**

**KNN (KNN with Standard Scaling)**

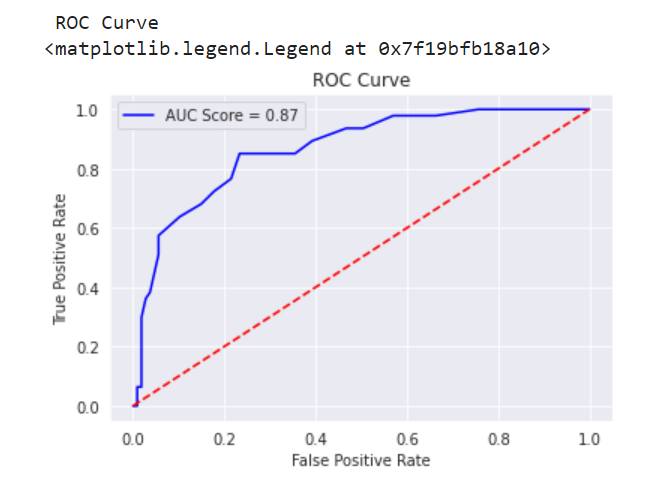
Text

Description automatically generatedCalendar

Description automatically generated with low confidence**K-NN algorithm** stores all the available data and classifies a new data point based on the similarity.

Table

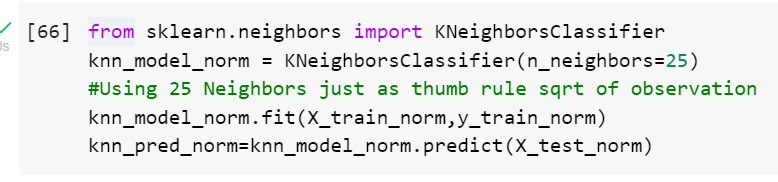
Description automatically generated

**OBSERVATION**

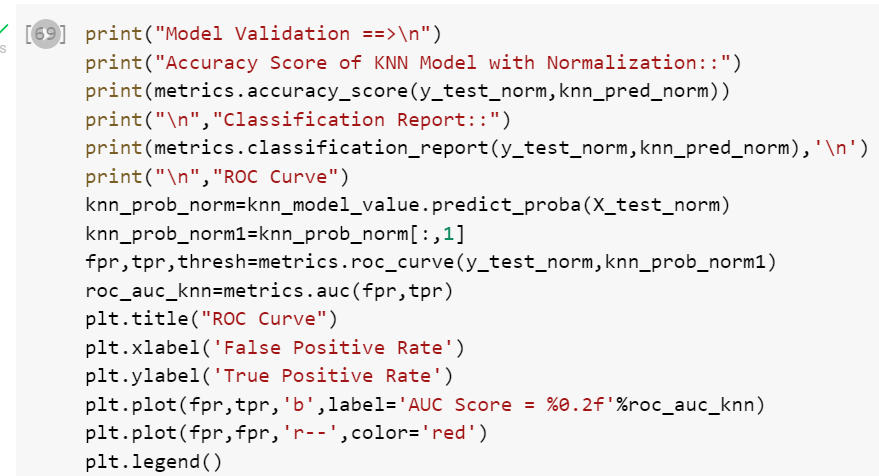
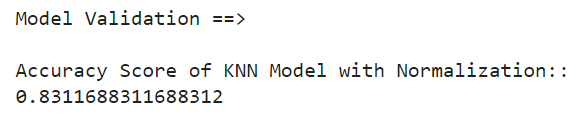
**SO, from KNN standard scaling we can got the Model Accuracy is 0.81 and from the classification report we got its Precision, recall, f1-scoare and support value. With Addition its ROC Curve AUC score is 87%. That is good.**

