Machine Learning Project

Submitted by:

(102003128) Shivesh Narula

(102003136) Arnav Singh

BE Second Year

Submitted to:

Niyaz Wani

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Jan -June 2022

Introduction

Machine Learning is used across many ranges around the world. The healthcare industry is no exclusion. Machine Learning can play an essential role in predicting presence/absence of locomotors disorders, Heart diseases and more. Such information, if predicted well in advance, can provide important intuitions to doctors who can then adapt their diagnosis and dealing per patient basis. We work on predicting possible Heart Diseases in people using Machine Learning algorithms.

This project aims to predict future Heart Disease by analyzing data of patients which classifies whether they have heart disease or not using machine-learning algorithm. Machine Learning techniques can be a boon in this regard. Even though heart disease can occur in different forms, there is a common set of core risk factors that influence whether someone will ultimately be at risk for heart disease or not. By collecting the data from various sources, classifying them under suitable headings & finally analysing to extract the desired data we can say that this technique can be very well adapted to do the prediction of heart disease.

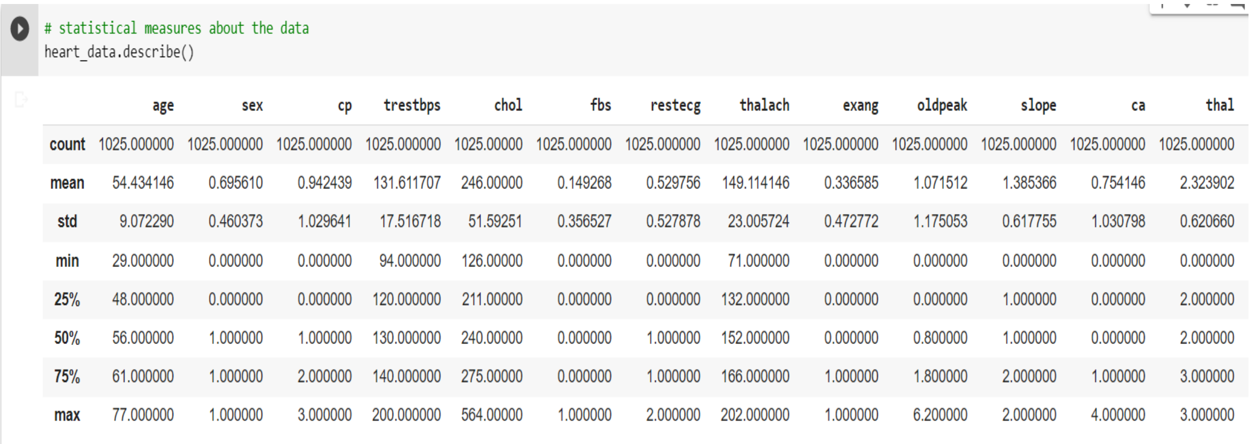
Data Set Documentation

Title: Heart Disease Prediction Model

Sources:

Dataset: https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

Attribute Statistics:



```
# statistical measures about the data
heart_data.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.529756	149.114146	0.336585	1.071512	1.385366	0.754146	2.323902
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	1.175053	0.617755	1.030798	0.620660
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.000000	132.000000	0.000000	0.000000	1.000000	0.000000	2.000000
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.000000	152.000000	0.000000	0.800000	1.000000	0.000000	2.000000
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.000000	166.000000	1.000000	1.800000	2.000000	1.000000	3.000000
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000

Problem Statement

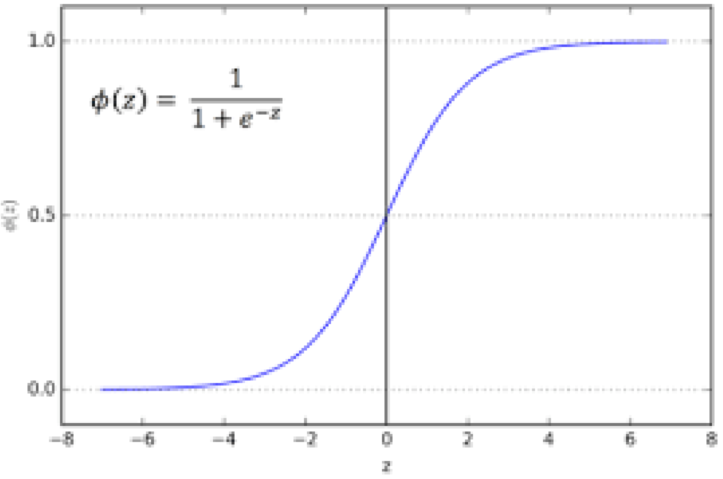
This video is about building a Heart Disease Prediction system using Machine Learning with Python.

Algorithm

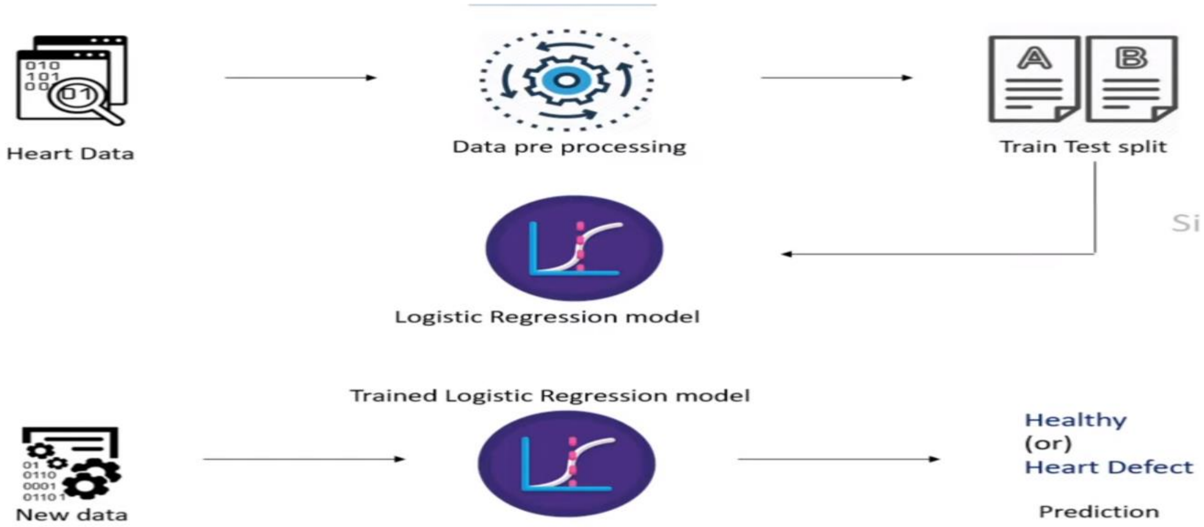
Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

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.

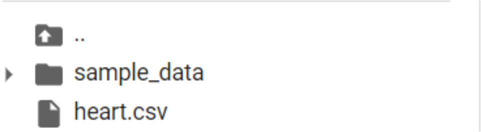


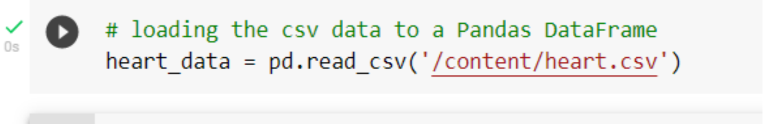
Work Flow



1. The first step for creating a model is collecting data. We need appropriate data to train our model. Data is usually collected in a csv (comma-separated values) file.

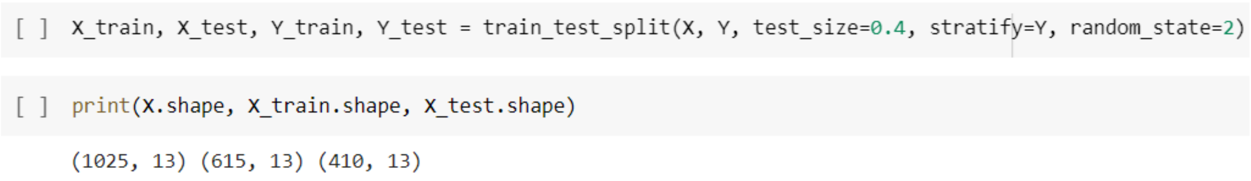
2. Once we have the data we cannot feed it directly to our machine learning algorithm so before feeding we need to process the data so this step is called as data preprocessing where we clean the data and do some processing on.





