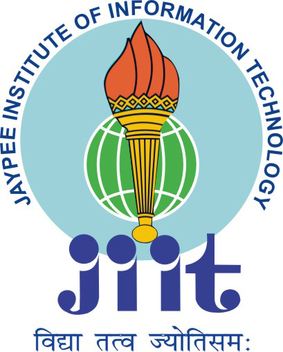
**DIABETES PREDICTION**

**MINI PROJECT**

**(FUNDAMENTAL OF MACHINE LEARNING)**

Submitted By:

**Bhavya Varshney (9919103142)**

****

**Department of CSE/IT**

**Jaypee Institute of Information Technology University, Noida**

**DECEMBER 2021**

**ABSTRACT**

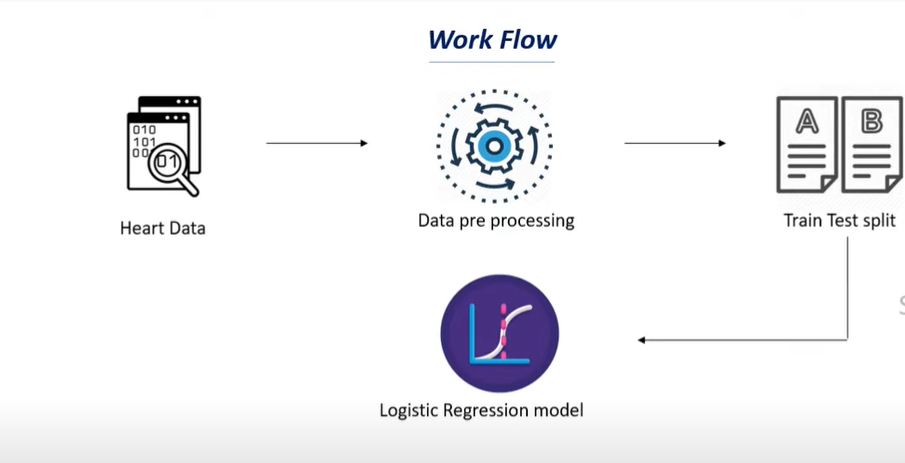
Diabetes mellitus is a chronic disease characterized by hyperglycemia. It may cause many complications. According to the growing morbidity in recent years, in 2040, the world’s diabetic patients will reach 642 million, which means that one of the ten adults in the future is suffering from diabetes. There is no doubt that this alarming figure needs great attention. With the rapid development of machine learning, machine learning has been applied to many aspects of medical health. In this study, we used decision tree, random forest and neural network to predict diabetes mellitus. The dataset is the Dropbox. It contains 9 attributes. In this study, five-fold cross validation was used to examine the models. In order to verity the universal applicability of the methods, we chose some methods that have the better performance to conduct independent test experiments. We randomly selected 768 healthy people and diabetic patients’ data, respectively as training set. In this person having diabetes is represented by 1 anis represented by 0.

**INTRODUCTION**

Diabetes mellitus is one of the non-communicable diseases that pose a threat to human health. It has become a major global health problem. It is a chronic disease that occurs either when the pancreas does not produce enough insulin or when the body cannot effectively use the insulin which it produces. It is found that diabetes causes blindness, amputation and kidney failure. Lack of awareness about diabetes, insufficient access to health services and essential medicines can lead to the above mentioned complications. According to a study by the World Health Organization (WHO), number of diabetic patients will raise to 552 million by 2030, which means that one in 10 adults will have diabetes by 2030. In 2014, the global prevalence of diabetes was estimated to be 9 % among adults aged 18+ years [1]. WHO insisted with an alarm that Diabetes is the 7th leading cause of death in the world. In 2012, an estimated 1.5 million deaths were directly caused by diabetes. Total deaths due to diabetes are projected to rise by more than 50 % in the next 10 years. Moreover, the International Diabetes Federation said that nearly 52 % of Indians are not aware that they are suffering from high blood sugar. More than 62 million diabetic individuals are currently diagnosed with the disease. It is predicted that, by 2030 diabetes mellitus may affect up to 79.4 million individuals in India. A nation-wide study, conducted by the Indian Council of Medical Research`s INDIAB (India Diabetes) has confirmed that one out of 10 people in Tamil Nadu is affected by diabetes, and every two persons with age group of 25 are in the pre-diabetic stage. It is stated that 14.8 per cent of urban population and 11 per cent of rural population of Tamil Nadu are affected by diabetes. Madras Diabetes Research Foundation suggested that about 42 lakh individuals have diabetes and 30 lakh people are in pre-diabetes stage. At least, 1,000 people avail treatment for diabetes out of the 12,000 outpatients who visit Rajiv Gandhi Government General Hospital (RGGGH), a leading Government hospital in Chennai.

**BACKGROUND STUDY**

Diabetes affects millions of people, and it remains the chief cause of death in the world. Medical diagnosis should be proficient, reliable, and aided with computer techniques to reduce the effective cost for diagnostic tests. Machine Learning helps computers to build and classify various attributes. This project uses classification techniques to predict Diabetes disease.