**KNN-Normalization**

Calendar

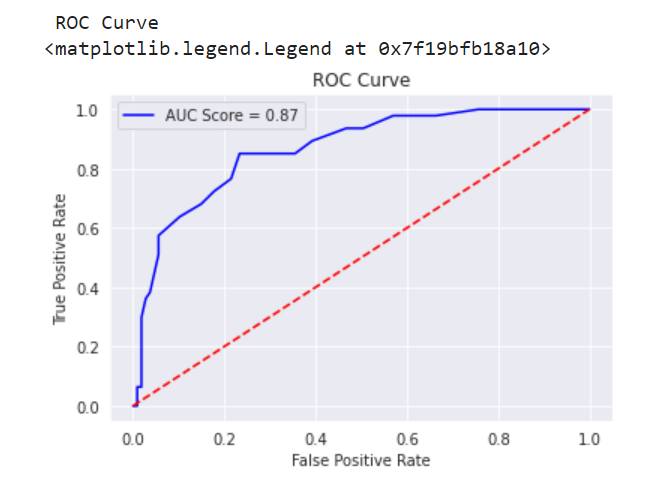
Description automatically generated

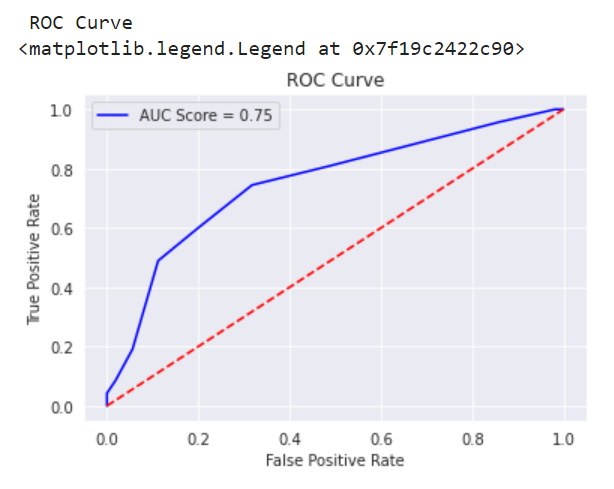
Table

Description automatically generated

**OBSERVATION**

**we compare, the KNN-Standardization and Normalization from the below ROC curve.**

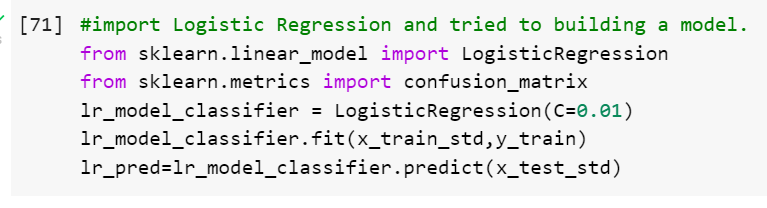
**KNN- Standardization KNN- Normalization**



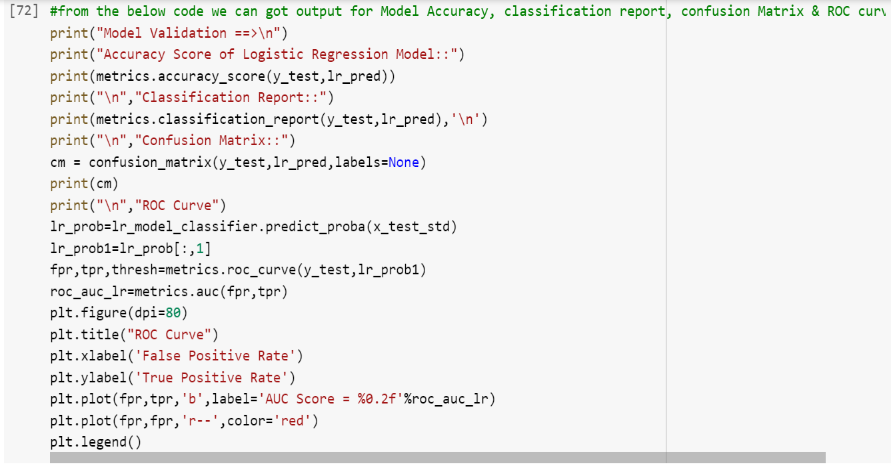
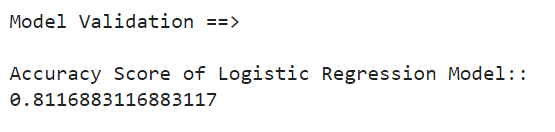
**We can clearly see that KNN with Standardization is better than Normalization.**

