Diabetes Predictions

Project by Gaurav Kumar Rai

**GitHub URL**:

[GauravRai1512/UCDPA\_GauravKumarRai (github.com)](https://github.com/GauravRai1512/UCDPA_GauravKumarRai)

**Abstract**:

This has been very common health issue with age >45 due to our unbalanced unhygienic lifestyle. To predict whether the patient has diabetes or not based on the dataset featured from the **National Institute of Diabetes** and **Digestive and Kidney Disease** become very easy through Machine Learning. Dataset has different columns like Glucose, Blood Pressure, Skin Thickness, Insulin, Age & BMI which will help us to predict whether the patient has diabetes or not. Prediction will be done through Machine Learning.

This diabetes problem could be a classic example of binary classification and used KNN model to predict the accuracy. The datasets consist of feature variables and one target variable, **Outcome**.

We have loaded the csv file which we took from **Kaggle** and then started **Exploratory Data Analysis** (EDA) through checking head (), info (), shape () function to get the actual rows and columns details.

Then Data Preparation where Data cleaning process will come into the picture where we must check and clean all possible data which has 0 value NaN and duplicate values.

We build a machine learning model which is **KNN classifier model** of supervised learning to train the data, fit the data, predict the data whether the patients in the dataset have diabetes or not.

**Introduction**:

Diabetes has become very frequent health issue and if we will not be taking care of this at the right time, you will be in trouble or you can say, you will be following towards the end of your life as high diabetes level can damage all your parts, along with this to identify whether patients have diabetes or not is a bit costly treatment and procedures related to treating diabetes and its complications. So, such kind of Machine Learning Model can help us to create the process where you must feed the data like Glucose, BMI and it will predict whether you have diabetes or not.

The prime objective of this project is to predict diabetes based on data available in the data frame, by identifying which type of algorithm model works best for this prediction.

Use Case:

1. A very simple way to identify the diabetes at early stage and better treatment can help to cure from this.
2. Reduce costs for your health care organization as this would help to avoid costly treatments and procedures related to treating diabetes and its complications.

**Dataset**:

Source: [Aditya-Mankar/Diabetes-Prediction: Predict Diabetes using Machine Learning. (github.com)](https://github.com/Aditya-Mankar/Diabetes-Prediction)

Description: 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. All patients here are females at least 21 years old of Pima Indian heritage.

This source has taken the reference from Kaggle only.

I have chosen this source because the details are very clear and very well explained without any complexity.

**Implementation Process**:

This process has started from importing all required python dependencies, then loading the csv file which I picked from Kaggle and did some modification like added NaN and 1 duplicate value.

Did data analysis as well to check the info, shape, and head values.

Then started cleaning data for further processing where removed all NaN value present in the dataset as the occurrence was less than 5% of total data in csv, then converted all 0 to NaN this time we have not removed the data as occurrence was more than 10% so instead of removing this, calculated the mean value and replaced with NaN. Then also deleted duplicate value as well.

Then started data visualization through hist before cleaning the data and then after cleaning the data and checked the state of data through **histogram**.

Chosen 2 columns Glucose and BMI from the dataset and started plotting through **scatter** and through visualization we can say that when BMI increased Glucose level has also increased and high Glucose level means probability of having diabetes in patient has been increased.

Then finally started Data Modeling where we have used multiple algorithms to predict the accuracy.

1. **K-NN Algorithm Accuracy**: 78.69565217391305
2. **SVC Algorithm Accuracy**: 76.08695652173914
3. **Linear Regression Accuracy**: 31.660043670199534
4. **Logistic Regression Accuracy**: 78.26086956521739

After applying multiple algo and the accuracy which we have seen except Linear all are almost same with Accuracy but if I must choose best algo accuracy as per data then I would say **K-NN Algorithm** & **Logistic Regression** would be the best Supervised model for Prediction but confusion Matrix and **classification report has best result for K-NN Algorithm so I would choose K-NN Algorithm for our diabetes data to predict correctly.**

Have used **Train/test split** to measure model performance where we have started to fetch feature(X) and output(y) where feature would be all column except the output and passed as an argument X and y in **train\_test\_split** also we took 30% data as a test data.

Also have used **iterator** to run the classifier with different K value to check **over/underfitting** also get the accuracy of K neighbor algorithm as it works based on number means K=1,2,3…. can put this as an argument in KNN Model as an KNeighborsClassifier to predict the accuracy. Considering function and all would be same so no need to do this one by one.

Larger k=less complex model=can cause underfitting

Larger k=more complex model=can lead to overfitting

So used for loop with range from 1 to 26 and pass this range value as a variable into this KNN model and the highest accuracy we have achieved with **n\_neighbors=7**

**Confusion Matrix** has also used to summarize the performance of a classification algorithm. As sometimes accuracy could be misleading if you have an unequal number of observations in each class.so calculating a confusion matrix can give you a better idea of what your classification model is getting right and what types of error its making and based on that you can tell the whether this classification model is the best or not in any circumstances.

Confusion Matrix of **K-Neighbor** classifier as below:

[[[[131 18]

[ 31 50]]

# As per above The Model produced 131 true positive, 50 true negative, 18 false negative, and 31 false positive.

Also calculated **Classification Report** to get the **precision, recall, f1-score**

# The classification report shows a better recall & f1 score for the zero class, which represents individuals who do not have diabetes.

