IoT Domain Analyst

Lab Record — Lab 3

- 22 February 2021

Programme	:	B.Tech(CSE)	Semester	:	Winter 2020–21
Course Title	:	IoT Domain Analyst – Lab	Code	:	ECE3502
			Slot	:	L5+L6
Name	:	**Aadhitya Swarnesh**	Registration. No	:	[redacted]
Faculty (s)	:	[redacted]	Expt. No	:	**3**

Experiment 1 :

To find the co-relation between the weights and heights dataset and then use it to make a model that predicts the height given the weight of a person.

Aim :

To build a Regression model in the R programming language that predicts the height of a person given their weight.

Description :

In order to create such a prediction model, we first train the regression model with the available dataset containing the heights and the weights, and once this is complete, the model would have learnt the co-relation between the heights and the weights.

We can then use this trained model, and feed in our query weight, and the model will predict the corresponding height of the person based on the knowledge obtained during the training phase.

We also plot a regression line or the best fit line after the model has finished the training phase to demonstrate the results of the training phase.

Code :

The code that is used for this Experiment is as follows :

```
weight <-c (150,180,140,128,133,152,131)
height <-c (62,84,55,52,54,63,53)
model <- lm (height ~ weight)
print (model)


test <- data.frame (weight = 165)
result <- predict (model , test)
print (result)

plot ( height, weight, col = "red", main = "Example of
    Linear Regression" , abline (lm(weight~height)),
    cex = 1.3, pch = 16, xlab = "Weight" , ylab =
    "Height")
```

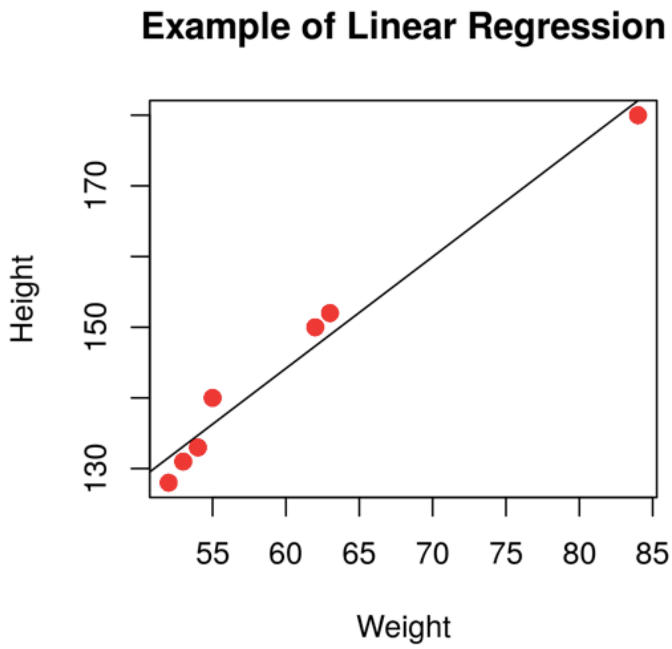
Result :

The result for the obtained code displays the height of the person when their weight is fed into the model, and the output for the same with the co-relation plot is as follows :

```
Call:
lm(formula = height ~ weight)

Coefficients:
(Intercept)       weight
   -28.7640       0.6157

        1
72.83109
```



Experiment 2 :

To find the co-relation between a set of given points, and thereby use this data to find the best-fit line.

Aim :

To find the co-relation between a set of 2D points, and thereby use it to find the slope of the Line and to create a regression model for the same.

Description :

In order to create such a prediction model, we first train the regression model with the available dataset containing the x and the y co-ordinates, and once this is complete, the model would have learnt the co-relation between the co-ordinates.

We then use this knowledge obtained to calculate the point of intersection of the line, the slope of the line and the coefficient of determination of the line.

We use the predict function of the created model to make predictions for the y co-ordinate given the point's x-coordinate, and given that the point lies in the regression line.

We then test the validity of the prediction result obtained before by using the previously obtained coefficient and intercept values, and using it to manually predict the y co-ordinate given the x co-ordinate.