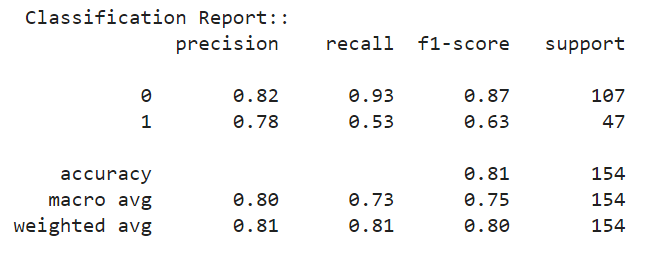
**Now Starting Different Model Building Process (LOGSTIC REGRESSION)**

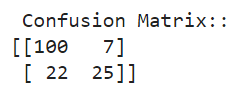
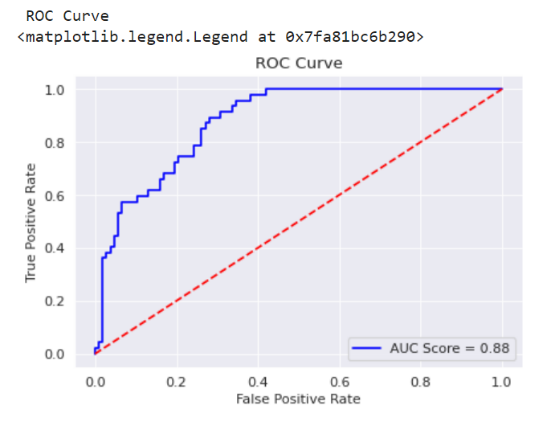
Logistic regression is an example of **supervised learning**. It is used to calculate or predict the probability of a binary (yes/no) event occurring.

**Model Creation:**

**From the below code we can got output for Model Accuracy, classification report, confusion Matrix & ROC curve:**

**Model Accuracy :**

 **Classification Report:**

**Confusion Matrix: ROC CURVE:**

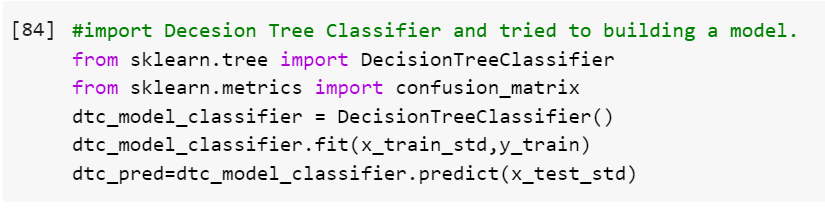
**Observation:**

Accuracy of KNN is better than Logistic Regression,but auc score of Logistic regression is better.

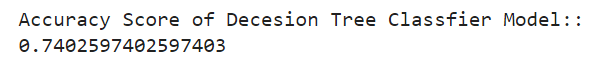
**Now Starting Different Model Building Process (Decision Tree Classifier)**

Decision Tree is **also a Supervised Machine Learning Algorithm that uses a set of rules to make decisions**, similarly to how humans make decisions. One way to think of a Machine Learning classification algorithm is that it is built to make decisions.

