PREDICTING THE DIABETES USING MACHINE LEARNING

INTERNAL GUIDE

Ms. D. LAKSHMI ROHITHA

TEAM MEMBERS

M INDU (18AG1A0534)

CH KAVYA(18AG1A0511)

P RATHNAMALA(18AG1A0544)

CONTENTS

- ABSTRACT
- INTRODUCTION
- LITERATURE SURVEY
- EXISTING SYSTEM
- PROPOSED MODEL
- SYSTEM ARCHITECTURE
- WORK FLOW

ABSTRACT

- Diabetes is an illness caused because of high glucose level in a human body.
- Diabetes should not be ignored if it is untreated then diabetes may cause some major issues in a person like : heart related problems, kidney problem, blood pressure, eye damage and it can also affects other organs of human body.
- To achieve this goal this project work we will do early prediction of diabetes in a human body or a patient for a higher accuracy through applying, Various Machine Learning Techniques.
- In this work we can use Machine Learning Classification and ensemble techniques on a dataset to predict diabetes. Which are K-Nearest Neighbor, Linear Regression , Logistic Regression, Decision Tree, Support Vector Machine, Random Forest. The accuracy is different for every model when compared to other models.

INTRODUCTION

- Diabetes occurs when body does not make enough insulin.
- According to World Health Organization about 422 million people suffering from diabetes. However prevalence of diabetes is found among various Countries like Canada, China and India etc.
- This work explores prediction of diabetes by taking various attributes related to diabetes disease. For this purpose we use the Pima Indian Diabetes Dataset, we apply various machine learning classification and ensemble Techniques to predict diabetes.
- Various Techniques of Machine Learning can capable to do prediction, however its tough to choose best technique.
- Thus for this purpose we apply popular classification and ensemble methods on dataset for prediction.

PROBLEM AND PURPOSE STATEMENT

Problem :

- Diabetes has become a global problem that the manual decisions of doctor' s can be alarming.
- It can be difficult to determine whether a patient having diabetes or not with better accuracy.

Purpose :

- This project aims to develop a model for predicting diabetes using machine learning algorithm.
- The main goal is to develop a model that predict whether a patient having diabetes or not.

LITERATURE SURVEY

- Researchers who have been worked prediction on diabetes are:
- In Amour Diwani's et al.'s study, all the patient's data are trained and tested using 10 cross-validations with Naïve Bayes and decision tree. Then the performance was evaluated, investigated, and compared with other classification algorithms. The results predicted that the best algorithm is Naïve Bayes with an accuracy of 65. 3021%.
- In Zou et al.'s study, they applied Random Forest, Decision Tree, ANN for classification algorithm on PIDD after the feature reduction using Principal Component Analysis and Minimum Redundancy. They found that Pima Indian's best accuracy is 77. 21% obtained from the random forest with feature reduction method.

