DIABETIC CELL DETECTION USING LOGISTIC REGRESSION

A PROJECT REPORT

In partial fulfilment of the requirements for the award for this training in **VALUABLE ADDED TRAINING (VAT)** *Under the guidance of*

Sofikul Mullick

BY

SWARNADEEP JANA	**10905519038**
KIRAN DEY	**10905519037**
KRISHANU ADAK	**10905519039**

Netaji Subhash Engineering College, Garia

In association with

Engineering Study Centre (ESC)

(Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)

1.	Title of the Project:	**DIABETIC CELL DETECTION USING LOGISTIC REGRESSION**
2.	Project Members:	**SWARNADEEP JANA** **KIRAN DEY** **KRISHANU ADAK**
3.	Name of the guide:	**SOFIKUL MULLICK**

Project Version Control History

Version	Primary Author	Description of Version	Date Completed
Final	Swarnadeep Jana Kiran Dey Krishanu Adak	Project Report	1st June,2021

*Swarnadeep Jana*_____

Signature of Head of the Team
Date: 1.06.2021

_____

Signature of Approver
Date:

For Office Use Only

Approved **Not Approved**

SOFIKUL MULLICK
Project Proposal Evaluator

DECLARATION

We hereby declare that the project work being presented in the project proposal entitled **"DIABETIC CELL DETECTION USING LOGISTIC REGRESSION"** in partial fulfilment of the requirements for the award for this training in **VALUABLE ADDED TRAINING (VAT)** Under the at **ENGINEERING STUDY CENTER, SALTLAKE, KOLKATA, WEST BENGAL**, is an authentic work carried out under the guidance of **SOFIKUL MULLICK**. The matter embodied in this project work has not been submitted elsewhere for the award of any degree of our knowledge and belief.

Date:

Name of the Student: Swarnadeep Jana
Kiran Dey
Krishanu Adak

Signature of the students:

1. *Swarnadeep Jana*
2. *Kiran Dey*
3. *Krishanu Adak*

CERTIFICATE

This is to certify that this proposal of minor project entitled **"DIABETIC CELL DETECTION USING LOGISTIC REGRESSION"** is a record of bona fide work, carried out by **Swarnadeep Jana, Kiran Dey, Krishanu Adak** under my guidance at **Engineers Study Center**. In my opinion, the report in its present form is in partial fulfilment of the requirements for the award for this training in **VALUABLE ADDED TRAINING (VAT)** and as per regulations of the **Engineers Study Center®**. To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report.

Guide / Supervisor

SOFIKUL MULLICK

Project Engineer
Engineers Study Center

ACKNOWLEDGEMENT

Success of any project depends largely on the encouragement and guidelines of many others. I take this sincere opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project work.

I would like to show our greatest appreciation to **SOFIKUL MULLICK, Project Engineer at Engineering Study Center, Kolkata**. I always feel motivated and encouraged every time by his valuable advice and constant inspiration; without his encouragement and guidance this project would not have materialized.

Words are inadequate in offering our thanks to the other trainees, project assistants and other members at Engineers Study Center. for their encouragement and cooperation in carrying out this project work. The guidance and support received from all the members and who are contributing to this project, was vital for the success of this project.

INDEX

ABSTRACT

Due to its continuously increasing occurrence, more and more families are influenced by diabetes mellitus. Most diabetics know little about their healtquality or the risk factors they face prior to diagnosis. In this study, we have proposed a novel model based on data mining techniques for predicting type 2 diabetes mellitus (T2DM). The main problems that we are trying to solve are to improve the accuracy of the prediction model, and to make the model adaptive to more than one dataset. Based on a series of preprocessing procedures, the model is comprised of two parts, the improved K-means algorithm and the logistic regression algorithm. The Pima Indians Diabetes Dataset and the Waikato Environment for Knowledge Analysis toolkit were utilized to compare our results with the results from other researchers. The conclusion shows that the model attained a 3.04% higher accuracy of prediction than those of other researchers. Moreover, our model ensures that the dataset quality is sufficient. To further evaluate the performance of our model, we applied it to two other diabetes datasets. Both experiments' results show good performance. As a result, the model is shown to be useful for the realistic health management of diabetes.

INTRODUCTION

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

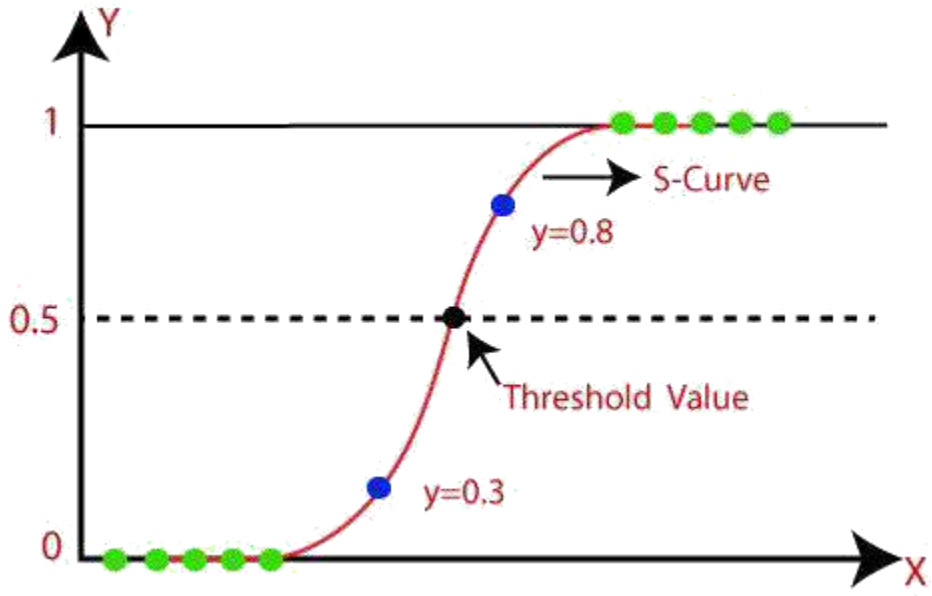
Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is

dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.



Type of Logistic Regression:

On the basis of the categories, Logistic Regression can be classified into three types:

- **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
- **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
- **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

PROBLEM DEFINITION

Diabetes mellitus, commonly known as diabetes, is a metabolic disease that causes high blood sugar. The hormone insulin moves sugar from the blood into your cells to be stored or used for energy. With diabetes, your body either doesn't make enough insulin or can't effectively use the insulin it does make.

So here we calculate how much the possibilities to have a diabetic cell. We can know that from where the diabetic Cell started increasing.

PROJECT GOAL

The goal is to predict whether a person has diabetic cell or not. The goal is achieved by using logistic regression in machine learning. Diabetes is a common for all people around the world,mainly in india we know that every 100people there will be 5people who have diabetes. Early detection of diabetic cell can greatly improve prognosis by promoting clinical treatment to patients.

METHODOLOGY

• **Data Selection:** Data is the foundation for any machine learning project. The job is to find ways and sources of collecting relevant and comprehensive data, interpreting it, and analyzing results with the help of statistical techniques.

• **Data cleaning:** This set of procedures allows for removing noise and fixing inconsistencies in data. A data scientist can fill in missing data using imputation techniques. A specialist also detects outliers — observations that deviate significantly from the rest of distribution.

• **Dependent and independent of data**: If the values in one sample affect the values in the sample,then the samples are dependent. If the values in one sample reveal no information about those of the other sample, then the samples are independent.

• **Data Visualization:** A large amount of information represented in graphic form is easier to understand and analyze. Some companies specify that a data analyst must know how to create slides, diagrams, charts, and templates.

• **Data Splitting:** A dataset used for machine learning should be partitioned into three subsets — training, test, and validation sets.

• **Model Selection:** After a data scientist has preprocessed the collected data and split it into three subsets, he or she can proceed with a model training. This process entails “feeding” the algorithm with training data. An algorithm will process data and output a model that is able to find a target value in new data. The purpose of model training is to develop a model.

• **Model Evaluation:** The goal of this step is to develop the simplest model able to formulate a target value fast and well enough and check the accuracy

PROJECT OBJECTIVE

The main objective of this is diabetic Cell Detection using Logistic Regression.

PROJECT IMPLEMENTATION

• **SELECTION OF DATA:** The process of selecting data depends on the type of project we desire to. The data set can be collected from various sources such as a file, database, sensor and many other such sources.

• **VISUALIZATION OF DATA:** Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns.

• **SELECTION OF DEPENDENT AND INDEPENDENT DATA:**
We need to select the dependent and independent data and store them in y and x.