**KNN** precision recall f1-score support

0 0.81 0.88 0.84 149

1 0.74 0.62 0.67 81

accuracy 0.79 230

macro avg 0.77 0.75 0.76 230

weighted avg 0.78 0.79 0.78 230

Confusion Matrix of **Logistic Regression** as below:

[[118 30]

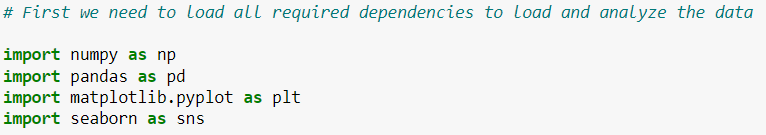
[ 60 22]]

# As per above The Model produced 118 true positive, 22 true negative, 30 false negative, and 60 false positive.

Also calculated **Classification Report** to get the **precision, recall, f1-score**

# The classification report shows a better recall for the zero class, which represents individuals who do not have diabetes but accuracy is less as comparison to KNN as KNN has 79% accuracy as per classification report

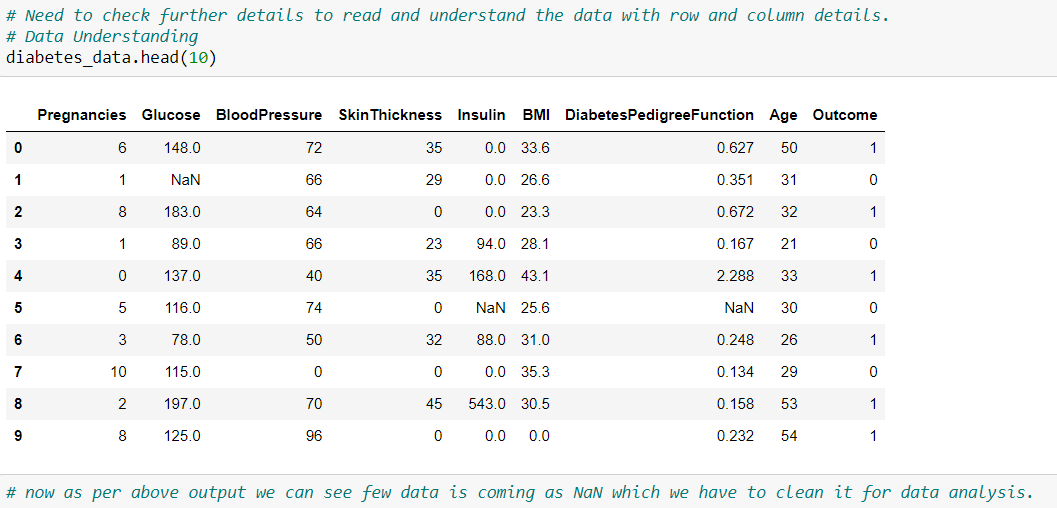
