MACHINE LEARNING

UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

DEPARTMENT OF SOFTWARE ENGINEERING

PROJECT-REPORT

Instructor

DR. HASSAN DAWOOD

Name	Registration No
Rafia Mehmooda	20-SE-62
Ghufran Ullah	20-SE-34
Muhammad Noman Shafique	20-SE-38

Date: 11th June 2023

Project Title: Heart Disease prediction Using Logistic Regression Model

Dataset Source: Kaggle

Tool: Google Collaboratory

Project Link:

https://colab.research.google.com/drive/1KJe-KxPof_P18ILAvhxqFBdj1GFromFq?usp=sharing#scrollTo=58j4WubdyOtw

PROBLEM STATEMENT:

"PREDICT WHETHER A PERSON HAS HEART DISEASE OR NOT"

Workflow:

- After getting data we have done some preprocessing on data and after spitting our data into features and labels we again split our data into training and testing data and make our model to learn our training data.
- Then we made our model to predict labels for both training and testing and in this way calculated accuracy score.
- Accuracy scores for both of them are quite near due to which we can say that model is trained well. If there is a void difference in our accuracy score of training and testing, then we assume that our model is over fitted which can be solved by increasing training data and by simplifying our model and so on.
- In last we have made a python program which is successfully predicting given one instance that whether it has a diseased hear or not.

Implementation:

importing dependencies

import numpy as np

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score

Data collection and processing

#loading the csv data to pandas dataframe

heart_data = pd.read_csv('/content/heart_disease_data.csv')

print first 5 rows of the dataset

```
heart_data.head()

#print last 5 rows of dataset

heart_data.tail()

# no of rows and columns in data set

heart_data.shape

#getting some info about data

heart_data.info()

#another method to check for missing values

heart_data.isnull().sum()

#statistical measures aboutt the data

heart_data.describe()

#checking the distrubution of tARGET VARIABLE

heart_data['target'].value_counts()
```

1--> defective heart

0-->healthy heart

splitting features and target

all other columns except target represents features because on them we are going to apply some model and predict that person ahs heart disease or not.

```
X = heart_data.drop(columns='target',axis=1)

Y = heart_data['target']

print(X)

print(Y)
```

Splitting data into training ad test data

```
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,stratify=Y,random_state=2)

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

X_test.head()

Y_test.head()
```

Model training using Logistic Regression Model

```
model = LogisticRegression()
```

```
#traing our model with training dataset
model.fit(X_train,Y_train)
# model evaluation using accuracy score
#accuracu on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction,Y_train)

print('accuracy on training data', training_data_accuracy)
#accuracu on testing data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction,Y_test)
print('accuracy on test data', test_data_accuracy)
Building a predicted system
input_data = (45,1,0,142,309,0,0,147,1,0.0,1,3,3)

# 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 = model.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')
```