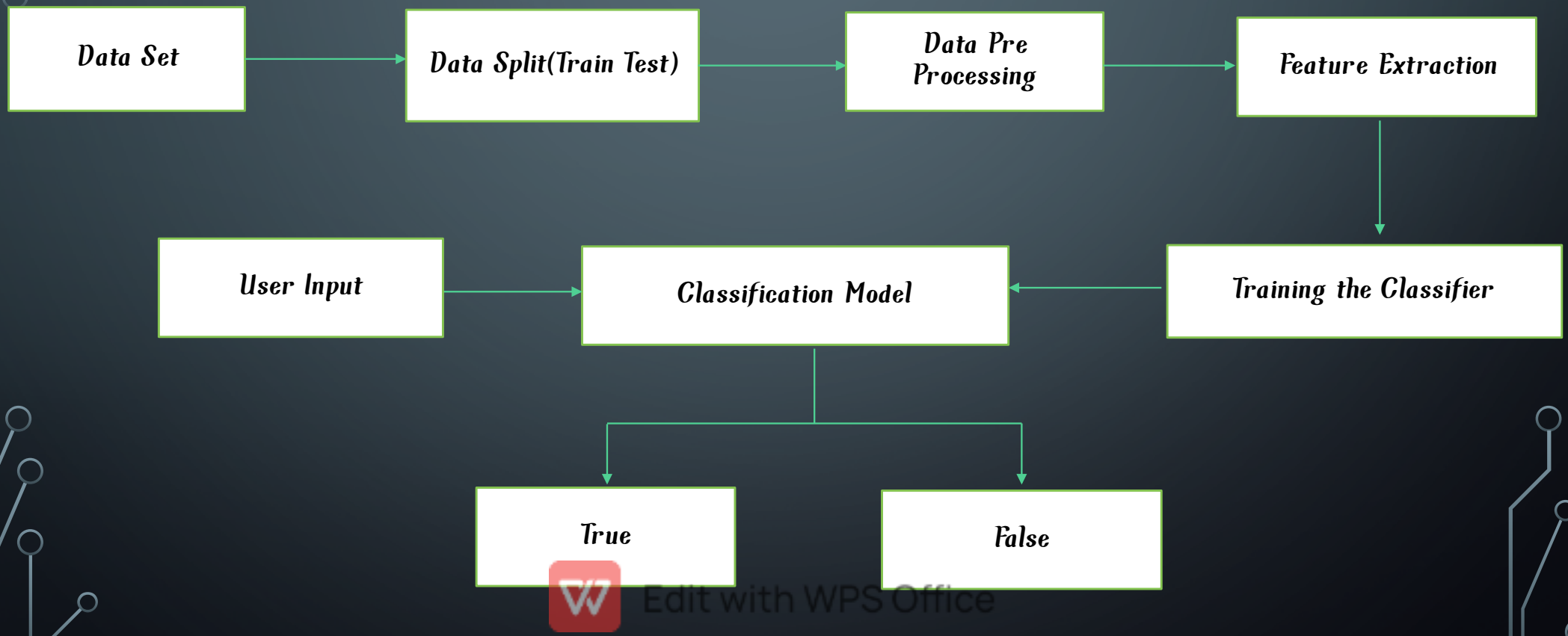
EXISTING SYSTEM

- The healthcare industry collects an enormous amount of data include hospital records, medical records of patients, and results of medical examinations.
- For early disease diagnosis, the disease's prediction is analyzed through a doctor's experience and knowledge, but that can be inaccurate and susceptible.
- Hence the manual decisions can be alarming. The hidden pattern of data can be unnoticed, which can impact decision-making; therefore, patients become deprived of the appropriate treatment.