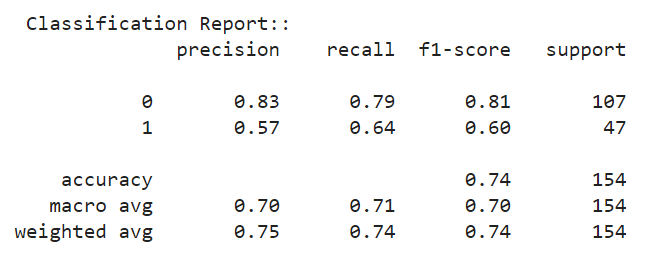
**Model Creation:**

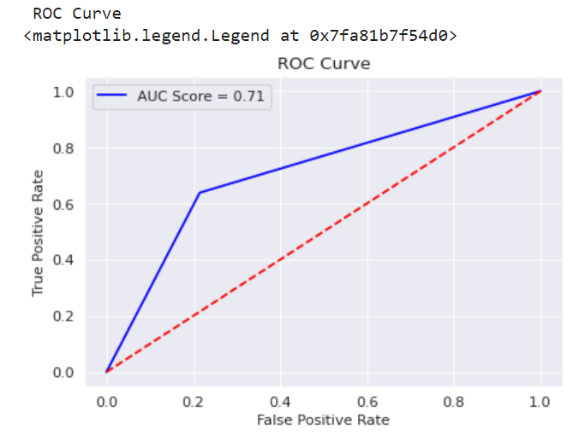


**From the below code we can got output for Model Accuracy, classification report, confusion Matrix & ROC curve:**

Text

Description automatically generated **Model Accuracy :**

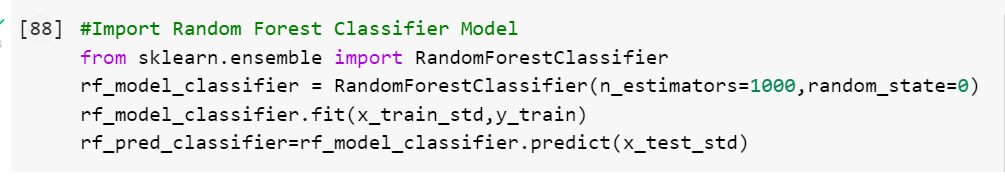
 **Classification Report :**

**ROC CURVE :**

**Now Starting Different Model Building Process (Random Forest Classifier)**

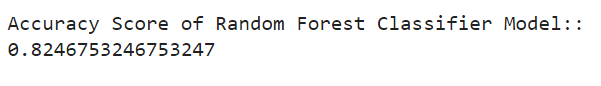
A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