1. Importing Libraries and Dataset

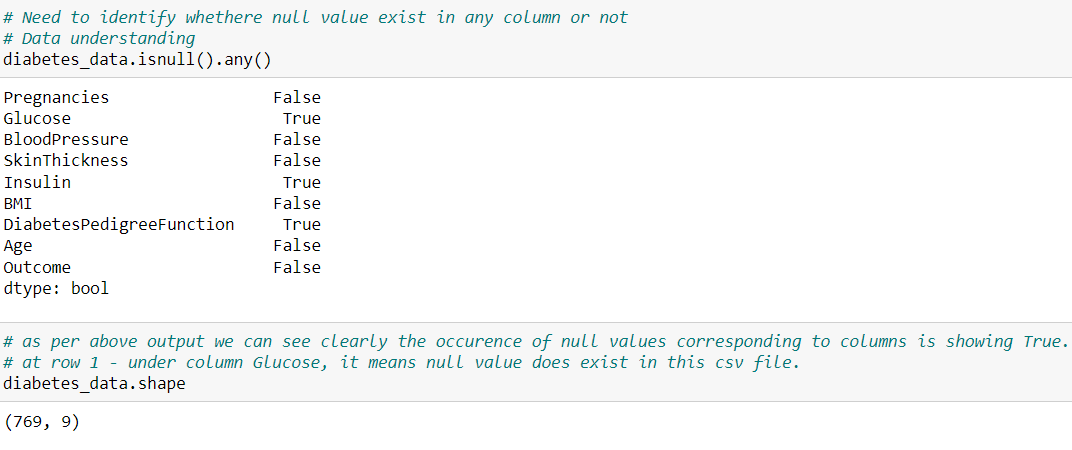


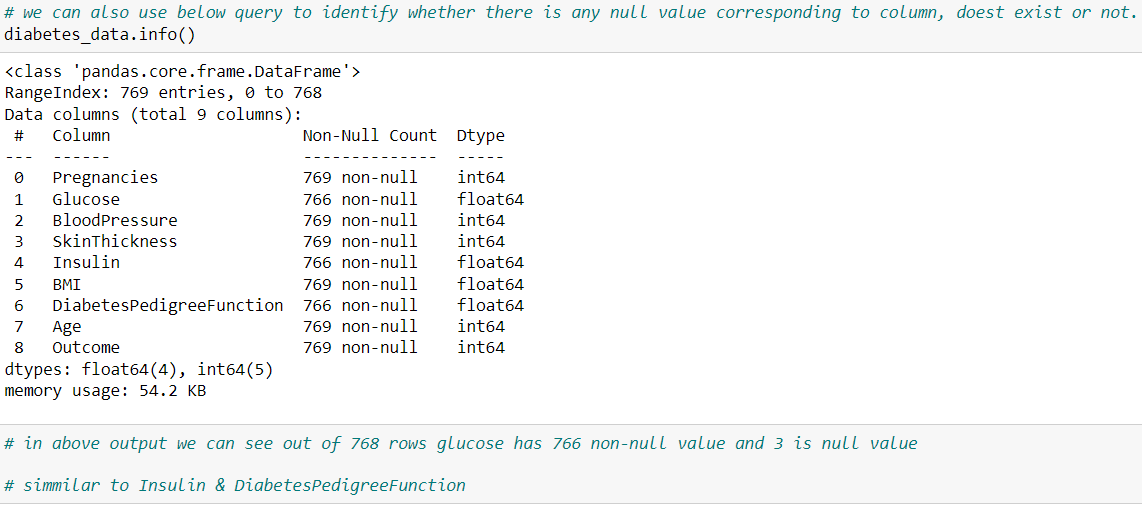
1. Data Loading

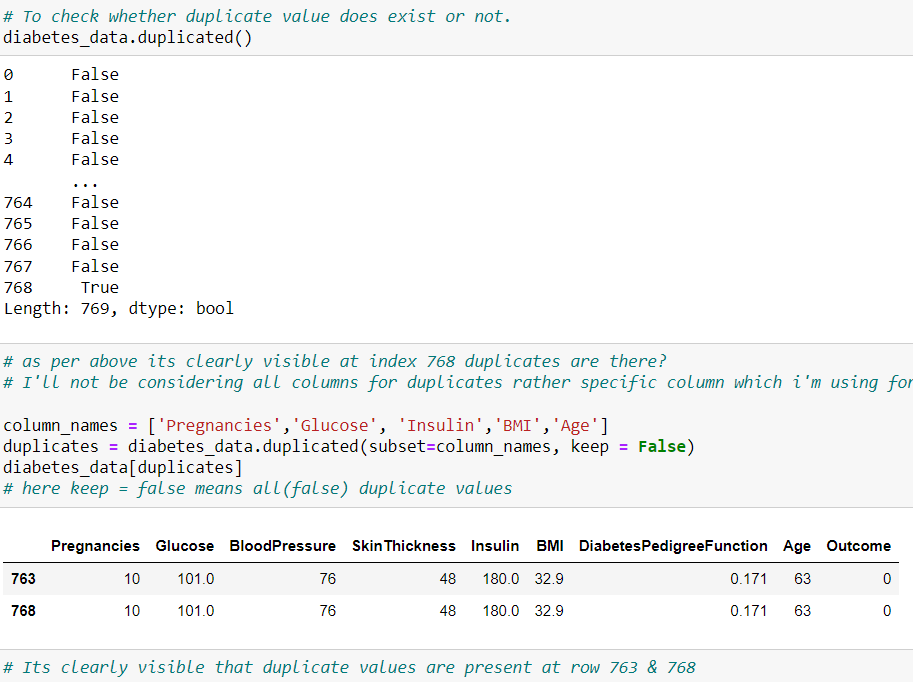


1. Exploratory Data Analysis

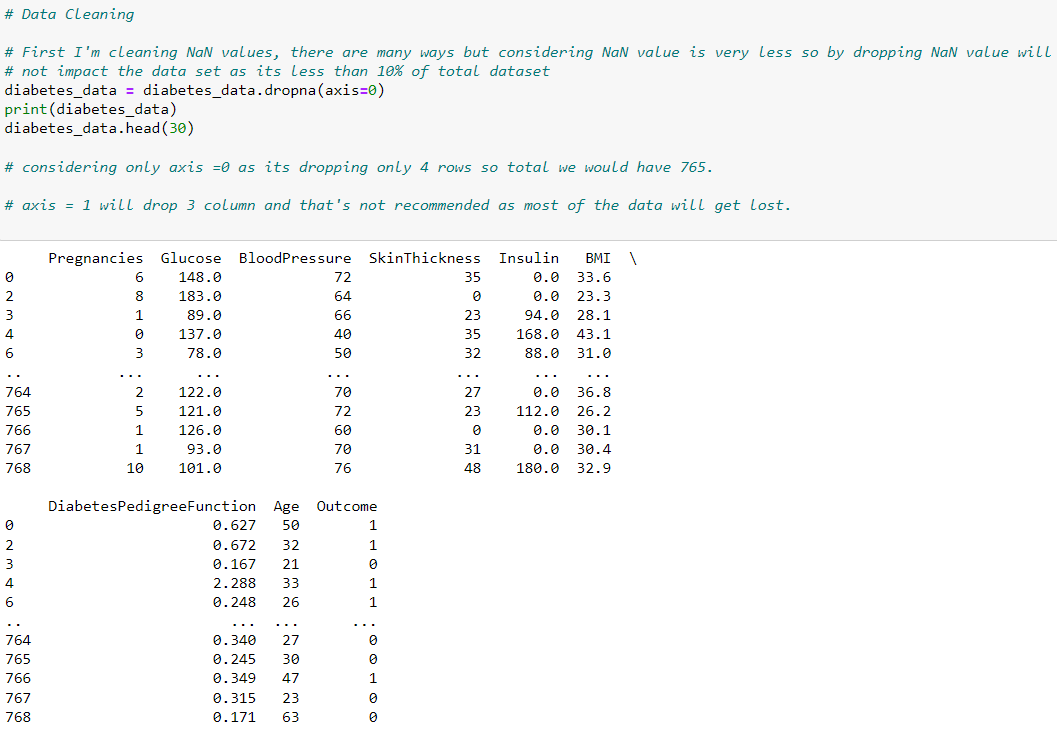


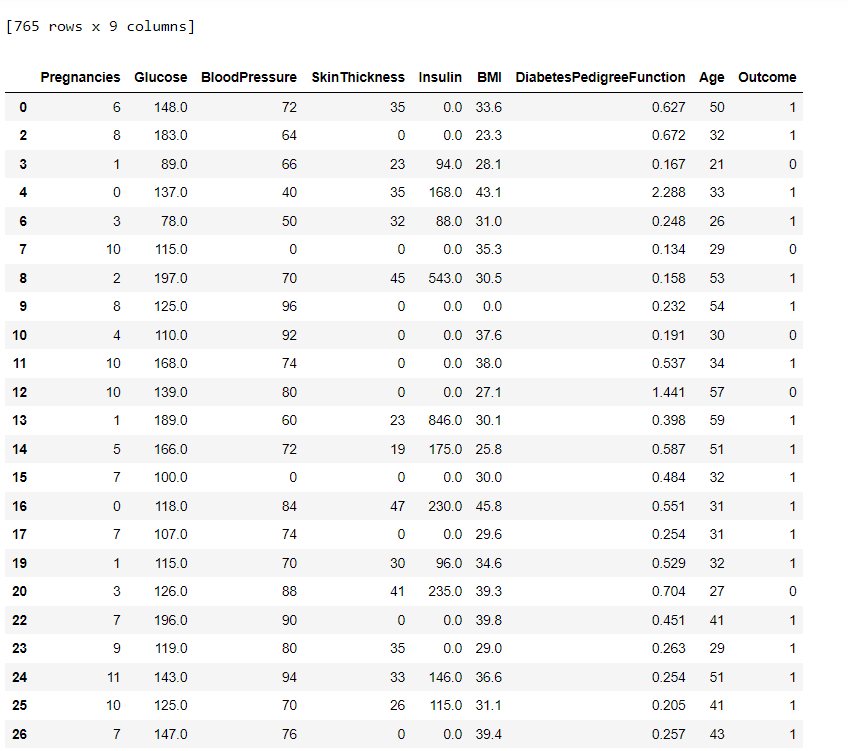


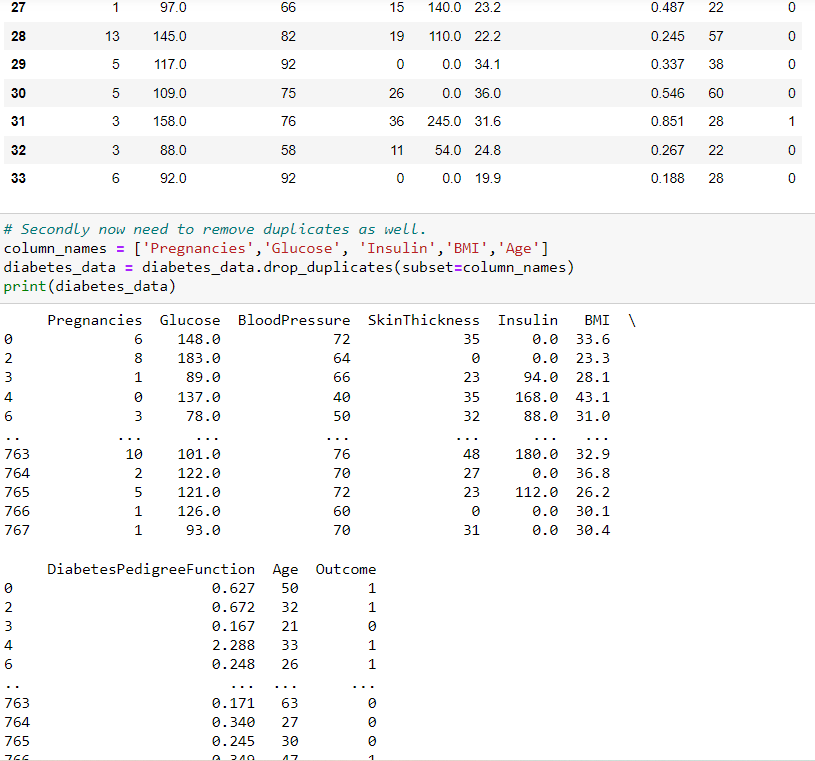