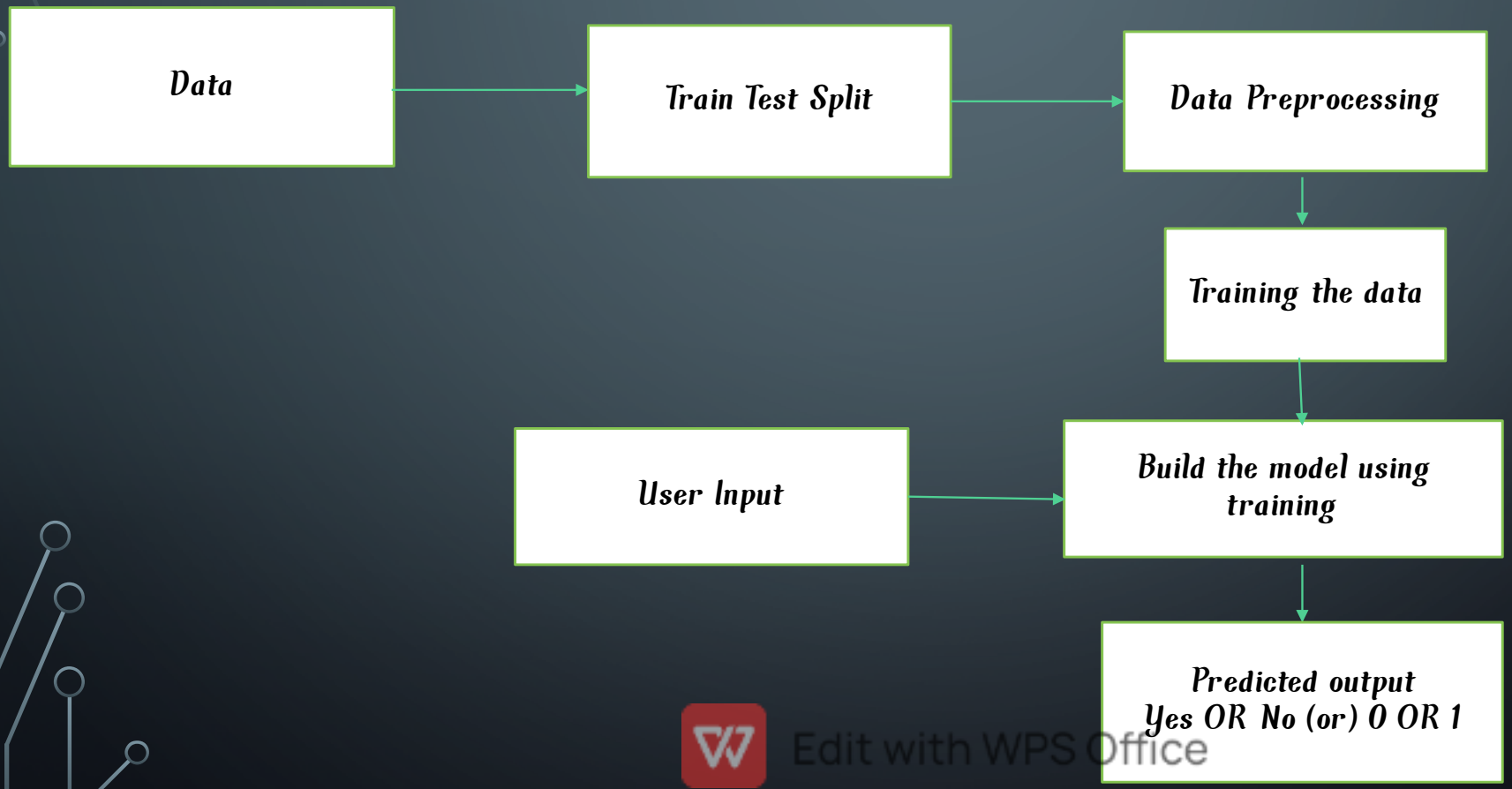
PROPOSED MODEL

- We aim to develop a model that can predict the diabetes . The model is built by implementing the Machine Learning with Linear and Logistic Regression algorithm.
- The developed model show that whether a patient having diabetes or not.

