NNFS Lab -2

Prediction of heart disease using Linear Regression.

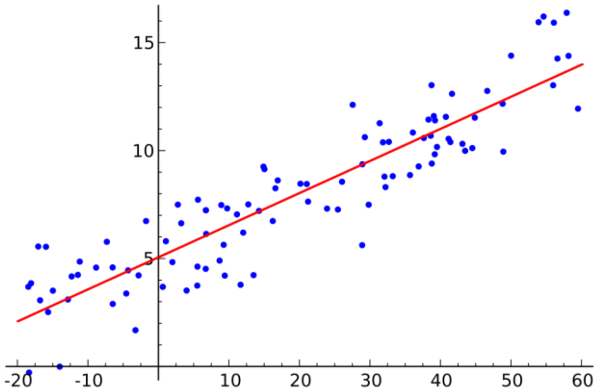
AIM: - Predicting the risk of having heart disease using Linear Regression.

PLATFORM & TOOLS USED:

1. Google colab
2. Python
3. Sklearn

THEORY:

Regression is a method of modelling a target value based on independent predictors. This method is mostly used for forecasting and finding out cause and effect relationship between variables. Regression techniques mostly differ based on the number of independent variables and the type of relationship between the independent and dependent variables.



Linear Regression

Simple linear regression is a type of regression analysis where the number of independent variables is one and there is a linear relationship between the independent(x) and dependent(y) variable. The red line in the above graph is referred to as the best fit straight line. Based on the given data points, we try to plot a line that models the points the best. The line can be modelled based on the linear equation shown below.

y = a_0 + a_1 * x ## Linear Equation

The motive of the linear regression algorithm is to find the best values for a_0 and a_1. Before moving on to the algorithm, let’s have a look at two important concepts you must know to better understand linear regression.

Making Predictions with Linear Regression

Given the representation is a linear equation, making predictions is as simple as solving the equation for a specific set of inputs.

Let’s make this concrete with an example. Imagine we are predicting weight (y) from height (x). Our linear regression model representation for this problem would be:

y = B0 + B1 * x1

or

weight =B0 +B1 * height

Where B0 is the bias coefficient and B1 is the coefficient for the height column. We use a learning technique to find a good set of coefficient values. Once found, we can plug in different height values to predict the weight. For example, let’s use B0 = 0.1 and B1 = 0.5. Let’s plug them in and calculate the weight (in kilograms) for a person with the height of 182 centimetres.

$$\text{weight} = 0.1 + 0.5 * 182$$

$$\text{weight} = 91.1$$

The B0 is our starting point regardless of what height we have. We can run through a bunch of heights from 100 to 250 centimetres and plug them to the equation and get weight values, creating our line.

CODE:

```
import pandas as pd #for loading the dataset
dataset = pd.read_csv('cardio_dataset.csv').values #load the dataset into a numpy a
rray

print(dataset[:10]) #first 10 entries of the dataset
data = dataset[:,0:7] #all rows and columns from 0 to 6
target = dataset[:,7] #all rows and 7th column
print(dataset.shape)

from sklearn.model_selection import train_test_split
train_data,test_data,train_target,test_target = train_test_split(data,target,test_s
ize=0.2)
from sklearn.linear_model import LinearRegression
model = LinearRegression() #Loading the algorithm
model.fit(train_data,train_target)
predicted_target = model.predict(test_data)

from sklearn.metrics import r2_score
r2 = r2_score(test_target,predicted_target)
print("r2: ",r2)
```

Using Feature Engineering

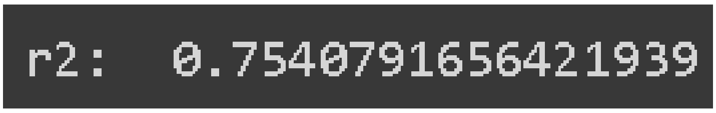
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler ()
train_data_new = sc.fit_transform(train_data)

test_data_new = sc.transform(test_data)


model.fit(train_data_new,train_target)
predicted_target_new = model.predict(test_data_new)
from sklearn.metrics import r2_score
r2_new = r2_score(test_target,predicted_target_new)
print("r2: ",r2_new)

import joblib
joblib.dump(model,'heart_risk_prediction_regression_model.sav')
```

EXPLANATION:

1. To first load the cardio dataset into memory we have used the pandas library which is a data analysis and manipulation tool. We use the read_csv() function to load the load the data in a Data Frame structure.
2. We then split the dataset into 2 parts, data and target. Data contains the matrix of features or the independent variable vector and target contains the dependent variable vector.
3. We then use the scikit train_test_split function so that we can partition our data into train set and test set. The test dataset contains data which the model has not seen before
4. We then import the Linear Regression class from Scikit_Learn and then use the model.fit() function to train the model.
5. Additionally we have also applied Feature Scaling by using the StandardScaler() and applied it on the training set.
6. We then measure the performance of the model by using r2_score which is recommended when utilizing multiple linear regression.
7. We then used joblib library to save the model weights.

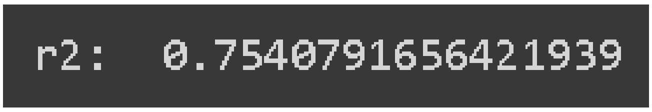
RESULTS:

After training the model on the train set and testing it on the test dataset which contains data the model has not seen before, the performance of the model by using r2_score is given below.



```
r2:  0.7540791656421939
```

Additionally, feature scaling has also been applied to the data and it was observed that there was little or no observed change in the accuracy.



```
r2:  0.7540791656421939
```

CONCLUSION:

A linear regression model has been successfully trained and implemented on the cardio dataset to predict the risk of heart disease. It is observed that with a larger dataset and using other algorithms we can achieve better performance metrics and help in accurate detections. Feature Scaling has also been implemented on the dataset but the performance metric has not drastically changed, since the features fall within range close to each other with no extremities.