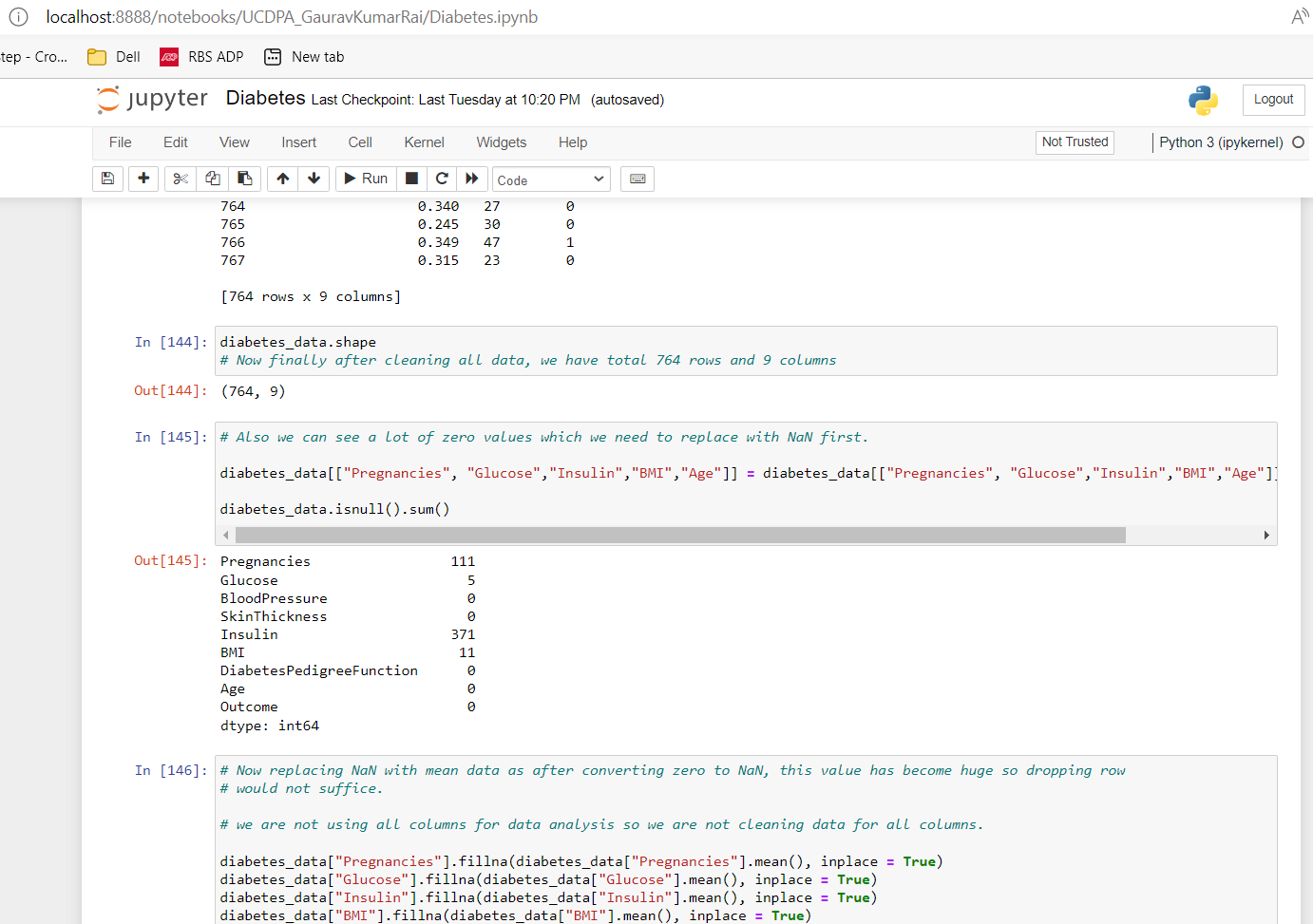


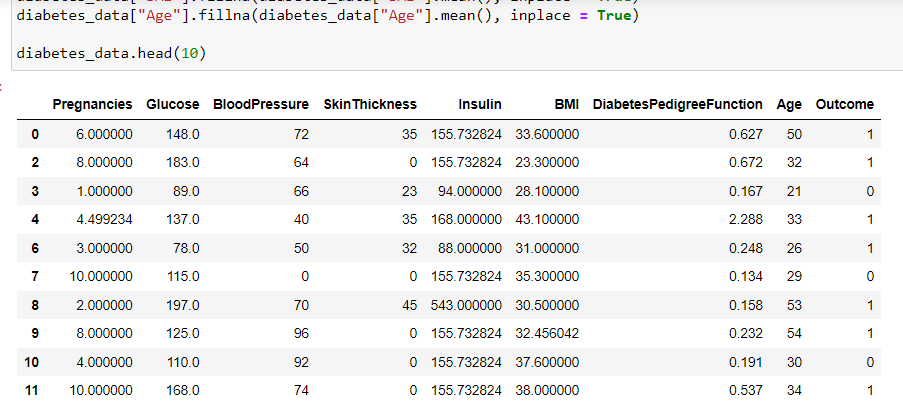
1. Data Cleaning



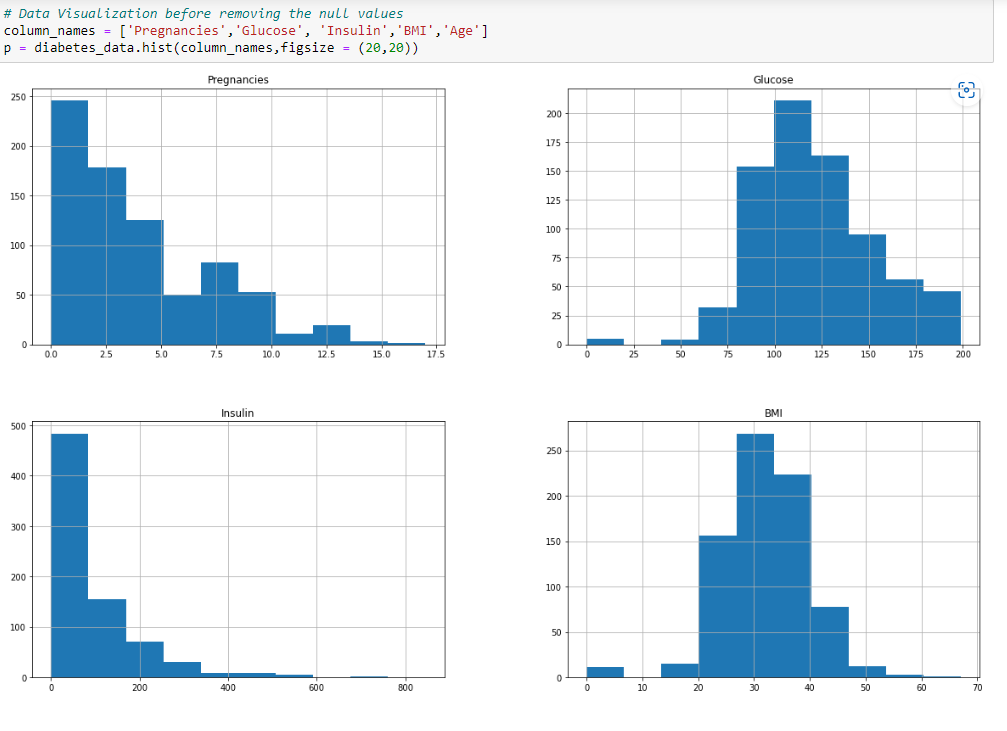


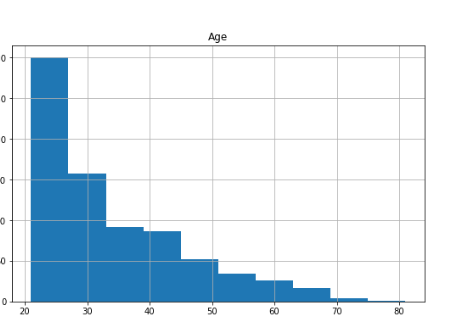


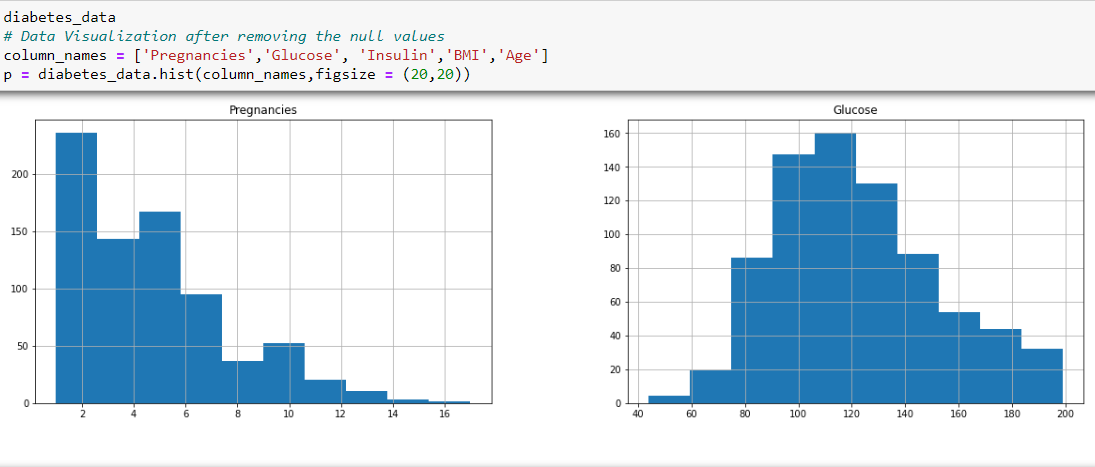


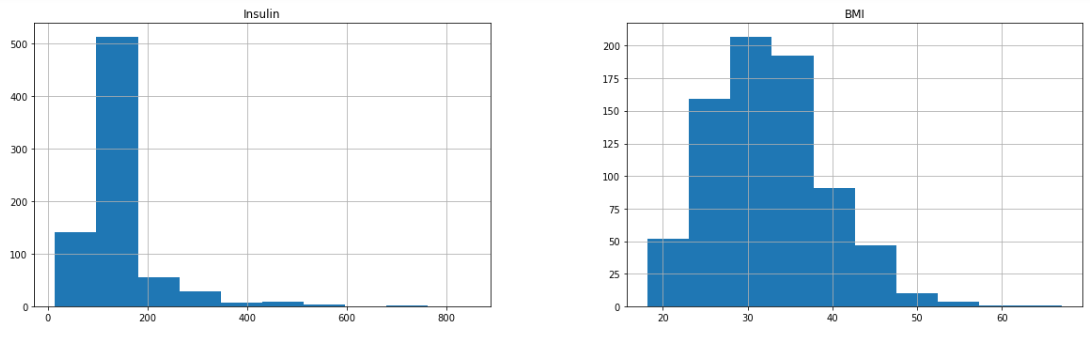


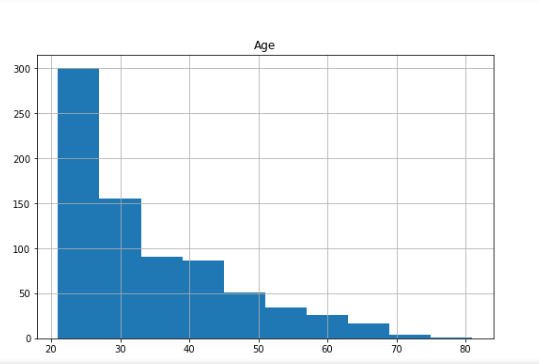
1. Data Visualization:

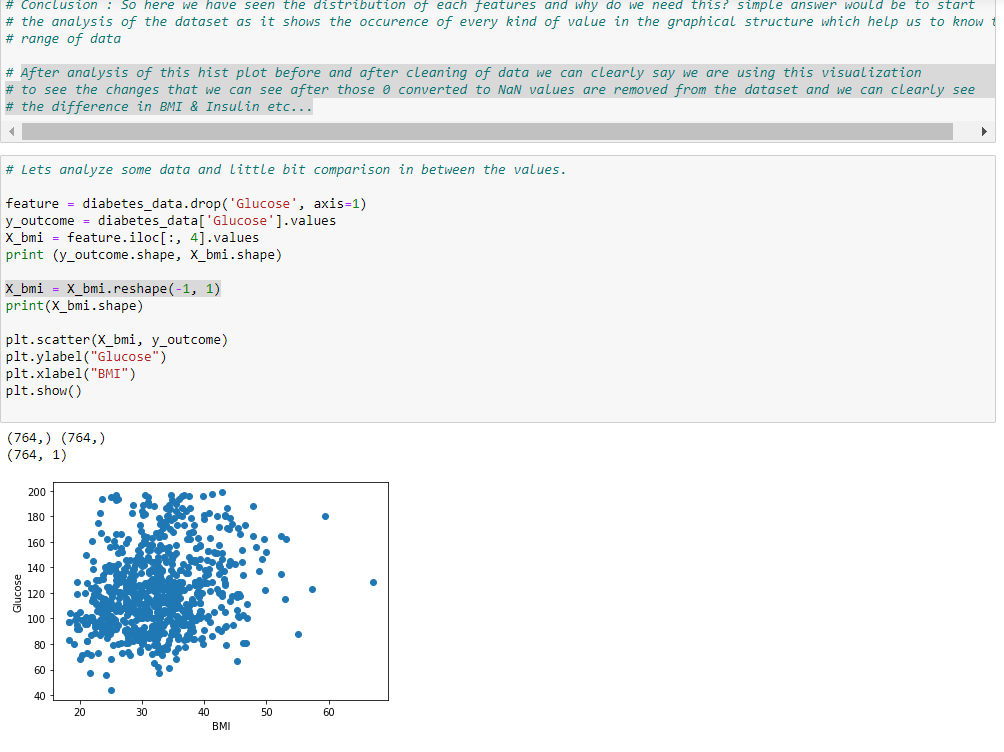






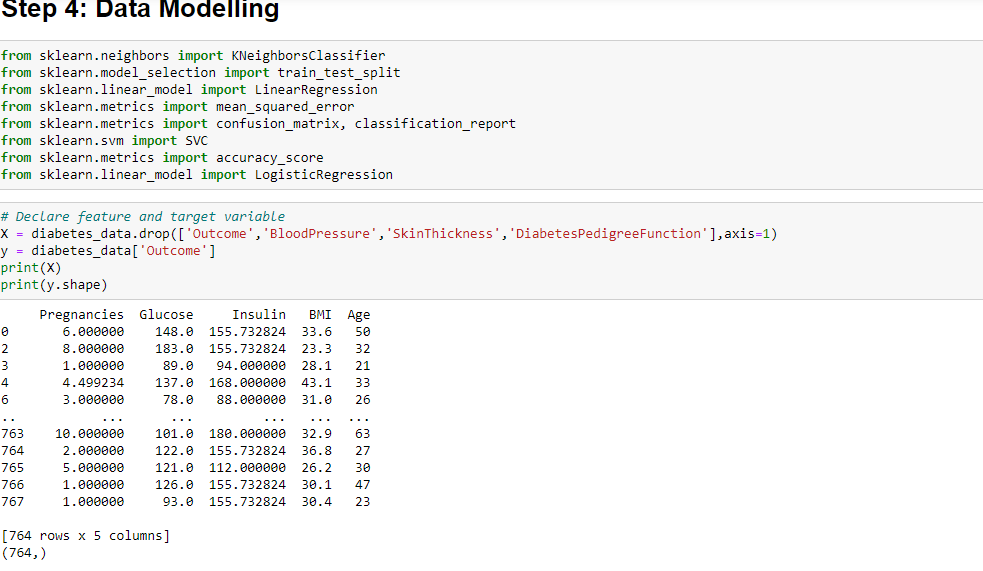


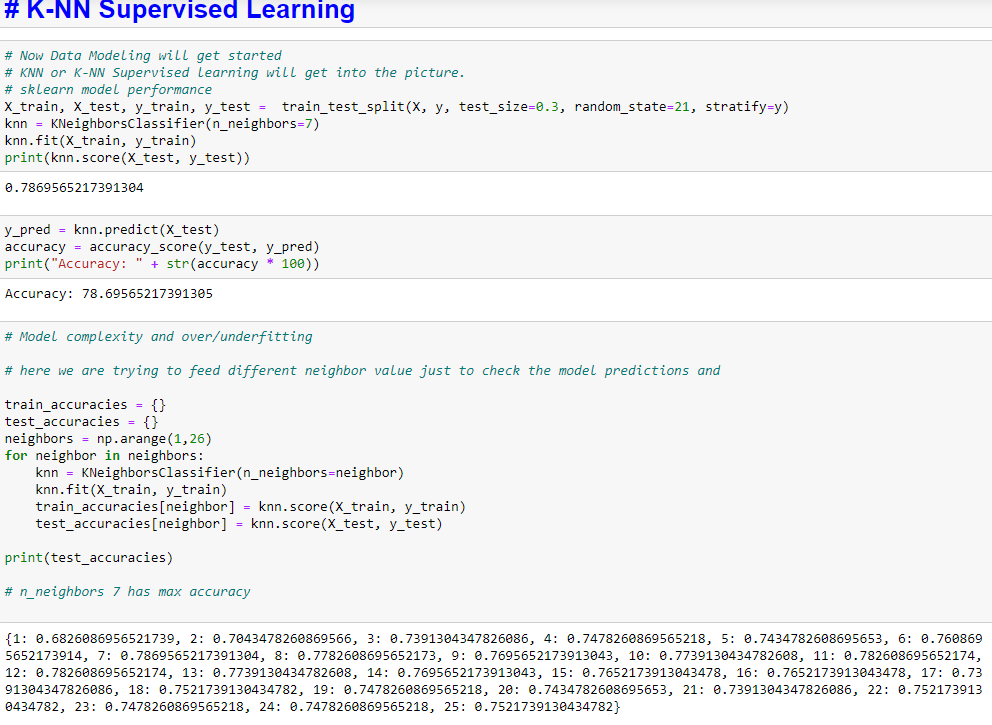