**ALGORITHM**

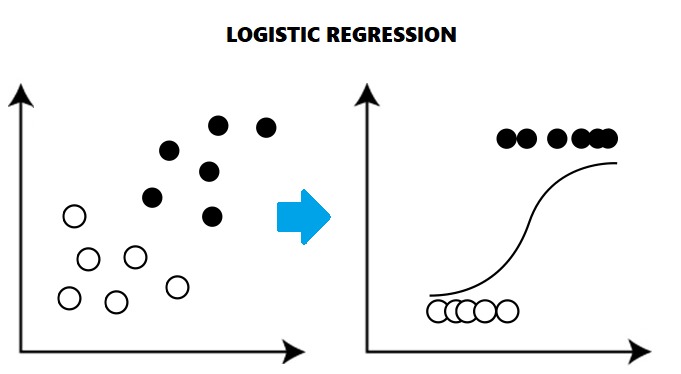
Diabetes can be certainly predicted by the aid of artificial intelligence which includes machine learning algorithms. The algorithm used in this project to propose diabetes prediction model is Logistic Regression.

**Logistic Regression**

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).

Binary Logistic Regression Model − The simplest form of logistic regression is binary or binomial logistic regression in which the target or dependent variable can have only 2 possible types either 1 or 0.



**import** numpy **as** np

**import** pandas **as** pd

**from** sklearn.preprocessing **import** StandardScaler

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.linear\_model **import** LogisticRegression

**from** sklearn.metrics **import** accuracy\_score

**IMPLEMENTATION**

# Data Collection and Analysis

PIMA Diabetes Dataset

In [3]:

*# loading the diabetes dataset to a pandas DataFrame*

diabetes\_dataset **=** pd**.**read\_csv(r"C:\Users\bhavya\Downloads\diabetes.csv")

In [4]:

pd.read\_csv**?**

In [5]:

*# printing the first 5 rows of the dataset*

diabetes\_dataset**.**head()

Out[5]:

|  | **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 |

In [6]:

*# number of rows and Columns in this dataset*

diabetes\_dataset**.**shape

Out[6]:

(768, 9)

In [7]:

*# getting the statistical measures of the data*

diabetes\_dataset**.**describe()

Out[7]:

|  | **Pregnancies** | **Glucose** | **BloodPressure** | **SkinThickness** | **Insulin** | **BMI** | **DiabetesPedigreeFunction** | **Age** | **Outcome** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 |
| **mean** | 3.845052 | 120.894531 | 69.105469 | 20.536458 | 79.799479 | 31.992578 | 0.471876 | 33.240885 | 0.348958 |
| **std** | 3.369578 | 31.972618 | 19.355807 | 15.952218 | 115.244002 | 7.884160 | 0.331329 | 11.760232 | 0.476951 |
| **min** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.078000 | 21.000000 | 0.000000 |
| **25%** | 1.000000 | 99.000000 | 62.000000 | 0.000000 | 0.000000 | 27.300000 | 0.243750 | 24.000000 | 0.000000 |
| **50%** | 3.000000 | 117.000000 | 72.000000 | 23.000000 | 30.500000 | 32.000000 | 0.372500 | 29.000000 | 0.000000 |
| **75%** | 6.000000 | 140.250000 | 80.000000 | 32.000000 | 127.250000 | 36.600000 | 0.626250 | 41.000000 | 1.000000 |
| **max** | 17.000000 | 199.000000 | 122.000000 | 99.000000 | 846.000000 | 67.100000 | 2.420000 | 81.000000 | 1.000000 |

In [8]:

diabetes\_dataset['Outcome']**.**value\_counts()

Out[8]:

0 500

1 268

Name: Outcome, dtype: int64

0 Represents Non-Diabetic

1 Represents Diabetic

In [9]:

diabetes\_dataset**.**groupby('Outcome')**.**mean()

Out[9]:

|  | **Pregnancies** | **Glucose** | **BloodPressure** | **SkinThickness** | **Insulin** | **BMI** | **DiabetesPedigreeFunction** | **Age** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Outcome** |  |  |  |  |  |  |  |  |
| **0** | 3.298000 | 109.980000 | 68.184000 | 19.664000 | 68.792000 | 30.304200 | 0.429734 | 31.190000 |
| **1** | 4.865672 | 141.257463 | 70.824627 | 22.164179 | 100.335821 | 35.142537 | 0.550500 | 37.067164 |

In [10]:

*# separating the data and labels*

X **=** diabetes\_dataset**.**drop(columns **=** 'Outcome', axis**=**1)

Y **=** diabetes\_dataset['Outcome']

In [11]:

print(X)

Pregnancies Glucose BloodPressure SkinThickness Insulin BMI \

0 6 148 72 35 0 33.6

1 1 85 66 29 0 26.6

2 8 183 64 0 0 23.3

3 1 89 66 23 94 28.1

4 0 137 40 35 168 43.1

.. ... ... ... ... ... ...

763 10 101 76 48 180 32.9

764 2 122 70 27 0 36.8

765 5 121 72 23 112 26.2

766 1 126 60 0 0 30.1

767 1 93 70 31 0 30.4

DiabetesPedigreeFunction Age

0 0.627 50

1 0.351 31

2 0.672 32

3 0.167 21

4 2.288 33

.. ... ...