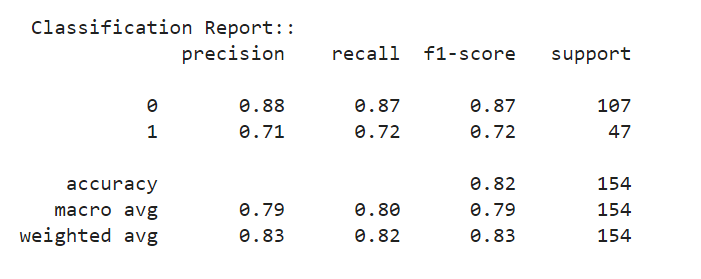
**Model Creation:**

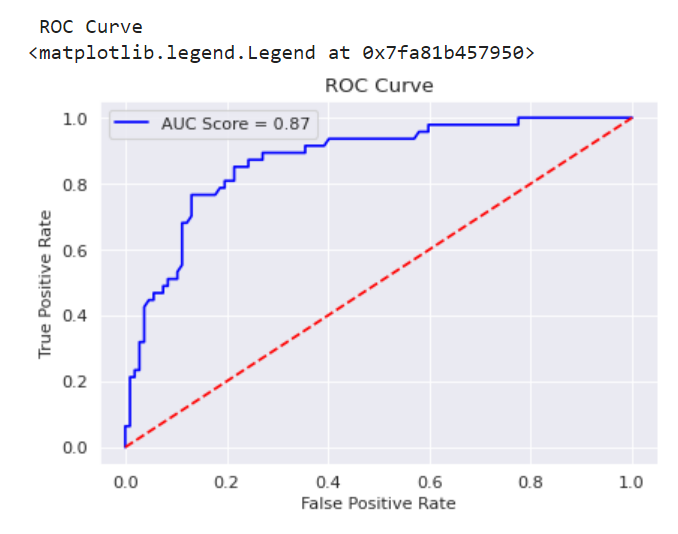


**From the below code we can got output for Model Accuracy, classification report, confusion Matrix & ROC curve:**



**Model Accuracy:**

**Classification Report**

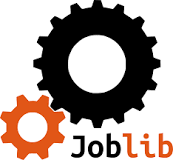
**ROC Curve**

**Observation from All Model Decision**

**So we can see Random Forest Classifier is best among all, you also wondering auc score is lesser by 1 than others also am considering it to be best because balance of classes between Precision (**is the fraction of relevant instances among the retrieved instances) **and Recall (** fraction of relevant instances that were retrieved) **is far better than other Models. So we can consider a loss in AUC by 1.**

**Model Deployment**

Deployment is the method by which you integrate a machine learning model into an existing production environment to make practical business decisions based on data. It is one of the last stages in the machine learning life cycle and can be one of the most cumbersome.

**Introduction to JOB-LIB** 

Joblib is a set of tools to provide **lightweight pipelining in Python.** In particular: transparent disk-caching of functions and lazy re-evaluation (memoize pattern) easy simple parallel computing.

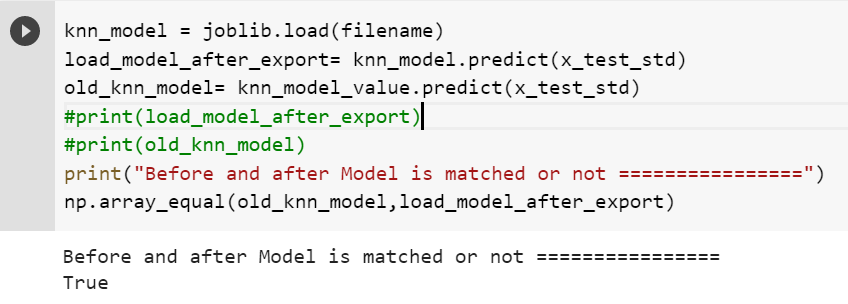
**IMPORT JOBLIB**

**KNN Model (Export):**





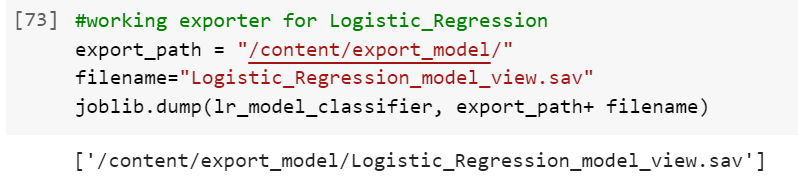
**Before and After Model is Matched or Not ?**