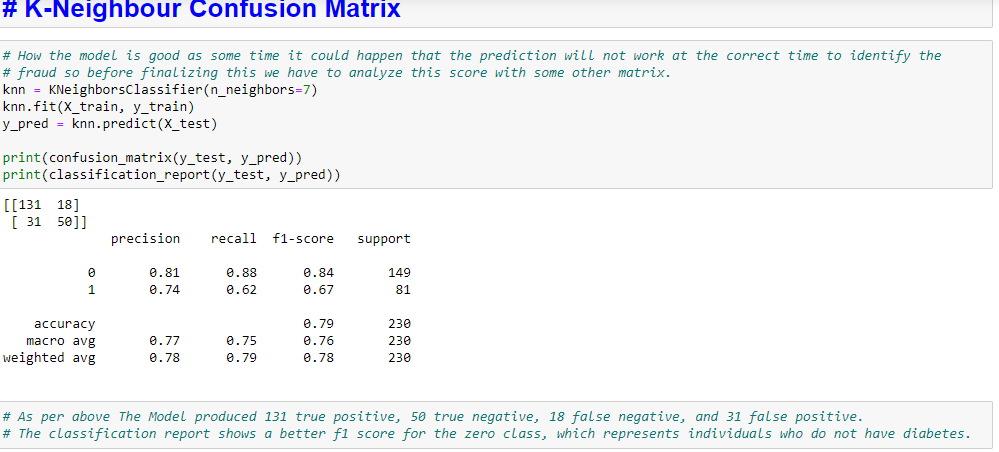


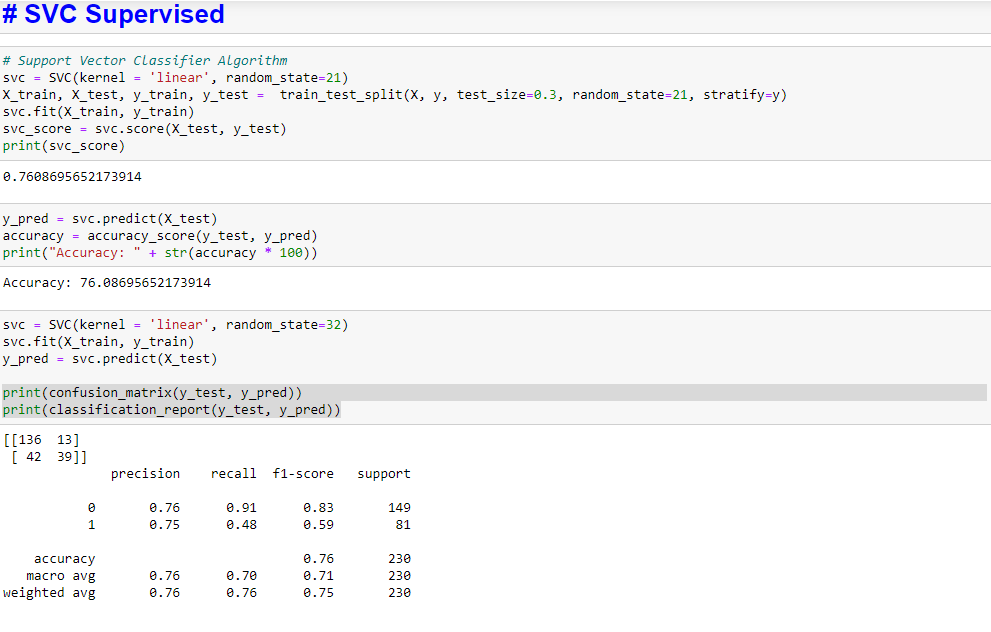
1. Data Modelling:

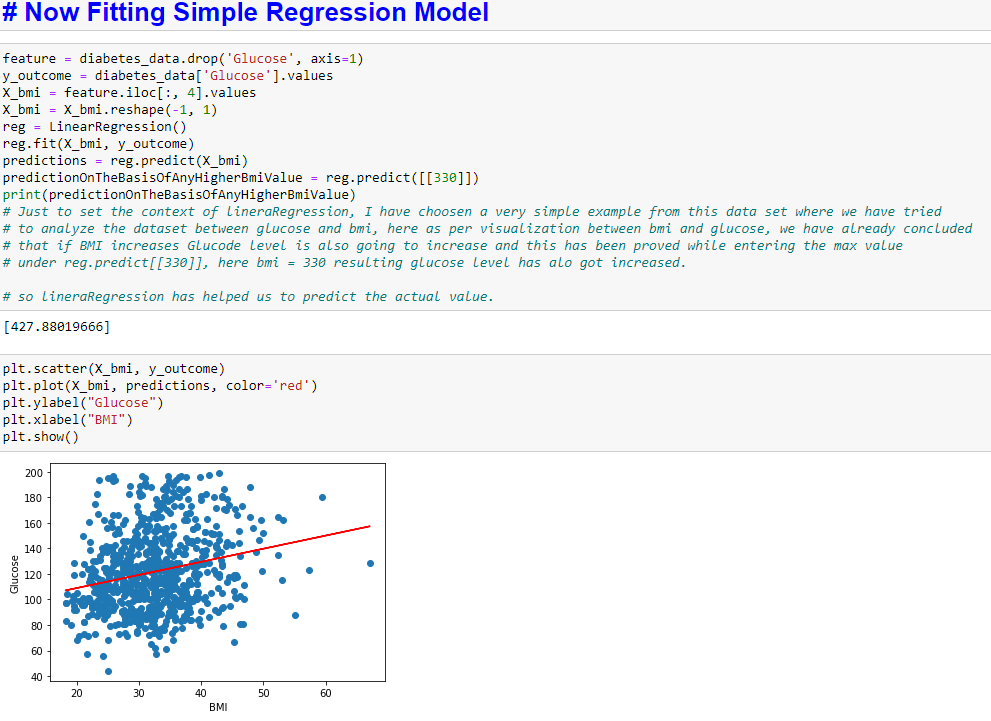




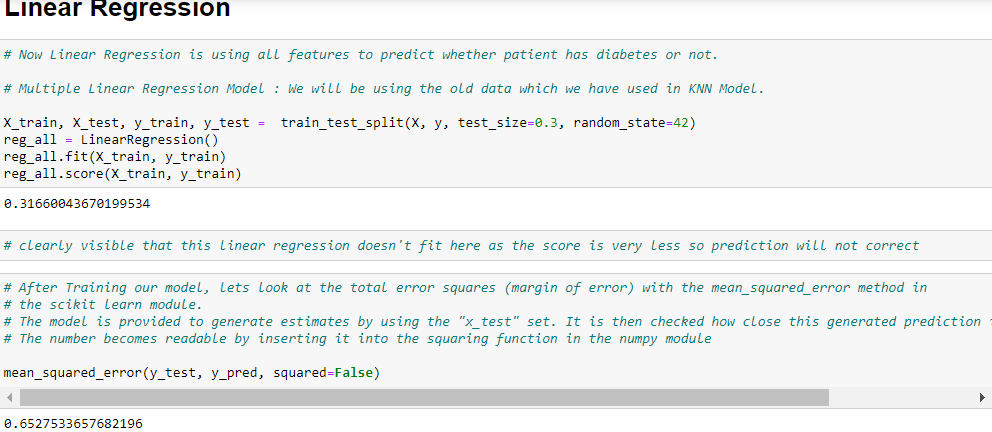


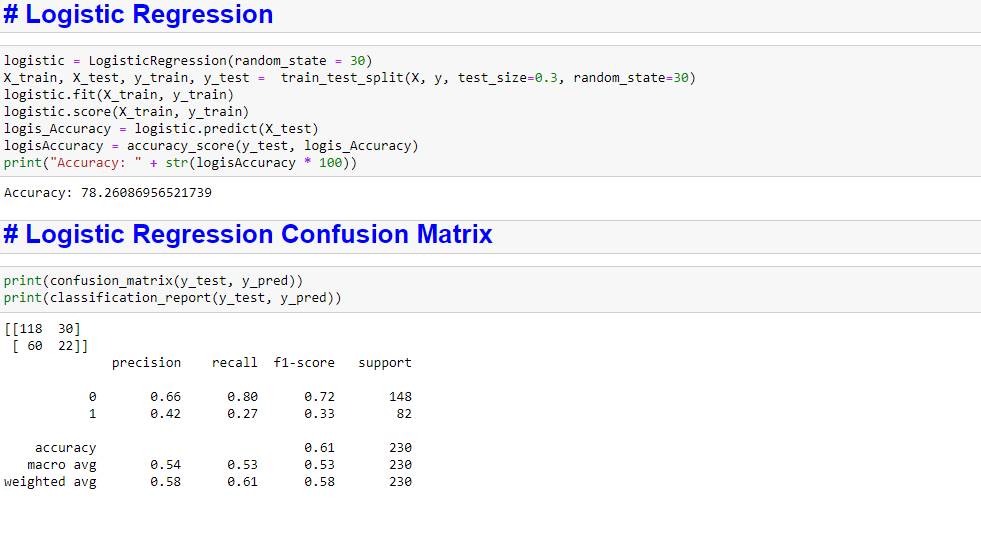


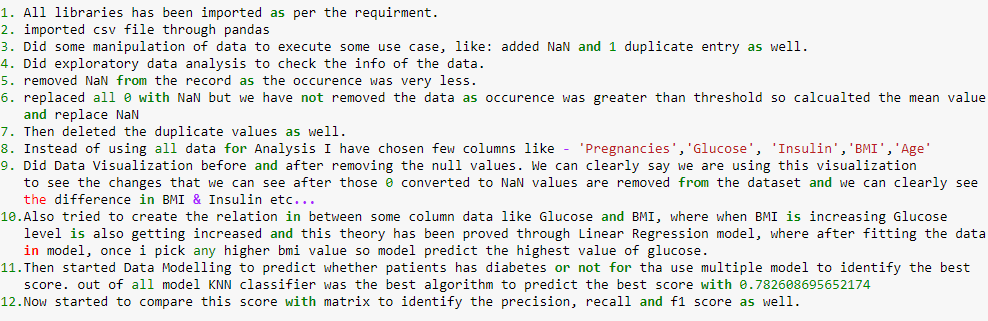












**Results**:

1. **K-NN Algorithm Accuracy**: 78.69565217391305
2. **SVC Algorithm Accuracy**: 76.08695652173914
3. **Linear Regression Accuracy**: 31.660043670199534
4. **Logistic Regression Accuracy**: 78.26086956521739

After implementing multiple Algorithm along with the confusion matrix and classification report, we can clearly say K-NN Algo has better accuracy as per below result.

As per above data K-NN Algo has better accuracy and better precision, recall & f1-score in comparison to Logistic Regression Model.

K-Neighbor classifier would be the best diabetes prediction for Patients.

**Insights:**