763 0.171 63

764 0.340 27

765 0.245 30

766 0.349 47

767 0.315 23

[768 rows x 8 columns]

In [12]:

print(Y)

0 1

1 0

2 1

3 0

4 1

..

763 0

764 0

765 0

766 1

767 0

Name: Outcome, Length: 768, dtype: int64

# Data Standardization

In [13]:

scaler **=** StandardScaler()

In [14]:

scaler**.**fit(X)

Out[14]:

StandardScaler()

In [15]:

standardized\_data **=** scaler**.**transform(X)

In [16]:

print(standardized\_data)

[[ 0.63994726 0.84832379 0.14964075 ... 0.20401277 0.46849198

1.4259954 ]

[-0.84488505 -1.12339636 -0.16054575 ... -0.68442195 -0.36506078

-0.19067191]

[ 1.23388019 1.94372388 -0.26394125 ... -1.10325546 0.60439732

-0.10558415]

...

[ 0.3429808 0.00330087 0.14964075 ... -0.73518964 -0.68519336

-0.27575966]

[-0.84488505 0.1597866 -0.47073225 ... -0.24020459 -0.37110101

1.17073215]

[-0.84488505 -0.8730192 0.04624525 ... -0.20212881 -0.47378505

-0.87137393]]

In [17]:

X **=** standardized\_data

Y **=** diabetes\_dataset['Outcome']

In [18]:

print(X)

print(Y)

[[ 0.63994726 0.84832379 0.14964075 ... 0.20401277 0.46849198

1.4259954 ]

[-0.84488505 -1.12339636 -0.16054575 ... -0.68442195 -0.36506078

-0.19067191]

[ 1.23388019 1.94372388 -0.26394125 ... -1.10325546 0.60439732

-0.10558415]

...

[ 0.3429808 0.00330087 0.14964075 ... -0.73518964 -0.68519336

-0.27575966]

[-0.84488505 0.1597866 -0.47073225 ... -0.24020459 -0.37110101

1.17073215]

[-0.84488505 -0.8730192 0.04624525 ... -0.20212881 -0.47378505

-0.87137393]]

0 1

1 0

2 1

3 0

4 1

..

763 0

764 0

765 0

766 1

767 0

Name: Outcome, Length: 768, dtype: int64

# Train Test Split

In [19]:

X\_train, X\_test, Y\_train, Y\_test **=** train\_test\_split(X,Y, test\_size **=** 0.2, stratify**=**Y, random\_state**=**2)

In [20]:

print(X**.**shape, X\_train**.**shape, X\_test**.**shape)

(768, 8) (614, 8) (154, 8)

# Training the Model

In [49]:

classifier **=** LogisticRegression()

In [50]:

*#training the support vector Machine Classifier*

classifier**.**fit(X\_train, Y\_train)

Out[50]:

LogisticRegression()

# Model Evaluation

Accuracy Score

In [23]:

*# accuracy score on the training data*

X\_train\_prediction **=** classifier**.**predict(X\_train)

training\_data\_accuracy **=** accuracy\_score(X\_train\_prediction, Y\_train)

In [31]:

print('Accuracy score of the training data : ', training\_data\_accuracy)

Accuracy score of the training data : 0.7866449511400652

In [25]:

*# accuracy score on the test data*

X\_test\_prediction **=** classifier**.**predict(X\_test)

test\_data\_accuracy **=** accuracy\_score(X\_test\_prediction, Y\_test)

In [32]:

print('Accuracy score of the test data : ', test\_data\_accuracy)

Accuracy score of the test data : 0.7727272727272727

**EXPERIMENTAL RESULT**

# Making a Predictive System

input\_data **=** (1,103,30,38,83,43.3,0.183,33)

​

​

input\_data\_as\_numpy\_array **=** np.asarray(input\_data)

​

​

input\_data\_reshaped **=** input\_data\_as\_numpy\_array.reshape(1,**-**1)

​

std\_data **=** scaler.transform(input\_data\_reshaped)

print(std\_data)

​

prediction **=** classifier.predict(std\_data)

print(prediction)

​

**if** (prediction[0] **==** 0):

print('The person is not diabetic')

**else**:

print('The person is diabetic')

[[-0.84488505 -0.56004775 -2.02166474 1.09545411 0.02778979 1.43512945

-0.87244072 -0.0204964 ]]

[0]

The person is not diabetic

In this project we have used a publicly available dataset from dropbox [diabetes.csv (dropbox.com)](https://www.dropbox.com/s/uh7o7uyeghqkhoy/diabetes.csv?dl=0)

This database contains 768attributes, but all published experiments refer to using a subset of 9 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to

this date. The "goal" field refers to the presence of diabetes in the patient.

**CONCLUSION**

A Diabetes detection model has been developed. This project predicts people with diabetes by extracting the patient medical history that leads to a diabetes from a dataset that includes patients’ medical history such as glucose,BMI etc

The logistic regression-based classifier generates Accuracy of 0.7866449511400652on Training data and Accuracy of 0.7727272727272727 on Testing data.