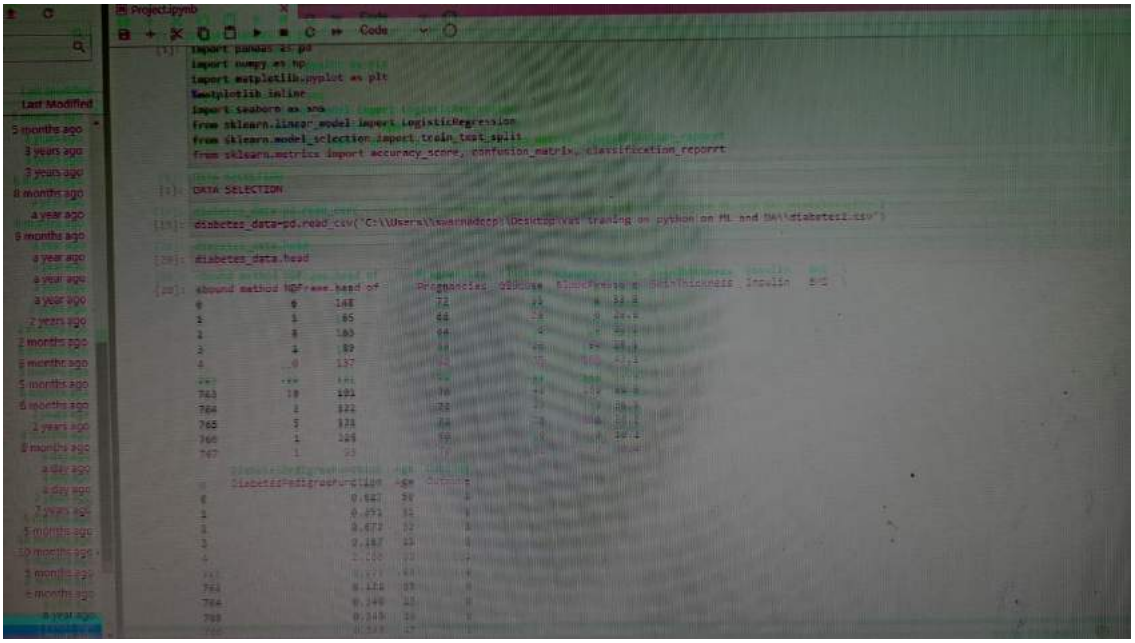
• **SPLITTING OF THE DATA:** We train the classifier using 'training data set', then test the performance of your classifier on unseen 'test data set'. We split the data for training and testing by using the 'train_test_split'.

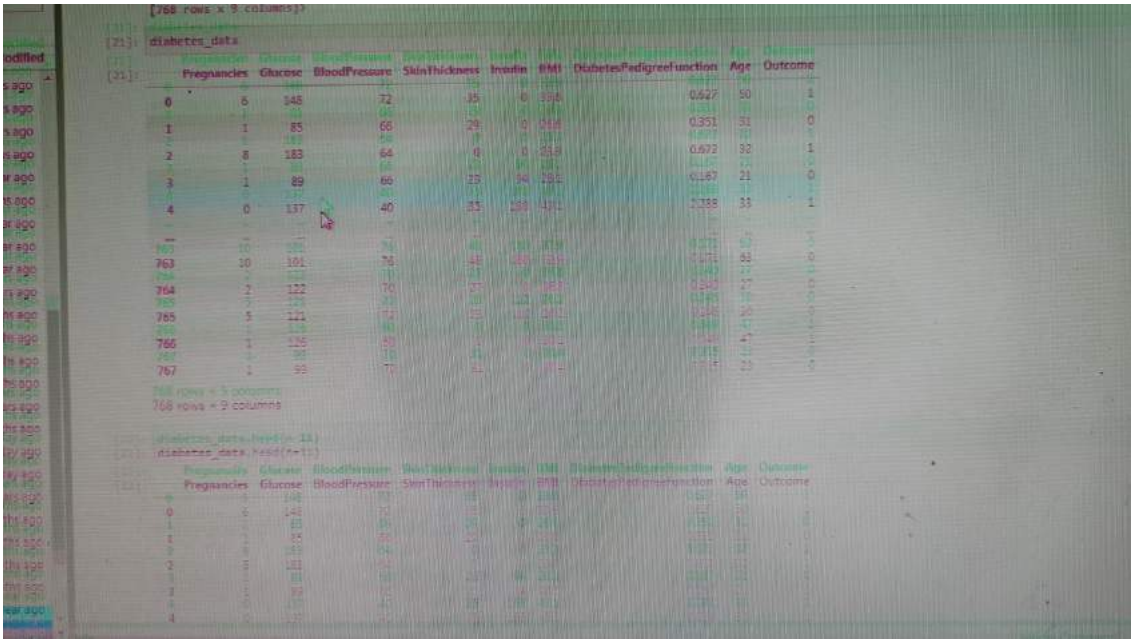
• **FITTING THE MODEL:** In a data set, a training set is implemented to build up a model. Once the model is trained, we can use the same trained model to predict using the testing data i.e., the unseen data. Once this is done, we can develop a confusion matrix, this tells us how well our model is trained.

• **MODEL EVALUATION:** It is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future.

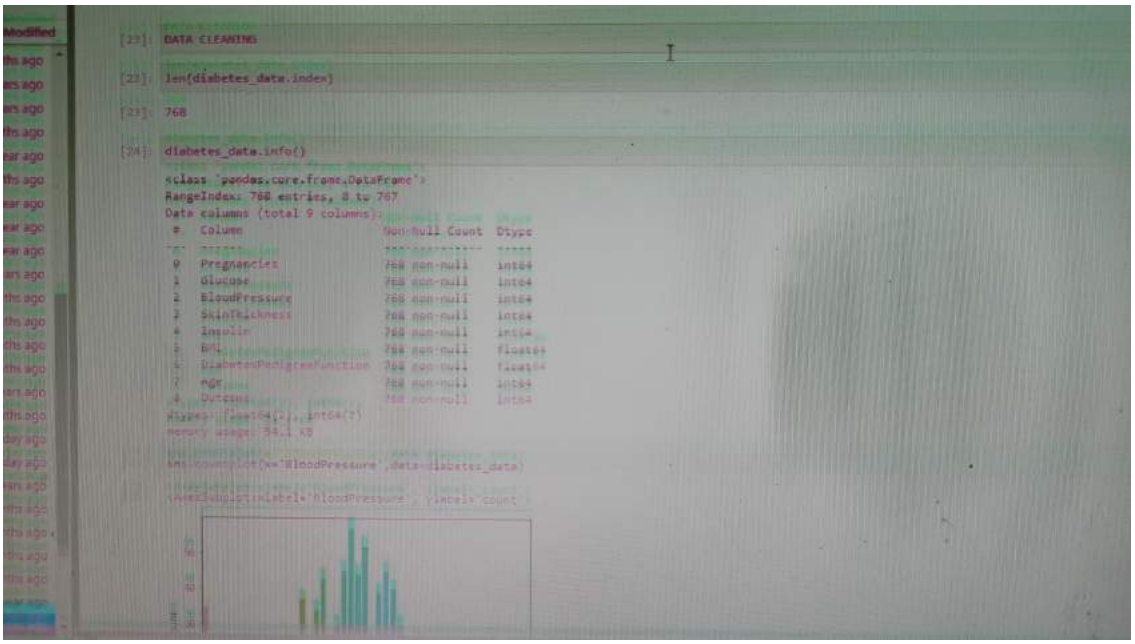
STEP BY STEP WORKING:

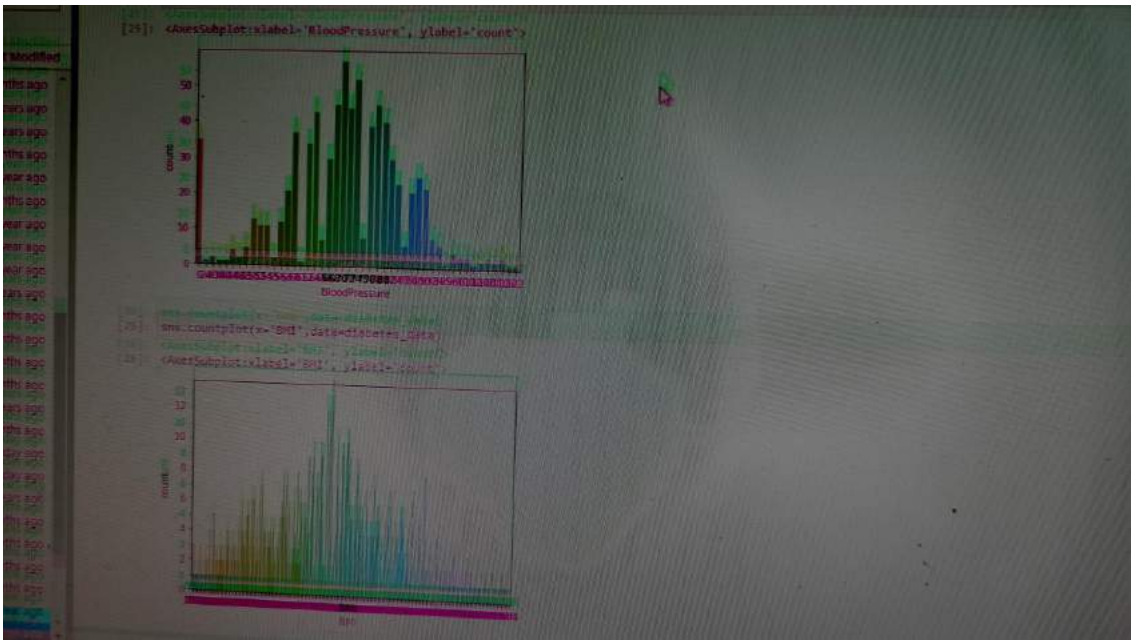
DATA SELECTION:



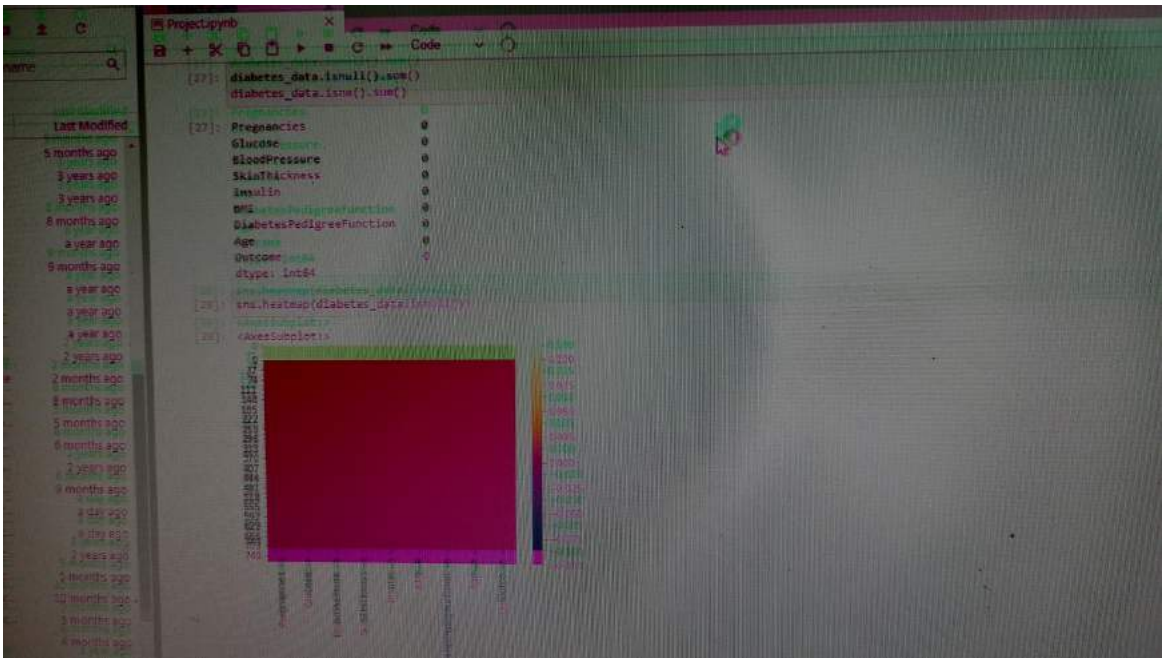


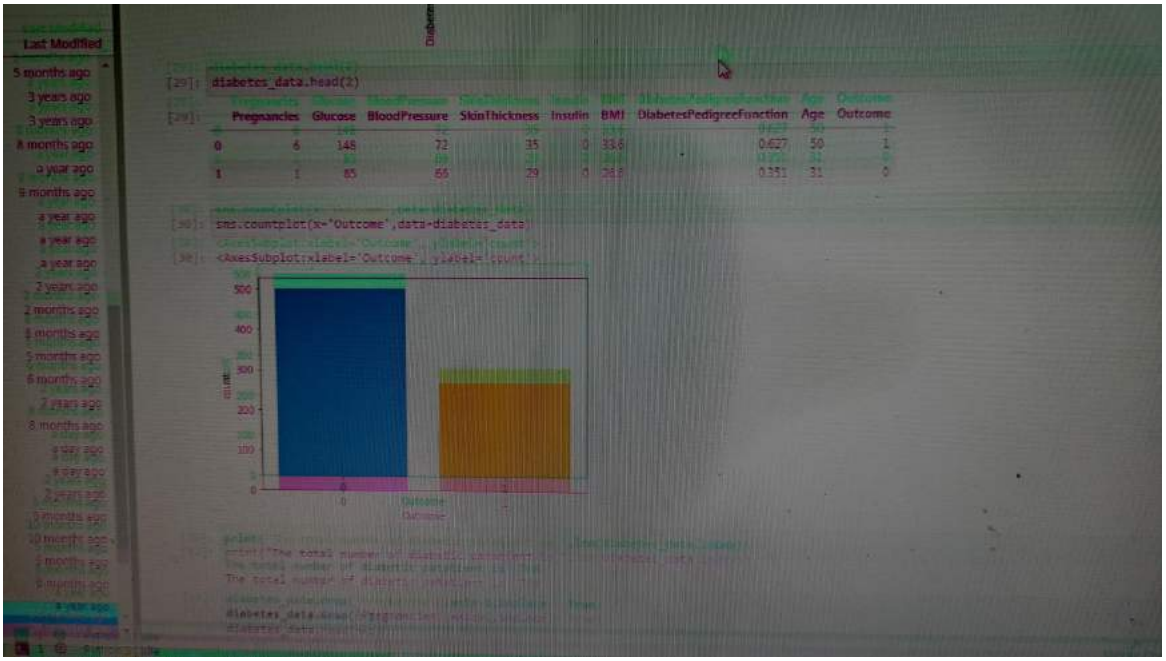
CLEANING DATA:



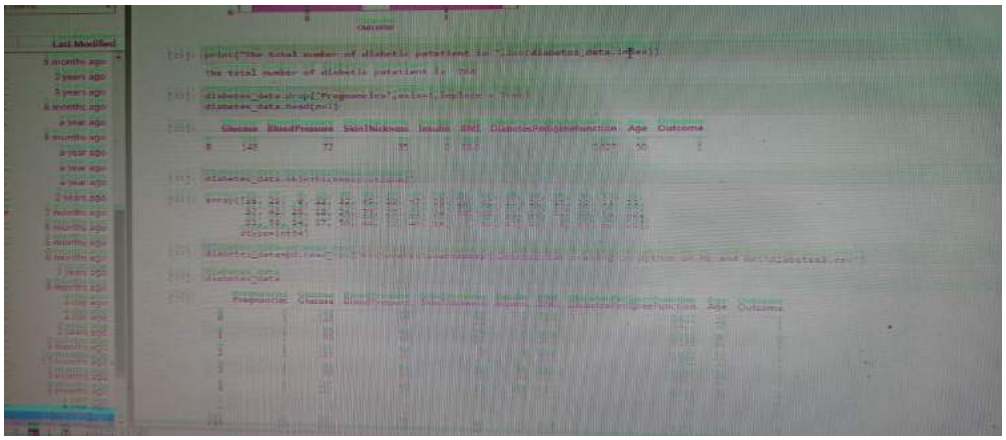


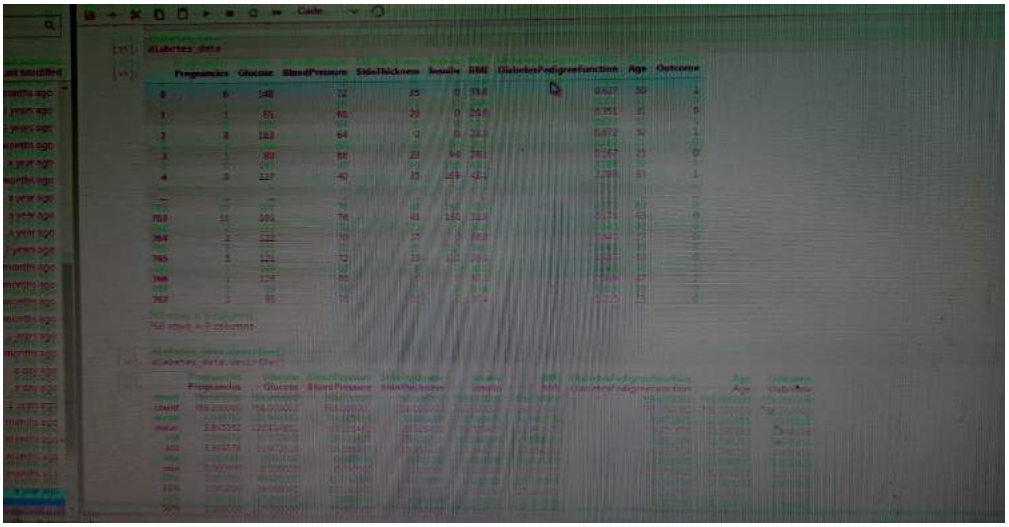
CLEANING DATA:

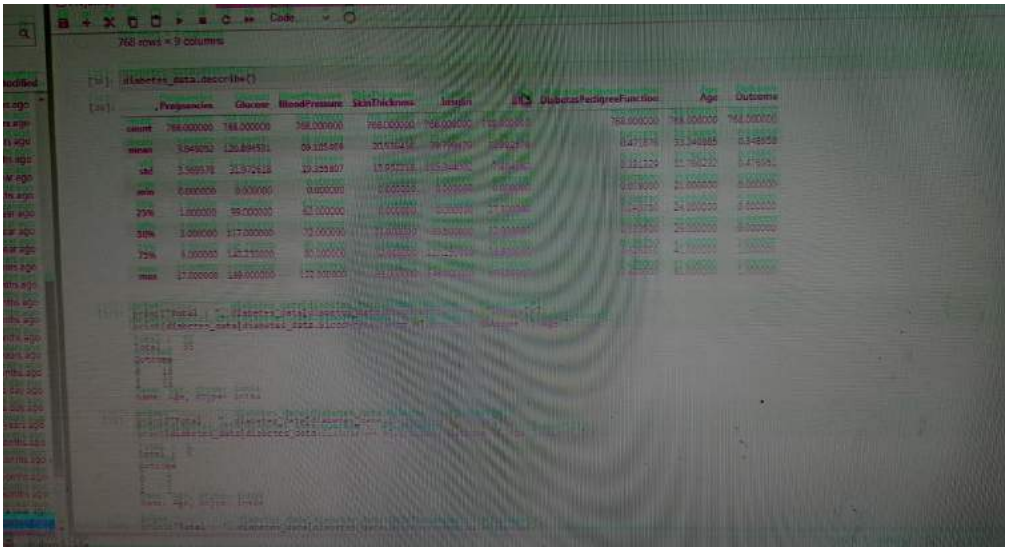




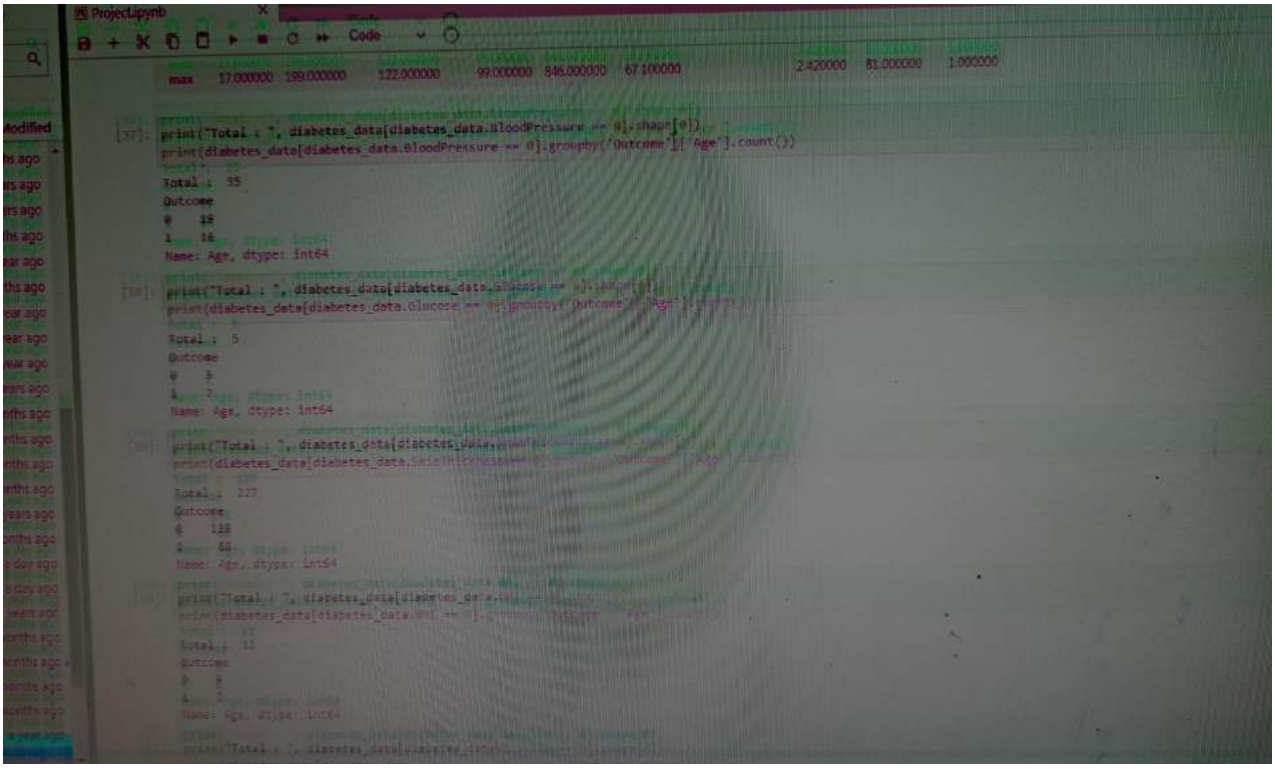
CLEANING DATA:

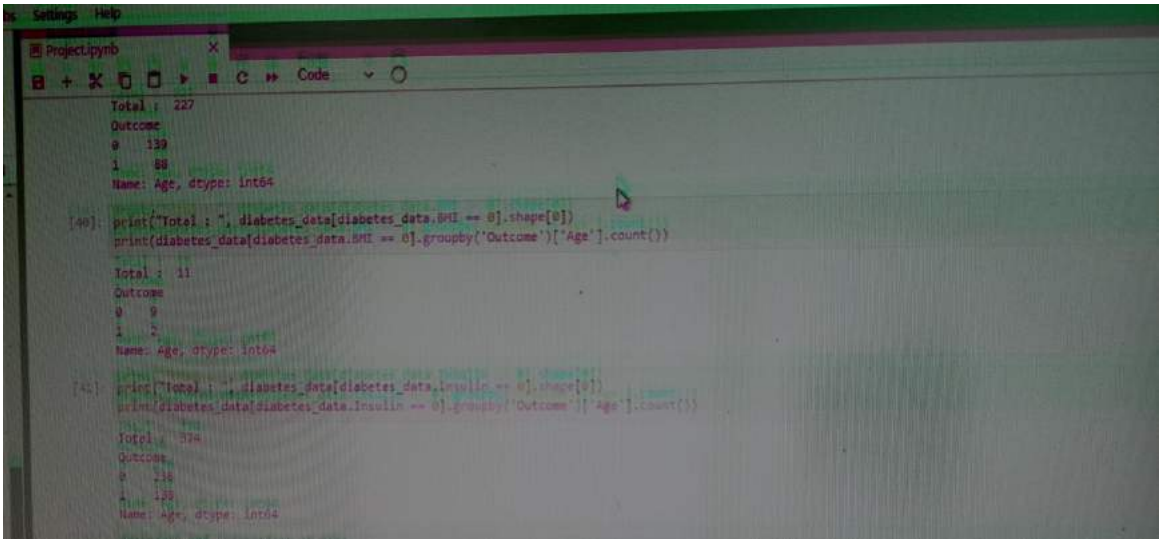




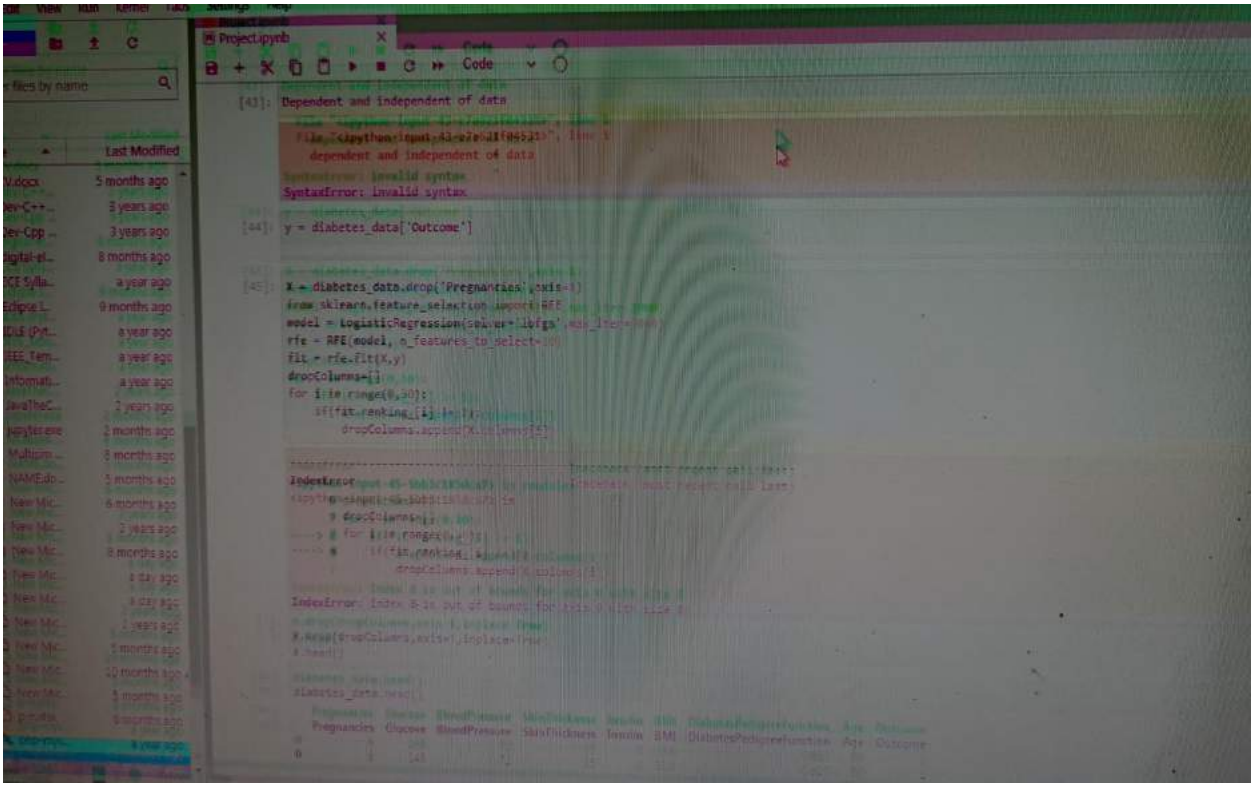


CLEANING DATA:

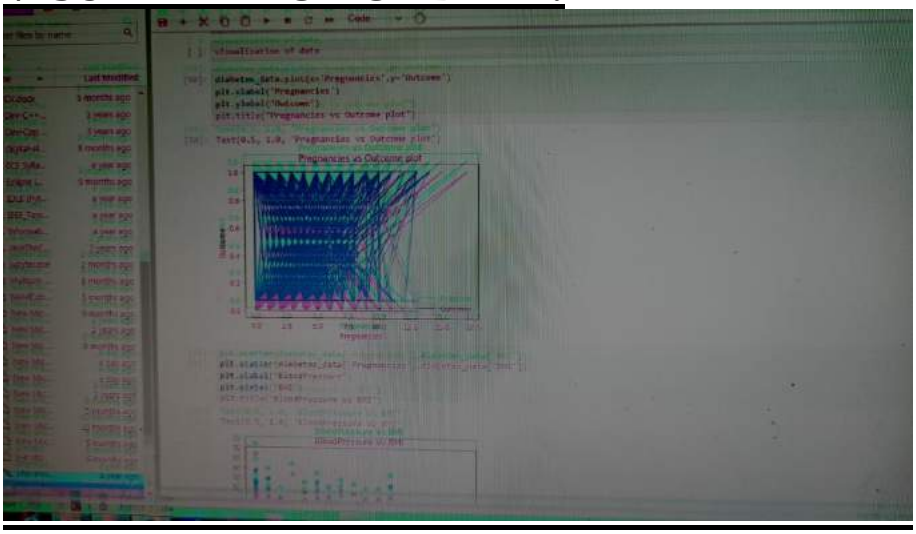


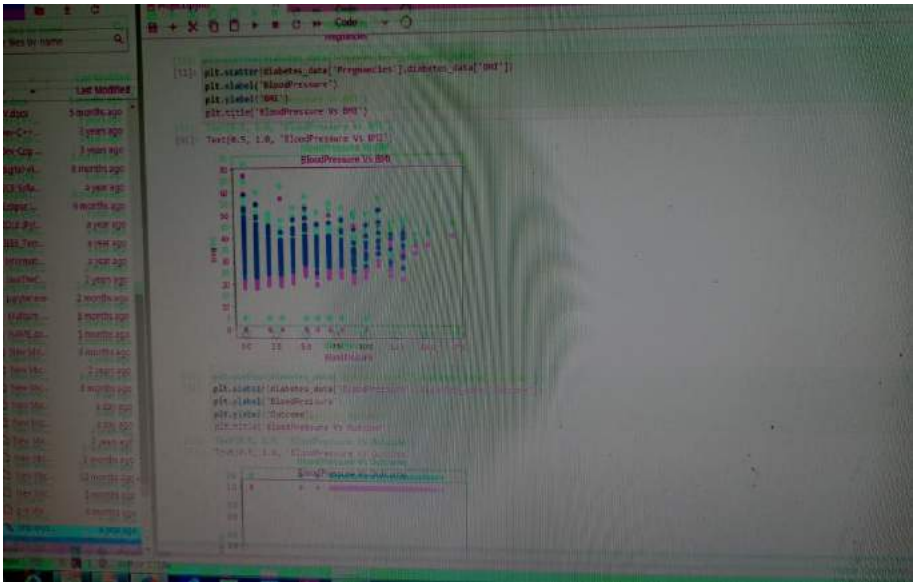


DEPENDENT AND INDEPENDENT OF DATA :

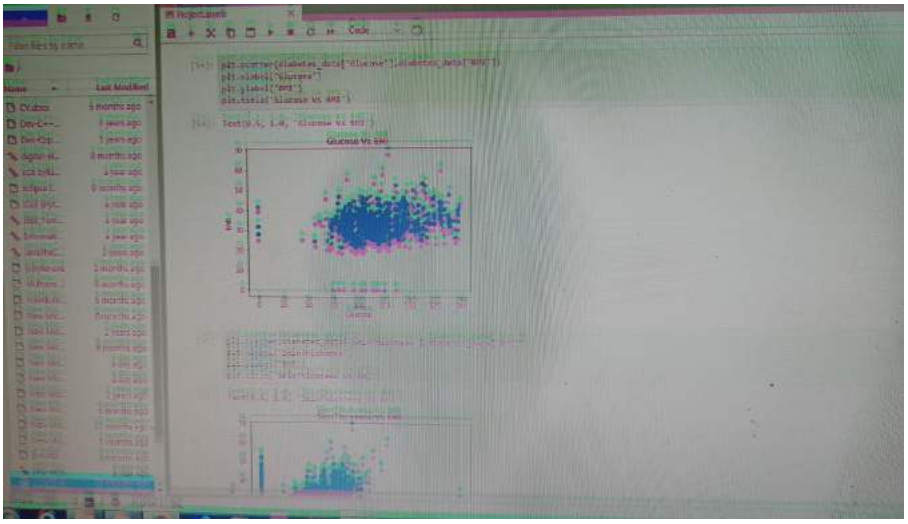


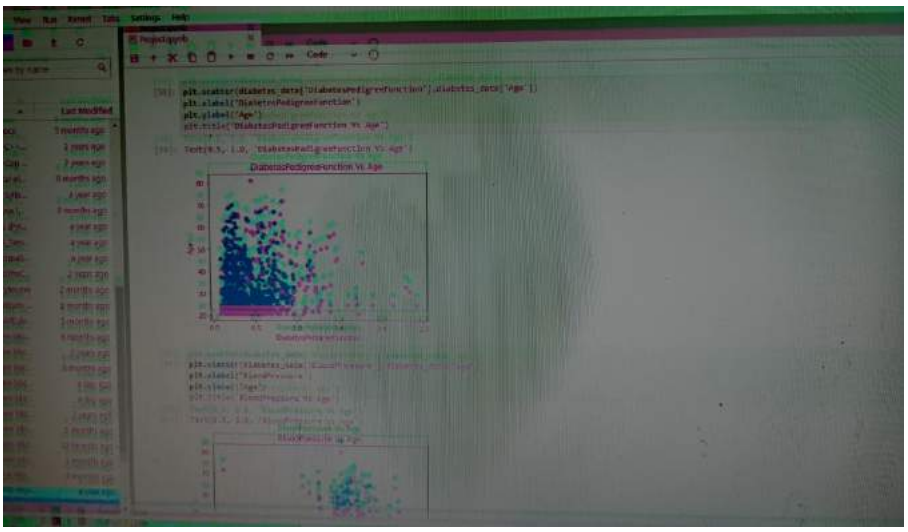
VISUALIZATION OF DATA:



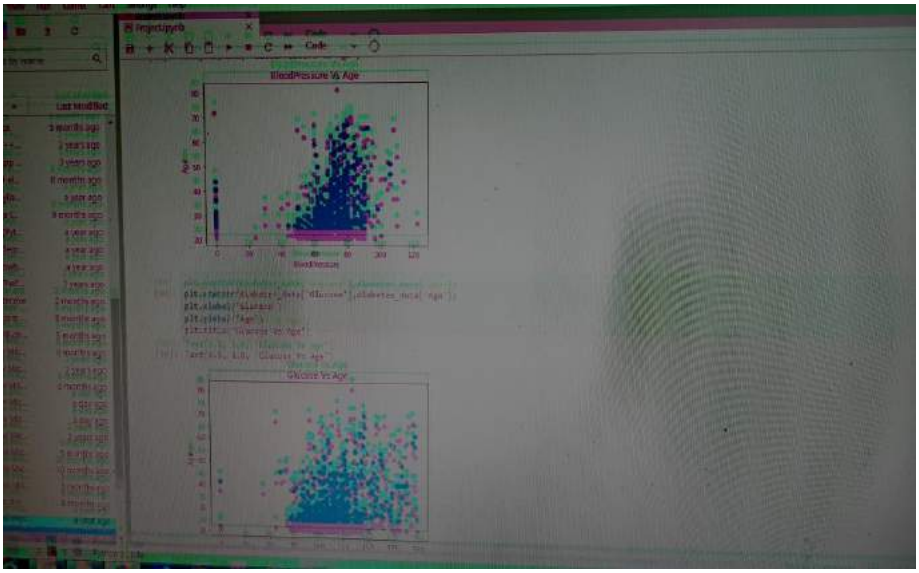


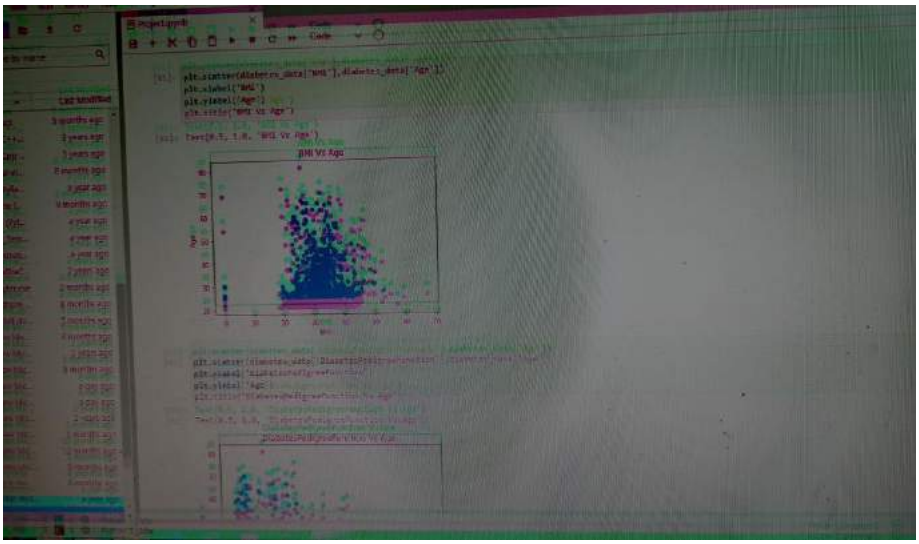
VISUALIZATION OF DATA:

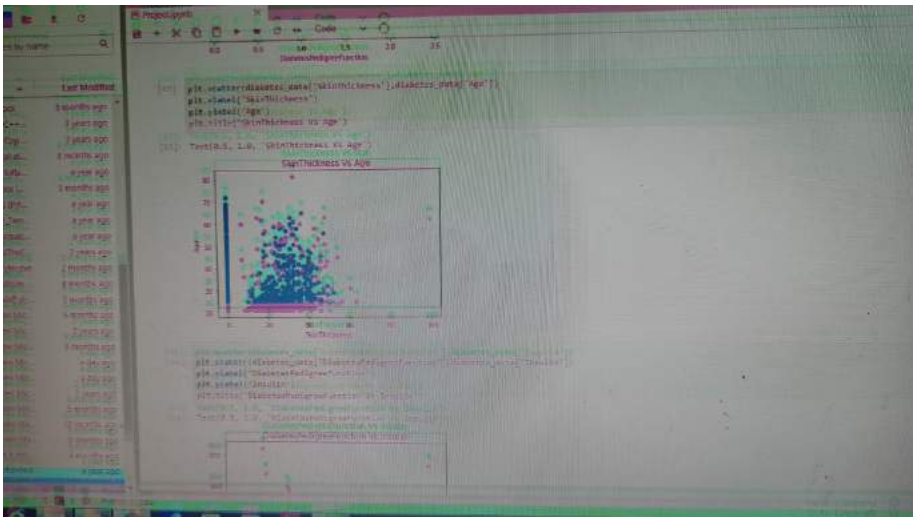




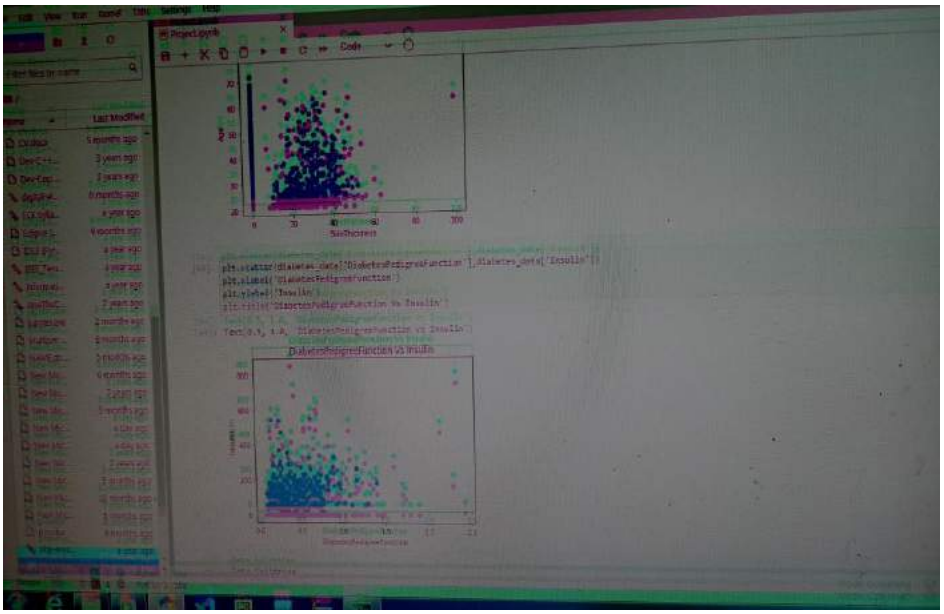
VISUALIZATION OF DATA:



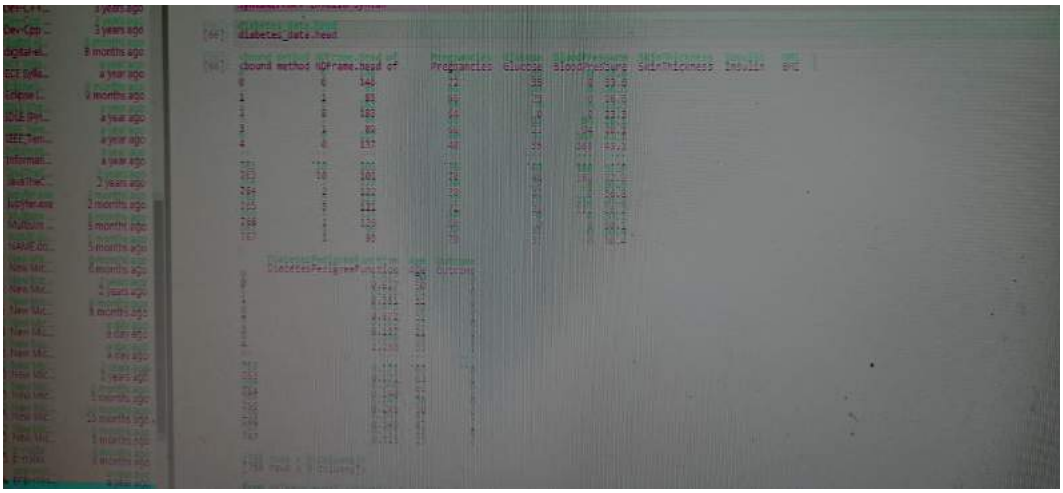




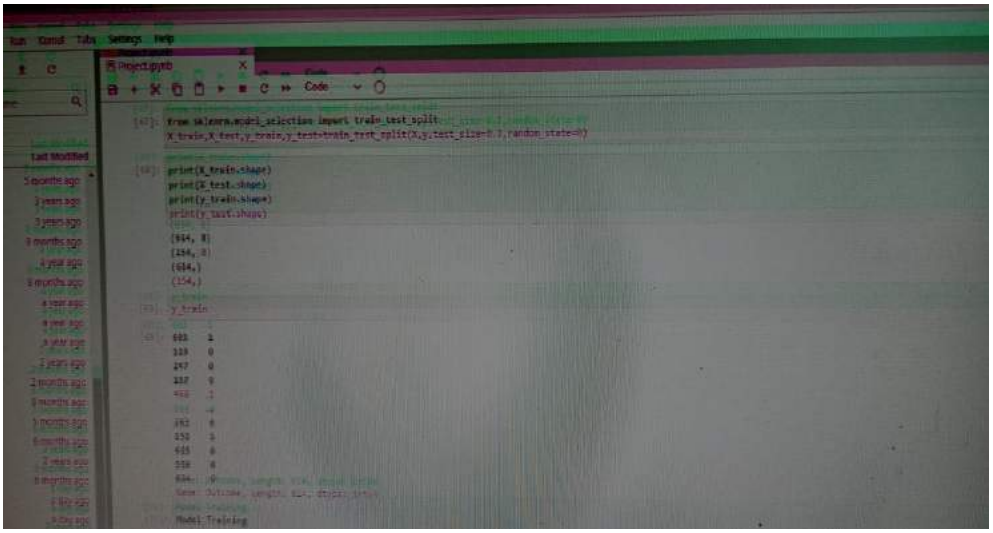
VISUALIZATION OF DATA:



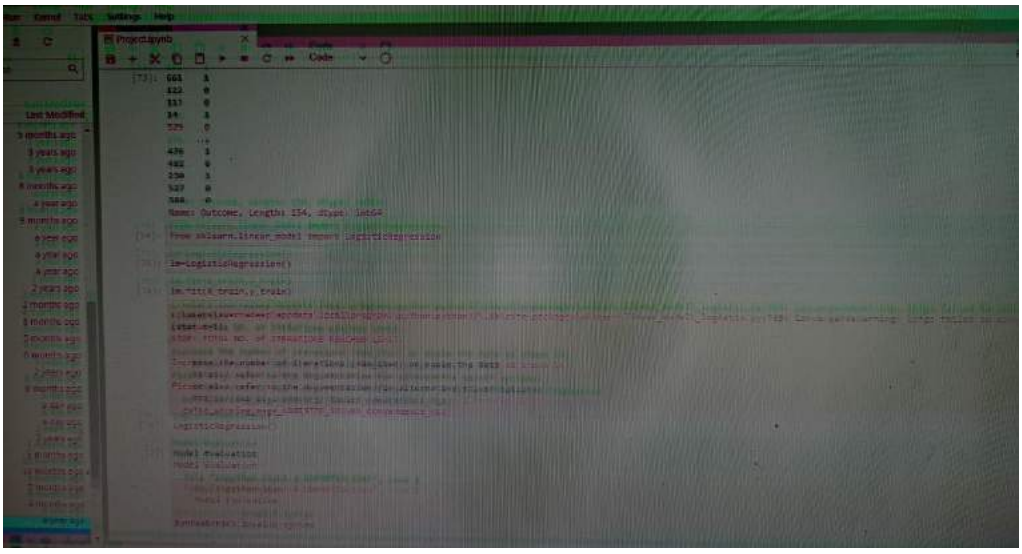
SPLITTING THE DATA:



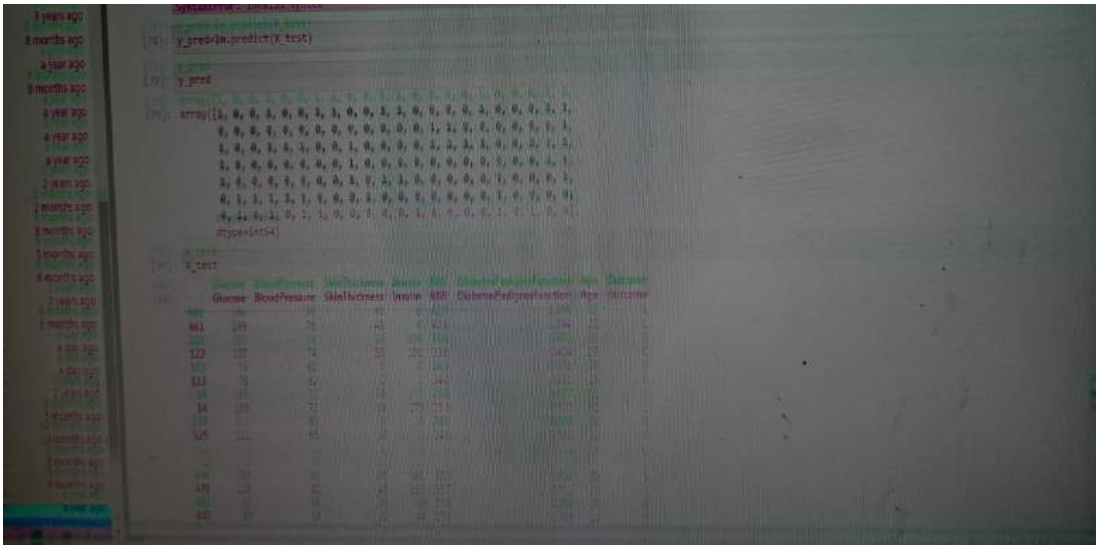
SPLITTING THE DATA:

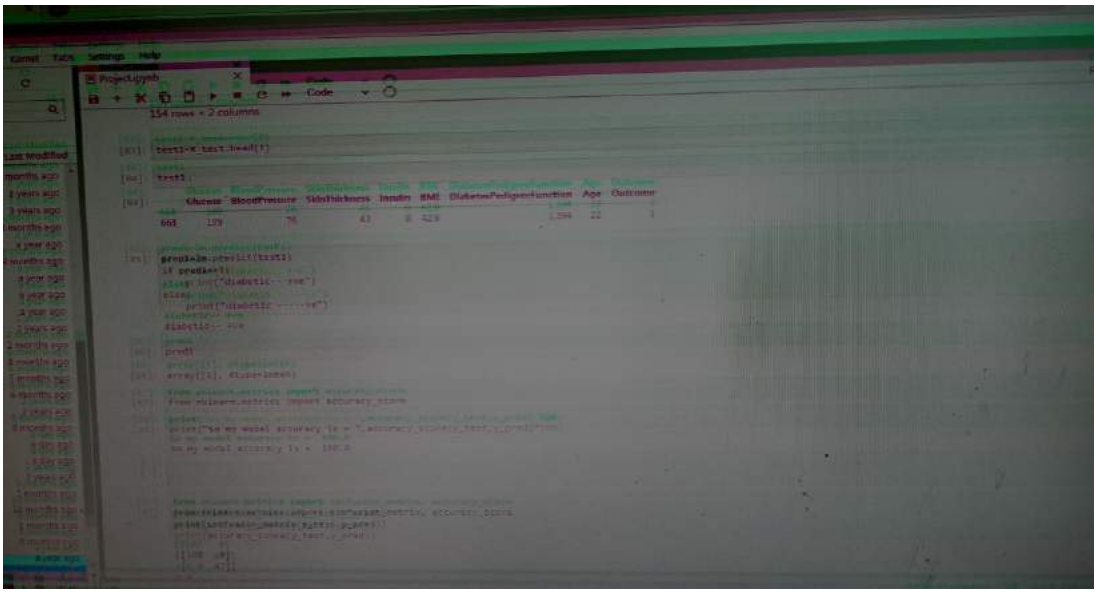


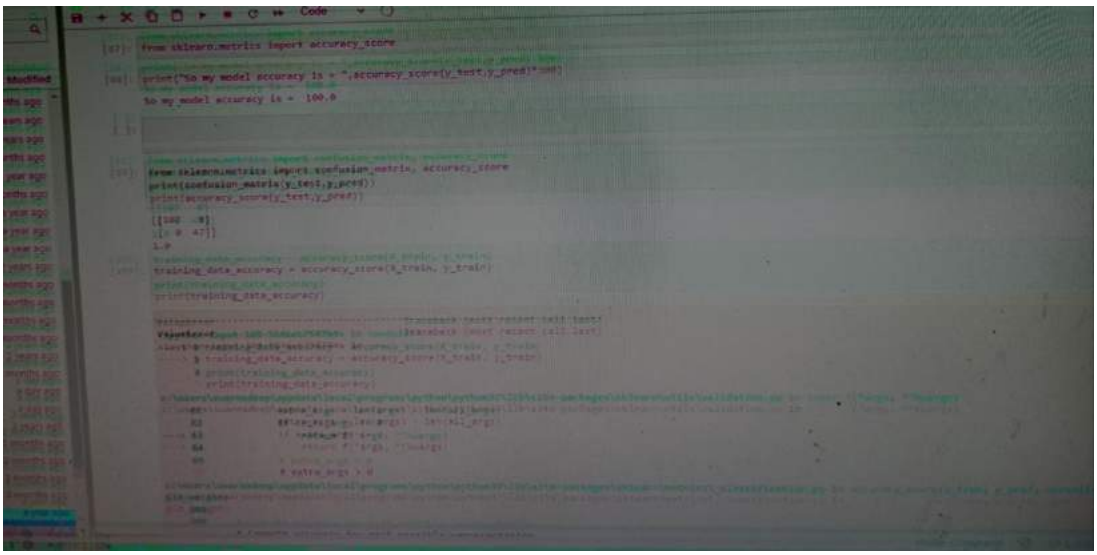
MODEL TRAINING:



MODEL EVALUATION:







FUTURE SCOPE

Proposed system uses “Logistic regression” to find the diabetes disease, in data science we have many algorithms for classification such as Naive Bayes, SVM, Decision Tree, ID3 etc… in future we can add more algorithms to find outputs and algorithms can be compared to find the efficient algorithm. We can add visitor query module, where visitors can post queries to administrator and admin can send reply to those queries. We can add treatment module, where doctors upload treatment details for patients and patient can view those treatment details.