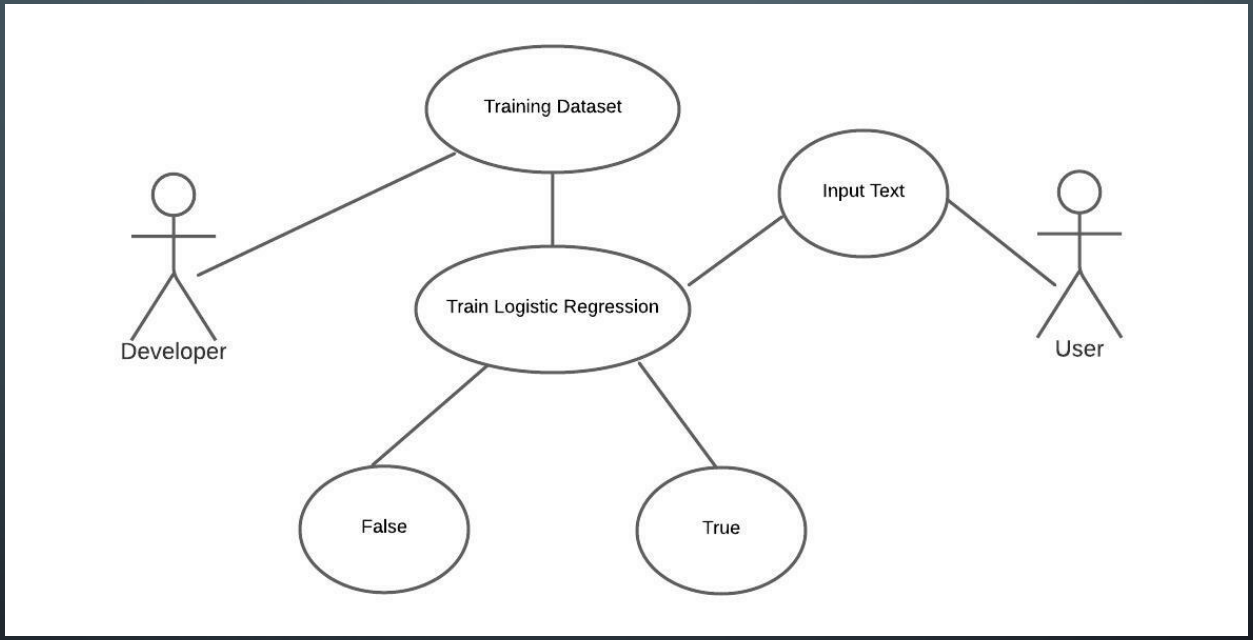
SYSTEM ARCHITECTURE



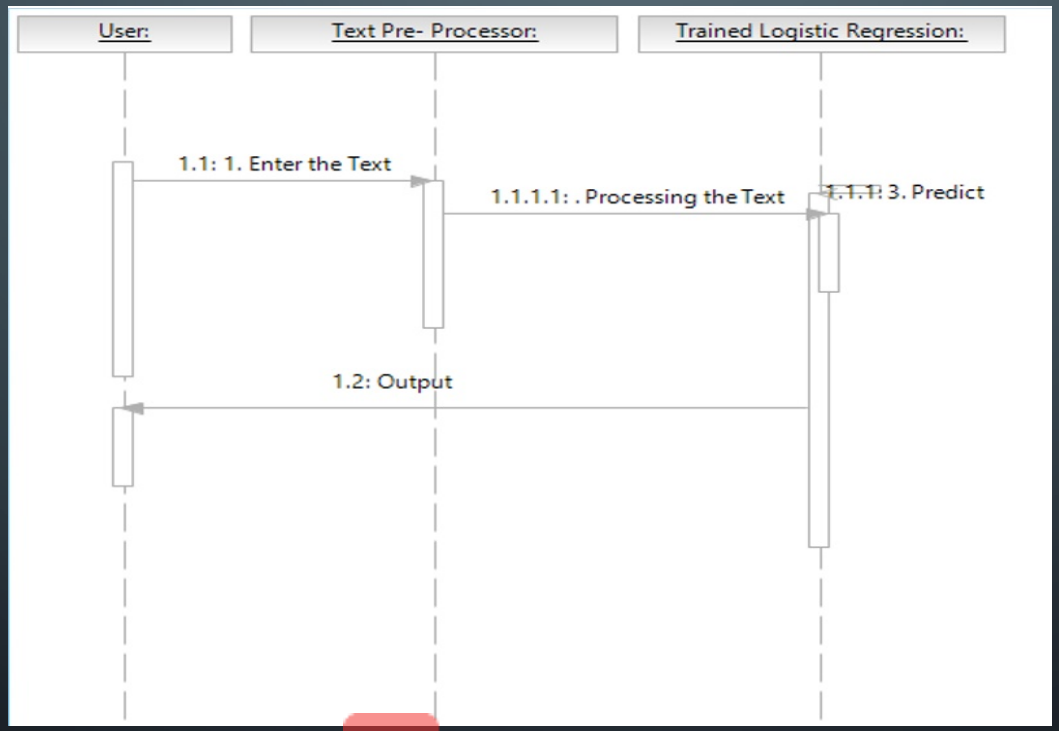
WORK FLOW



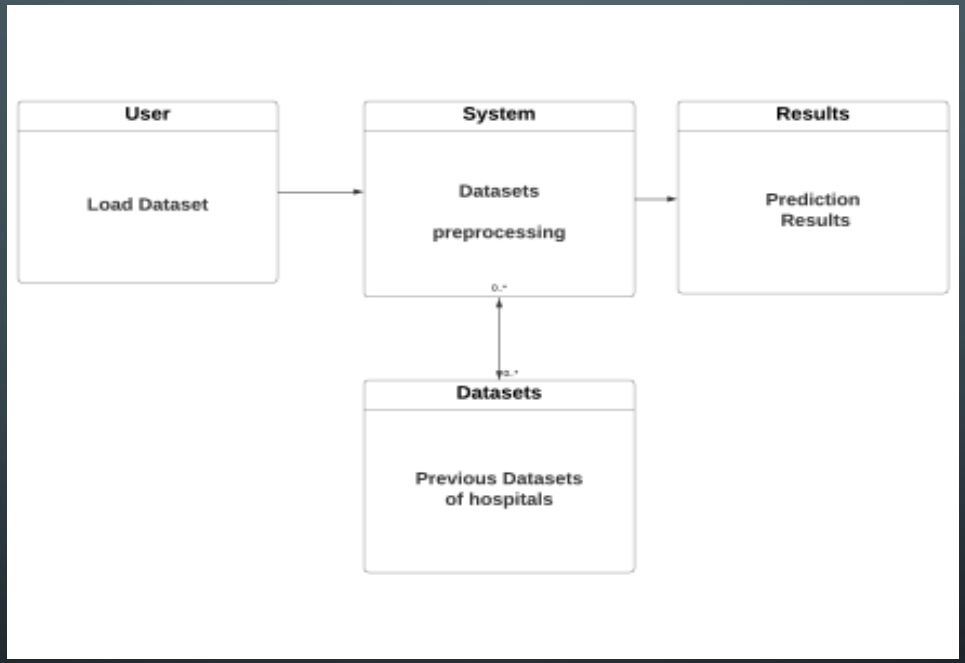
USE CASE DIAGRAM



SEQUENCE DIAGRAM



CLASS DIAGRAM



STATE CHART DIAGRAM

REQUIREMENTS

• SOFTWARE REQUIRMENTS:

Operating system : Windows 10 or MAC OS.

Platform : Google Colab

Programming Language : Python

• HARDWARE REQUIREMENTS:

Processor : Intel core i5 and above

Hard Disk : 512 GB or above

RAM :8GB or above

Internet :4 Mbps or above(Wireless)

LIBRARIES

- *lmport pandas as pd*
- *lmport numpy as np*
- *lmport matplotlib. pyplot as plt*
- *lmport seaborn as sns*
- *from numpy import nan*
- *from sklearn. model_selection import train_test_split*
- *from sklearn. linear_model import LinearRegression*
- *from sklearn import metrics*
- *from sklearn. linear_model import LogisticRegression*
- *from sklearn. metrics import accurarcy_score*