```
# loading the csv data to a Pandas DataFrame
heart_data = pd.read_csv('/content/heart.csv')
```

3. Train Test Split:- We split the data set into training data and test data. The reason for this is to train our machine learning model with the training data and we will evaluate our model. Then we will test our model with this test. After we split the data we need to train our machine learning algorithm.

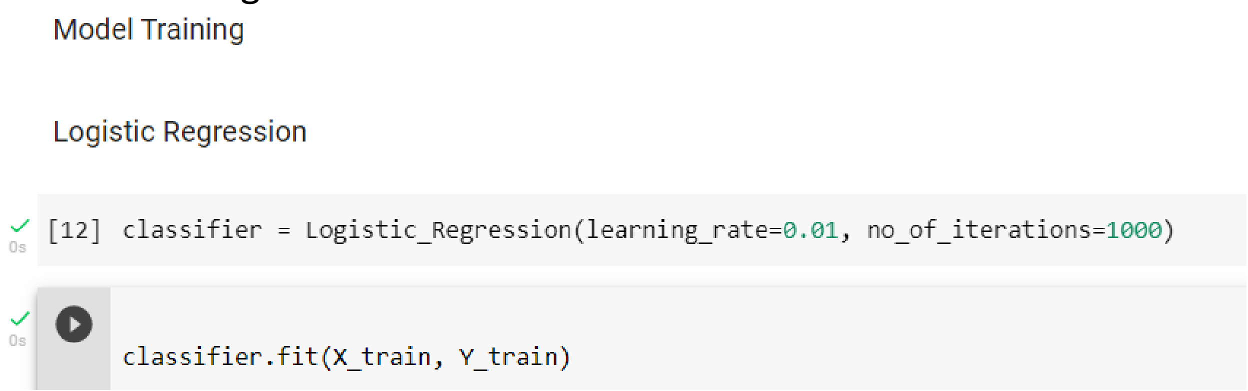


```
[ ]  X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, stratify=Y, random_state=2)

[ ]  print(X.shape, X_train.shape, X_test.shape)

     (1025, 13) (615, 13) (410, 13)
```

4. Model Training

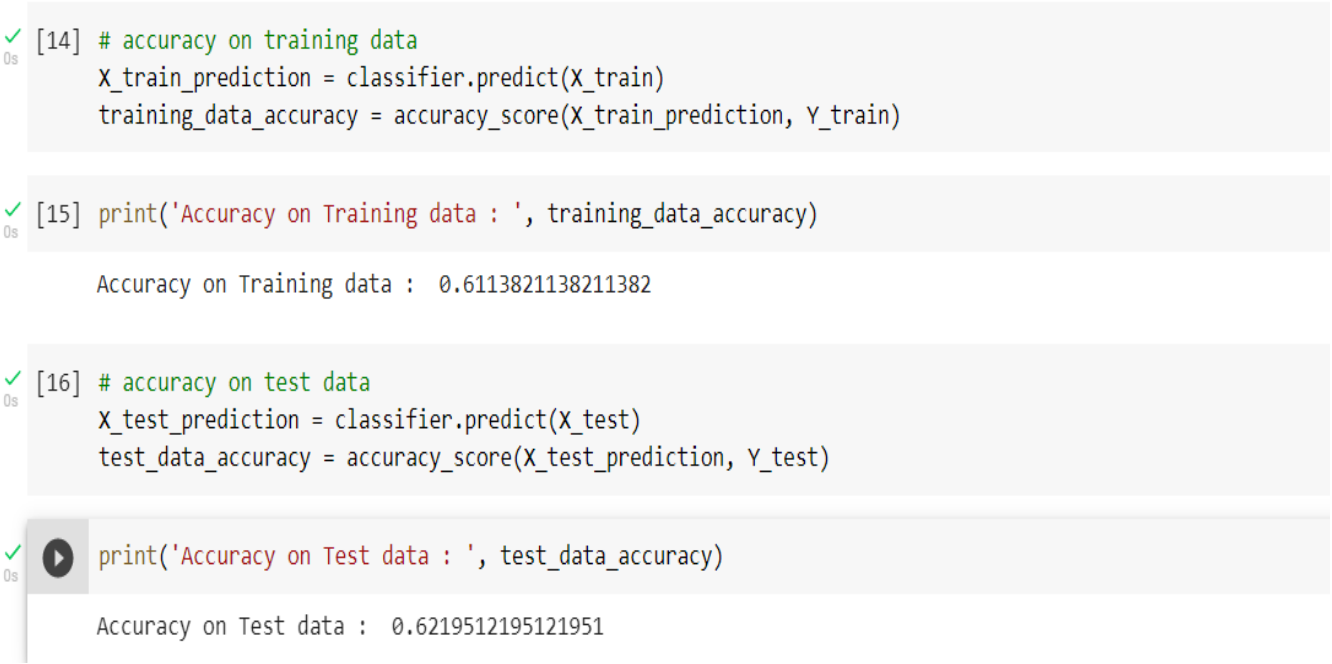


```
[12]  classifier = Logistic_Regression(learning_rate=0.01, no_of_iterations=1000)

      classifier.fit(X_train, Y_train)
```

5. We will evaluate our model with the help of test data.

Results



```
[14] # accuracy on training data
     X_train_prediction = classifier.predict(X_train)
     training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