SUMMARY

Logistic Regression is a powerful Machine Learning tool, and we can use it successfully for predicting categorical outputs of biomedical data. Data wrangling and data mining can benefit from excellent performances offered by Python and its libraries so well supported by the community. Linear Algebra programming has intrinsic advantages in avoiding, where possible, ‘while’ and ‘for’ loops. It is implementable by NumPy, a package that vectorizes the matrixes. NumPy makes working on them more comfortable, and guarantees better control over the operations, especially for large arrays.

Moreover, the Machine Learning scenario with Python is enriched by the presence of many powerful packages (i.e., Scikit-learn,) which provide excellently optimized classifications and predictions on data.

The Data Set, with its patients and features, offers an exhaustive assortment of parameters for classification and for this reason represents a perfect example for Machine Learning applications. Anyway, many of these features seem to be redundant, and a definite impact on classification and prediction by some of them remains still unknown.

CODE

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt %matplotlib inline

import seaborn as sns

from sklearn.linear_model
importLogisticRegression

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score,
confusion_matrix, classification_report


diabetes_data=pd.read_csv('C:\\Users\\swarnadeep\\Desktop\V
at traning on python on ML and DA\\diabetes2.csv')

diabetes_data.head

diabetes_data

diabetes_data.head(n=11)

diabetes_data.info()

sns.countplot(x='BMI',data=diabetes_data)
```

```
diabetes_data.isnull().sum()

diabetes_data.isna().sum()

 sns.heatmap(diabetes_data.isnull())

 diabetes_data.head(2)

sns.countplot(x='Outcome',data=diabetes_data)

print("The total number of diabetic patatient is
",len(diabetes_data.index))

diabetes_data.drop('Pregnancies',axis=1,inplace = True)

diabetes_data.head(n=1)

diabetes_data.SkinThickness.unique()


diabetes_data=pd.read_csv('C:\\Users\\swarnadeep\\Desktop\V
at traning on python on ML and DA\\diabetes2.csv')

 diabetes_data


diabetes_data.describe()

print("Total : ", diabetes_data[diabetes_data.BloodPressure ==
0].shape[0])

print(diabetes_data[diabetes_data.BloodPressure ==
0].groupby('Outcome')['Age'].count())
```

```
    print("Total : ", diabetes_data[diabetes_data.Glucose ==
0].shape[0])

print(diabetes_data[diabetes_data.Glucose ==
0].groupby('Outcome')['Age'].count())

    print("Total : ", diabetes_data[diabetes_data.SkinThickness
== 0].shape[0])

print(diabetes_data[diabetes_data.SkinThickness ==
0].groupby('Outcome')['Age'].count())

print("Total : ", diabetes_data[diabetes_data.BMI == 0].shape[0])

print(diabetes_data[diabetes_data.BMI ==
0].groupby('Outcome')['Age'].count())

print("Total : ", diabetes_data[diabetes_data.Insulin ==
0].shape[0])

print(diabetes_data[diabetes_data.Insulin ==
0].groupby('Outcome')['Age'].count())
```

Dependent and independent of data

```
y = diabetes_data['Outcome']

X.drop(dropColumns,axis=1,inplace=True)

X.head()

diabetes_data.head()

diabetes_data.head(0)

y = diabetes_data['Outcome']
```

```
y.head()
```

Visualization of data

```
diabetes_data.plot(x='Pregnancies',y='Outcome')
plt.xlabel('Pregnancies')
plt.ylabel('Outcome')
plt.title("Pregnancies vs Outcome plot")

plt.scatter(diabetes_data['Pregnancies'],diabetes_data['BMI'])
plt.xlabel('BloodPressure')
plt.ylabel('BMI')
plt.title('BloodPressure Vs BMI')

plt.scatter(diabetes_data['BloodPressure'],diabetes_data['Outcom
e'])
plt.xlabel('BloodPressure')
plt.ylabel('Outcome')
plt.title('BloodPressure Vs Outcome')

plt.scatter(diabetes_data['BloodPressure'],diabetes_data['BMI'])
plt.xlabel('BloodPressure')
```

```
plt.ylabel('BMI')
plt.title('BloodPressure Vs BMI')

plt.scatter(diabetes_data['Glucose'],diabetes_data['BMI'])
plt.xlabel('Glucose')
plt.ylabel('BMI')
plt.title('Glucose Vs BMI')

plt.scatter(diabetes_data['SkinThickness'],diabetes_data['BMI'])
plt.xlabel('SkinThickness')
plt.ylabel('BMI')
plt.title('SkinThickness Vs BMI')

plt.scatter(diabetes_data['Glucose'],diabetes_data['BMI'])
plt.xlabel('Glucose')
plt.ylabel('BMI')
plt.title('Glucose Vs BMI')
```

```
plt.scatter(diabetes_data['DiabetesPedigreeFunction'],diabetes_d
ata['BMI'])

plt.xlabel('DiabetesPedigreeFunction')

plt.ylabel('BMI')

plt.title('DiabetesPedigreeFunction Vs BMI')


plt.scatter(diabetes_data['DiabetesPedigreeFunction'],diabetes_d
ata['Age'])

plt.xlabel('DiabetesPedigreeFunction')

plt.ylabel('Age')

plt.title('DiabetesPedigreeFunction Vs Age')

plt.scatter(diabetes_data['BloodPressure'],diabetes_data['Age'])

plt.xlabel('BloodPressure')

plt.ylabel('Age')

plt.title('BloodPressure Vs Age')

plt.scatter(diabetes_data['Glucose'],diabetes_data['Age'])

plt.xlabel('Glucose')

plt.ylabel('Age')

plt.title('Glucose Vs Age')
```

```
plt.scatter(diabetes_data['BMI'],diabetes_data['Age'])

plt.xlabel('BMI')

plt.ylabel('Age')

plt.title('BMI Vs Age')


plt.scatter(diabetes_data['DiabetesPedigreeFunction'],diabetes_d
ata['Age'])

plt.xlabel('DiabetesPedigreeFunction')

plt.ylabel('Age')

plt.title('DiabetesPedigreeFunction Vs Age')


plt.scatter(diabetes_data['SkinThickness'],diabetes_data['Age'])

plt.xlabel('SkinThickness')

plt.ylabel('Age')

plt.title('SkinThickness Vs Age')


plt.scatter(diabetes_data['DiabetesPedigreeFunction'],diabetes_d
ata['Insulin'])

plt.xlabel('DiabetesPedigreeFunction')

plt.ylabel('Insulin')

plt.title('DiabetesPedigreeFunction Vs Insulin')
```

Data splitting

```
diabetes_data.head

from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,rand
om_state=0)

print(X_train.shape)

print(X_test.shape)

print(y_train.shape)

print(y_test.shape)

y_train

from sklearn.metrics import confusion_matrix, accuracy_score

print(confusion_matrix(y_test,y_pred))

print(accuracy_score(y_test,y_pred))
```

Model training

```
[  from sklearn.linear_model import LinearRegression

lm=LinearRegression()

lm.fit(X_train,y_train) ]  For linear regerssion

from sklearn.linear_model import LogisticRegression

lm=LogisticRegression()

lm.fit(X_train,y_train)
```

Data prediction/model evaluation

```
y_pred=lm.predict(X_test)
y_pred
X_test
df=pd.DataFrame({'Actual':y_test,'MC predicted':y_pred})
df
test1=X_test.head(1)
test1
pred1=lm.predict(test1)
if pred1==1:
   print("diabetic-- +ve")
else:
   print("diabetic -----ve")

pred1
from sklearn.metrics import accuracy_score
print("So my model accuracy is =
",accuracy_score(y_test,y_pred)*100)
```

Data prediction/model evaluation

```
y_pred=lm.predict(X_test)
y_pred
X_test
df=pd.DataFrame({'Actual':y_test,'MC predicted':y_pred})
df
test1=X_test.head(1)
test1
pred1=lm.predict(test1)
if pred1==1:
   print("diabetic-- +ve")
else:
   print("diabetic -----ve")
pred1
from sklearn.metrics import accuracy_score
print("So my model accuracy is =
",accuracy_score(y_test,y_pred)*100)
```