ATTRIBUTES

- Pregnancies : Number of times pregnant
- Glucose : Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- Blood Pressure : Diastolic blood pressure (mm Hg)
- Skin Thickness : Triceps skin fold thickness (mm)
- Insulin : 2-Hour serum insulin (mu U/ml)
- BMI : Body mass index (weight in kg/(height in m)^2)
- Diabetes Pedigree Function : Diabetes pedigree function
- Age : Age (years)
- Outcome : Class variable (0 or 1) 268 of 768 are 1, the others are 0
- Total rows and columns in the dataset

LINEAR REGRESSION

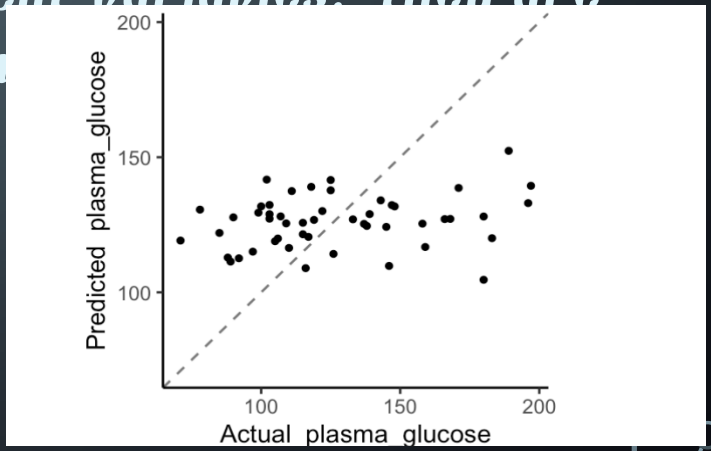
- Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables.
- Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent va
- Linear Regression Equation is Y=a+bX.

  a=constant

  b=regression coefficient

  X is the independent variable,

  Y is known as the predicted value of the dependent variable.