Code :

The code that is used for this Experiment is as follows :

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import scipy as sp
from sklearn.linear_model import LinearRegression

x = np.array([5, 15, 25, 35, 45, 55]).reshape((-1, 1))
y = np.array([5, 20, 14, 32, 22, 38])
print(x)
print(y)

model = LinearRegression().fit(x, y)
r_sq = model.score(x, y)
print('coefficient of determination:', r_sq)
print('intercept:', model.intercept_)
print('slope:', model.coef_)

new_model = LinearRegression().fit(x, y.reshape((-1, 1)))
print('intercept:', new_model.intercept_)
print('slope:', new_model.coef_)

y_pred = model.predict(x)
print('predicted response:', y_pred, sep='\n')
y_pred = model.intercept_ + model.coef_ * x
print('predicted response:', y_pred, sep='\n')

print(" First few points of the line :”)
x_new = np.arange(5).reshape((-1, 1))
print(x_new)
y_new = model.predict(x_new)
print(y_new)
```

Result :

The result for the above code displays the slope of the regression line obtained after the training phase, and also the prediction values for the test point. It shows the validity of the **predict** function, as the values obtained here are the same as when the calculations are done manually. We also see how the **Linear Regression model function** works different if we change the dimensionality of the x or y values. We then go ahead and find the first few points of the line from the intercept up in the first quadrant.

The output for all theses tasks are as follows :

```
(base) Aadhityas-MacBook-Air:lab 3 aadhitya$ python Linear_Regression.py
[[ 5]
 [15]
 [25]
 [35]
 [45]
 [55]]
[ 5 20 14 32 22 38]
coefficient of determination: 0.715875613747954
intercept: 5.633333333333329
slope: [0.54]
intercept: [5.63333333]
slope: [[0.54]]
predicted response:
[ 8.33333333 13.73333333 19.13333333 24.53333333 29.93333333 35.33333333]
predicted response:
[[ 8.33333333]
 [13.73333333]
 [19.13333333]
 [24.53333333]
 [29.93333333]
 [35.33333333]]
 First few points of the line :
[[0]
 [1]
 [2]
 [3]
 [4]]
[5.63333333 6.17333333 6.71333333 7.25333333 7.79333333]
(base) Aadhityas-MacBook-Air:lab 3 aadhitya$
```

Experiment 3 :

To create a Multiple Linear Regression model to predict the Miles run by the vehicle based on a few parameters.

Aim :

To create a Multiple Linear Regression model to predict the Miles run by the vehicle based on the Horse Power of the vehicle and the displacement values in the R programming Language.

Description :

In order to create such a prediction model, we first train the regression model with the available dataset of the MTCARS containing all the required details, and once this is complete, the model would have learnt the co-relation between the parameters and the target variable.

We then use this knowledge obtained to predict the Miles run by the vehicle based on the Horse Power of the vehicle and the displacement values. We use the default R predict function of the created model to make predictions miles run by the vehicle.

We then plot this co-relation on a graph to both test the validity of the training process, and also to gain insights about the task in hand through visualisations.

Code :

The code that is used for this Experiment is as follows :

```
data <- mtcars[, c ("mpg" , "disp", "hp")]
head (data)

model <- lm(mpg~disp + hp, data = data)
summary (model)

predict (model, newdata = data.frame (disp =140, hp = 80))
predict (model, newdata = data.frame (disp =160, hp = 70))

plot (model)
```

Result :

The result for the obtained code displays the miles run by the vehicle when its horse power capacity and the displacement values are fed into the model, and the output for the same with the co-relation plot is as follows :

```
$Rscript main.r
                   mpg disp  hp
Mazda RX4         21.0  160 110
Mazda RX4 Wag     21.0  160 110
Datsun 710        22.8  108  93
Hornet 4 Drive    21.4  258 110
Hornet Sportabout 18.7  360 175
Valiant           18.1  225 105

Call:
lm(formula = mpg ~ disp + hp, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-4.7945 -2.3036 -0.8246  1.8582  6.9363

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 30.735904   1.331566  23.083  < 2e-16 ***
disp        -0.030346   0.007405  -4.098 0.000306 ***
hp          -0.024840   0.013385  -1.856 0.073679 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.127 on 29 degrees of freedom
Multiple R-squared:  0.7482,    Adjusted R-squared:  0.7309
F-statistic: 43.09 on 2 and 29 DF,  p-value: 2.062e-09

       1
24.50022
       1
24.14169
```