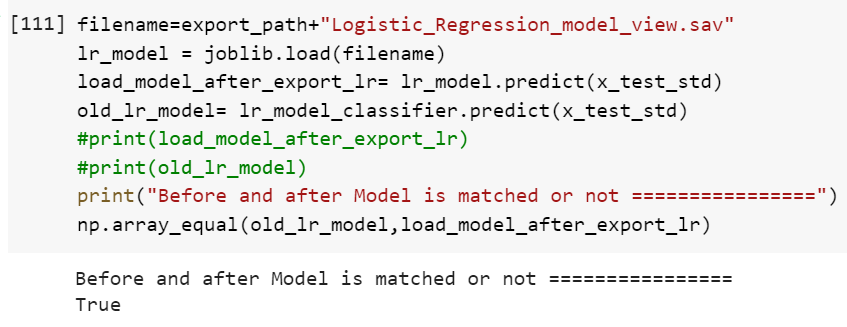
**Observation**

**So, when we started model prediction and when we export the model and import both array value are same.**

**Logistic Regression Model (Export):**





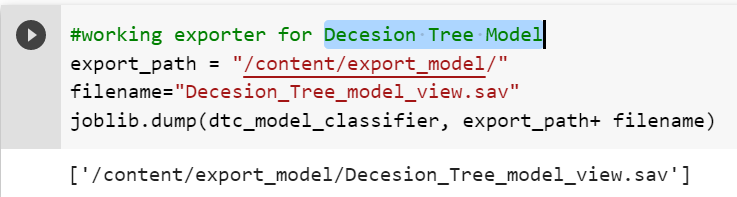


**Before and After Model is Matched or Not ?**

**Observation**

**So, when we started model prediction and when we export the model and import both array value are same.**

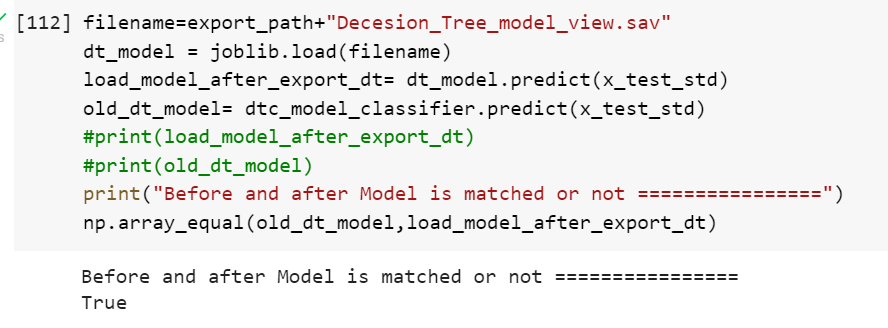
**Decision Tree Model (Export):**







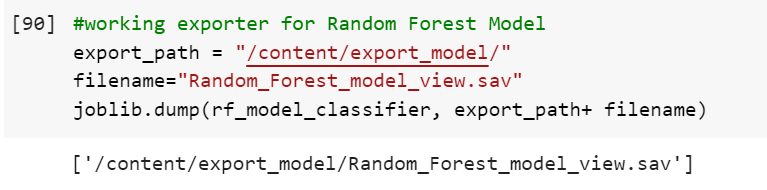
**Before and After Model is Matched or Not ?**



**Observation**

**So, when we started model prediction and when we export the model and import both array value are same.**

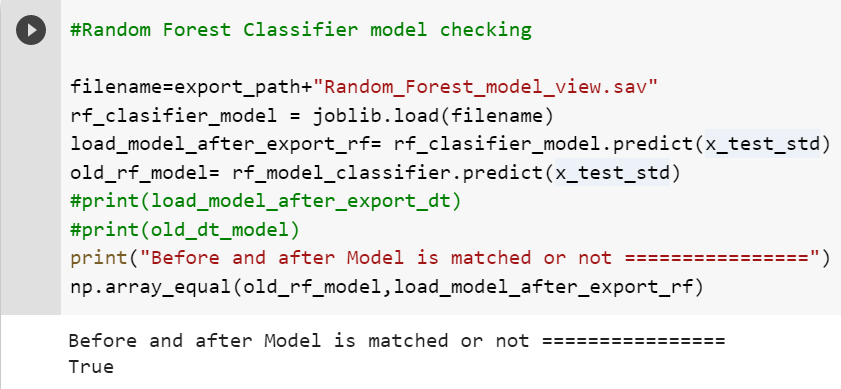
**Random Forest Classifier (Export):**







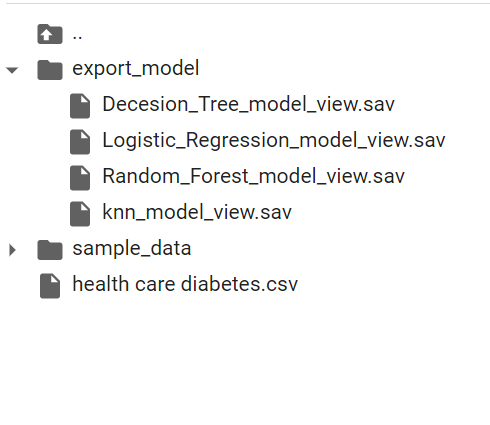
**Before and After Model is Matched or Not ?**



**Observation**

**So, when we started model prediction and when we export the model and import both array value are same.**

**Entire Project Structure**



**Data Reporting**

1. 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.

**Methodology for Data Reporting**