LINEAR REGRESSION

Mean absolute error

Mean squared error

Root mean squared error

```
[ ]  from sklearn import metrics
```

```
[ ]  print(metrics.mean_absolute_error(y_test, predictions))
```

```
0.3530674332857286
```

```
[ ]  print(metrics.mean_squared_error(y_test, predictions))
```

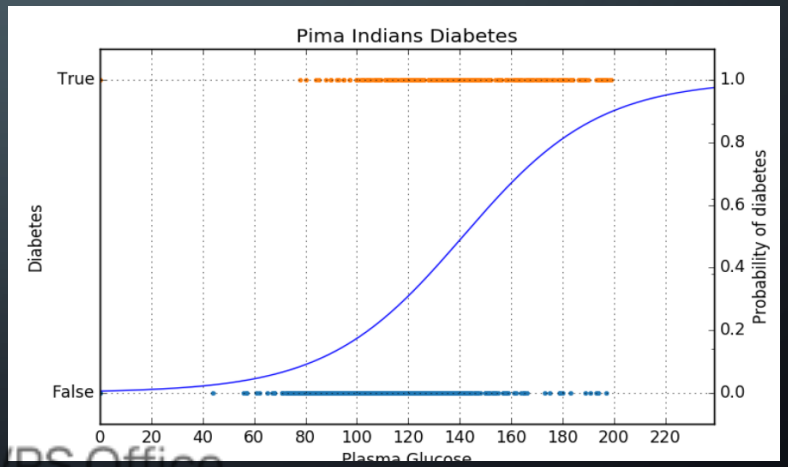
```
0.17856859622487978
```

```
print(np.sqrt(metrics.mean_squared_error(y_test,predictions)))
```

```
0.4225737760733382
```

LOGISTIC REGRESSION

- Logistic regression is a Machine Learning algorithm based on Supervised Learning . It is used for predicting the categorical dependent variable using a given set of independent variables.
- The outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, True or False.



LOGISTIC REGRESSION

```
[ ]  y_data=data['Outcome']
     x_data=data.drop('Outcome',axis=1)
```

```
[ ]  x_train,x_test,y_train,y_test=train_test_split(x_data,y_data,test_size=0.3,stratify=y_data,random_state=5)
```

```
[ ]  model=LogisticRegression()
```

```
[ ]  model.fit(x_train,y_train)
```

```
[ ]  predictions=model.predict(x_train)
```

```
[ ]  train_data_accuracy=accuracy_score(predictions,y_train)
```

```
[ ]  print("accuracy score of the training data:",train_data_accuracy)

     accuracy score of the training data: 0.7970204841713222
```

RANDOM FOREST

- Random Forest is a popular machine learning algorithm .
- It belongs to the supervised learning technique made up of decision tree.
- It can be used for both Classification and Regression problems in ML.

RANDOM FOREST

RANDOM FOREST

```
from sklearn.ensemble import RandomForestClassifier
RF_Classifier=RandomForestClassifier()
RF_Classifier.fit(x_train,y_train)
```

```
[ ]  ypred_RF= RF_Classifier.predict(x_test)
```

```
from sklearn.metrics import confusion_matrix,accuracy_score,classification_report,roc_auc_score
cm_rf=confusion_matrix(y_test,ypred_RF)
cm_rf
```

```
array([[128,  29],
       [ 27,  47]])
```

```
[ ]  score_rf=accuracy_score(y_test,ypred_RF)
     print('accuracy based on RandomForest model',score_rf)
```

```
accuracy based on RandomForest model 0.7575757575757576
```

SAMPLE INPUT

```
input_data=(1,120.0,50.0,30.0,100.0,55.1,0.177,31)
input_data_as_numpy_array=np.asarray(input_data)
input_data_reshaped=input_data_as_numpy_array.reshape(1,-1)
prediction=model.predict(input_data_reshaped)
print(prediction)
if (prediction[0]==0):
  print("The person does not have diabetes")
else:
   print("The person have diabete")
```

```
[1]
The person have diabete
```

SAMPLE INPUT

SAMPLE INPUT(2)

```
input_data=(1,89.0,66.0,23.0,94.0,28.1,0.167,21)
input_data_as_numpy_array=np.asarray(input_data)
input_data_reshaped=input_data_as_numpy_array.reshape(1,-1)
prediction=model.predict(input_data_reshaped)
print(prediction)
if (prediction[0]==0):
  print("The person does not have diabetes")
else:
   print("The person have diabete")
```

```
[0]
The person does not have diabetes
```

REFERENCES

- https://www.kaggle.com/uciml/pima-indians-diabetes-database
- www.Kaggle.com
- www.ijert.org