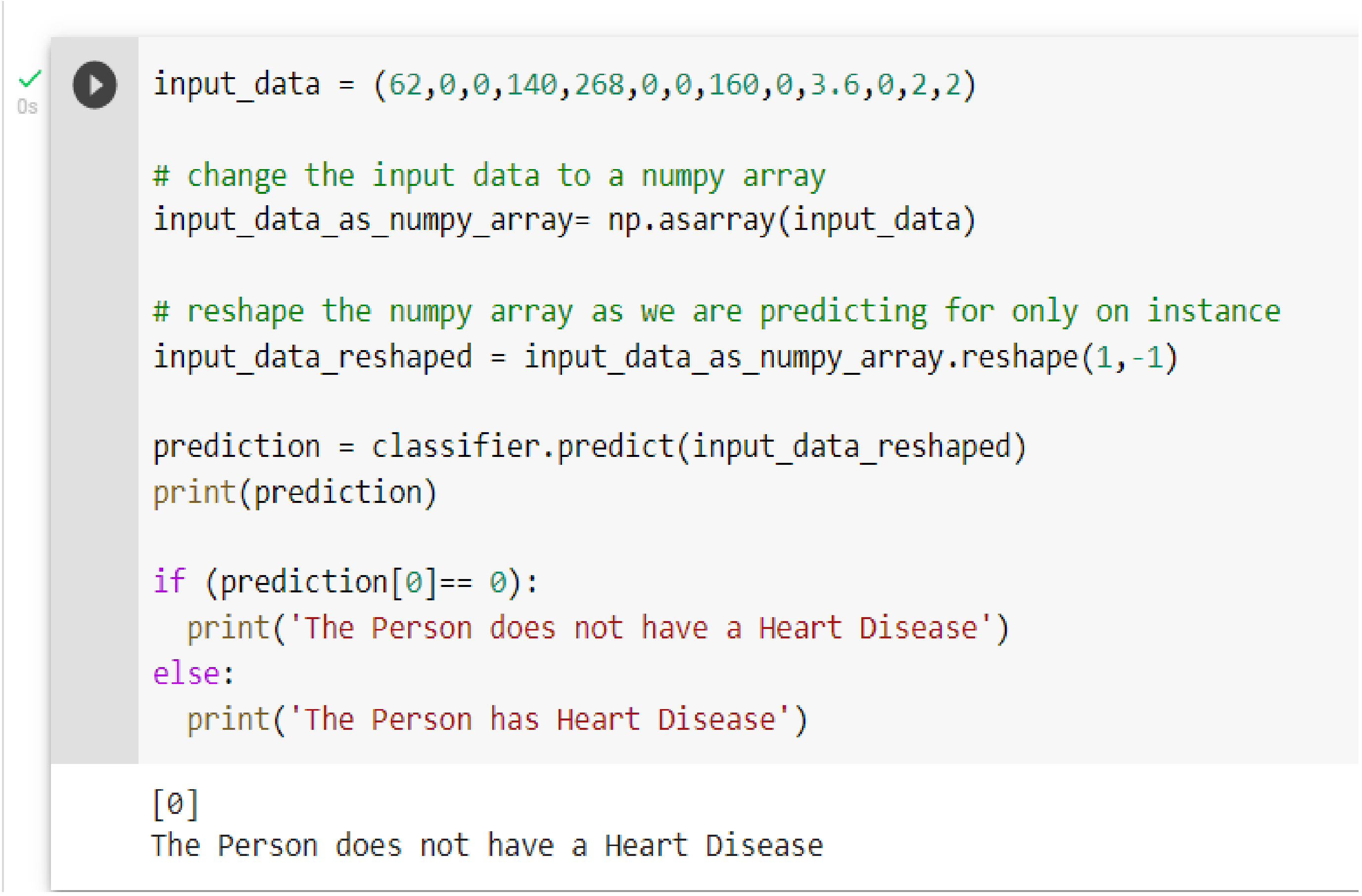
```
[15] print('Accuracy on Training data : ', training_data_accuracy)
```

```
Accuracy on Training data :  0.6113821138211382
```

```
[16] # accuracy on test data
     X_test_prediction = classifier.predict(X_test)
     test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
print('Accuracy on Test data : ', test_data_accuracy)
```

```
Accuracy on Test data :  0.6219512195121951
```



```
input_data = (62,0,0,140,268,0,0,160,0,3.6,0,2,2)

# change the input data to a numpy array
input_data_as_numpy_array= np.asarray(input_data)

# reshape the numpy array as we are predicting for only on instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = classifier.predict(input_data_reshaped)
print(prediction)

if (prediction[0]== 0):
  print('The Person does not have a Heart Disease')
else:
  print('The Person has Heart Disease')
```

```
[0]
The Person does not have a Heart Disease
```