pratical-4

April 17, 2024

---

1 Data Analytics I

Create a Linear Regression Model using Python/R to predict home prices using Boston Housing Dataset (https://www.kaggle.com/c/boston-housing). The Boston Housing dataset contains information about various houses in Boston through different parameters. There are 506 samples and 14 feature variables in this dataset. The objective is to predict the value of prices of the house using the given features

[ ]:
```
# # Import the required libraries
# %matplotlib inline
# import numpy as np
# import pandas as pd
# import matplotlib.pyplot as plt
# plt.rcParams['figure.figsize'] = (20.0, 10.0)
# from sklearn.linear_model import LinearRegression
# from sklearn.metrics import mean_squared_error, r2_score


# # Reading data
# data = pd.read_csv('BostonHousing.csv')
# print(data.shape)
# data.head()

# # Collecting X and Y
# X = data['dis'].values
# Y = data['medv'].values

# # Calculating Coefficient
# # Mean X and Y
# mean_x = np.mean(X)
# mean_y = np.mean(Y)

# # Total number of values
# n = len(X)
# n
```

```
# # using the formula to calculate b1 and b2
# numer = 0
# denom = 0
# for i in range(n):
#     numer += (X[i] - mean_x) * (Y[i] - mean_y)
#     denom += (X[i] - mean_x) ** 2

# b1 = numer/denom
# b0 = mean_y - (b1 * mean_x)

# # m(b1) and c(b0)
# # Pinting coefficients
# print("Coefficients")
# print(f"m = {b1}")
# print(f"c = {b0}")

# # Plotting Values and Regression Line
# max_x = np.max(X)
# min_x = np.min(X)

# # Calculating Line values x and y
# x = np.linspace(min_x, max_x, 1000)
# y = b0 + b1 * x

# # Ploting Line
# plt.plot(x, y, color='green', label='Regression Line')
# # Ploting Scatter Points
# plt.scatter (X, Y, c='red', label='Scatter Plot')
# plt.xlabel('Head Size in cm3')
# plt.ylabel('Brain Weight in grams')
# plt.legend()
# plt.show()

# # Calculating R2 Score
# ss_tot = 0
# ss_res = 0
# for i in range(n):
#     y_pred = b0 + b1 * X[i]
#     ss_tot += (Y[i] - mean_y) ** 2
#     ss_res += (Y[i] - y_pred) ** 2
# r2 = 1 - (ss_res/ss_tot)
# print("R2 Score")
# print(r2)

# data=pd.read_csv('BostonHousing.csv')
# X = data.iloc[:,7].values.reshape(-1,1) #converts it into numpy array
```

```
# Y = data.iloc[:,13].values.reshape(-1,1)
# linear_regressor=LinearRegression() # create obect for class
# linear_regressor.fit(X, Y) # perform Linear regression
# y_pred=linear_regressor.predict(X) # make prediction

# plt.scatter(X,Y)
# plt.plot(X, y_pred, color =  'red')

# # The coefficients
# print(f"Coefficients:\n{linear_regressor.coef_}")

# print("Coefficient of determination: %.2f" % r2_score(Y, y_pred))
```

[1]:
```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (20.0, 10.0)
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

[2]:
```
# Reading data
data = pd.read_csv('BostonHousing.csv')
print(data.shape)
data.head()
```

```
(506, 14)
```

[2]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	medv
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2

[3]:
```
# Collecting X and Y
X = data['dis'].values
Y = data['medv'].values
```

[4]:
```
# Calculating Coefficient

# Mean X and Y
mean_x = np.mean(X)
mean_y = np.mean(Y)

# Total number of values
n = len(X)
```

[5]:
```
n
```

[5]:
```
506
```

[6]:
```
# using the formula to calculate b1 and b2
numer = 0
denom = 0
for i in range(n):
    numer += (X[i] - mean_x) * (Y[i] - mean_y)
    denom += (X[i] - mean_x) ** 2
```

[7]:
```
b1 = numer/denom
b0 = mean_y - (b1 * mean_x)
```

[8]:
```
# m(b1) and c(b0)
# Pinting coefficients
print("Coefficients")
print(f"m = {b1}")
print(f"c = {b0}")
```

```
Coefficients
m = 1.0916130158411097
c = 18.390088330493384
```

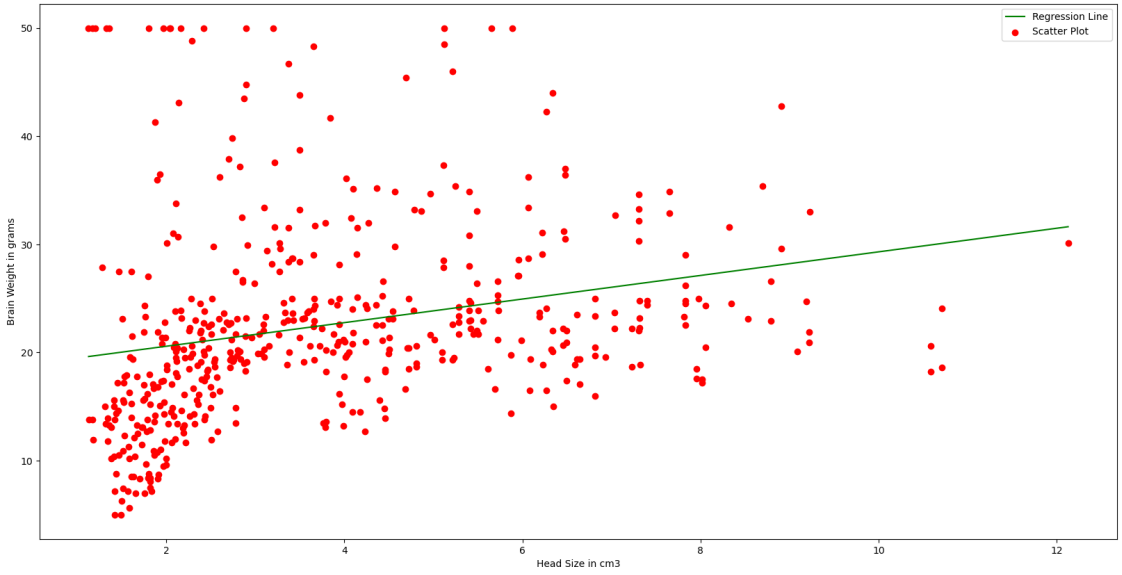
[9]:
```
# Plotting Values and Regression Line

max_x = np.max(X)
min_x = np.min(X)

# Calculating Line values x and y
x = np.linspace(min_x, max_x, 1000)
y = b0 + b1 * x

# Ploting Line
plt.plot(x, y, color='green', label='Regression Line')
# Ploting Scatter Points
plt.scatter (X, Y, c='red', label='Scatter Plot')
plt.xlabel('Head Size in cm3')
```

```
plt.ylabel('Brain Weight in grams')
plt.legend()
plt.show()
```



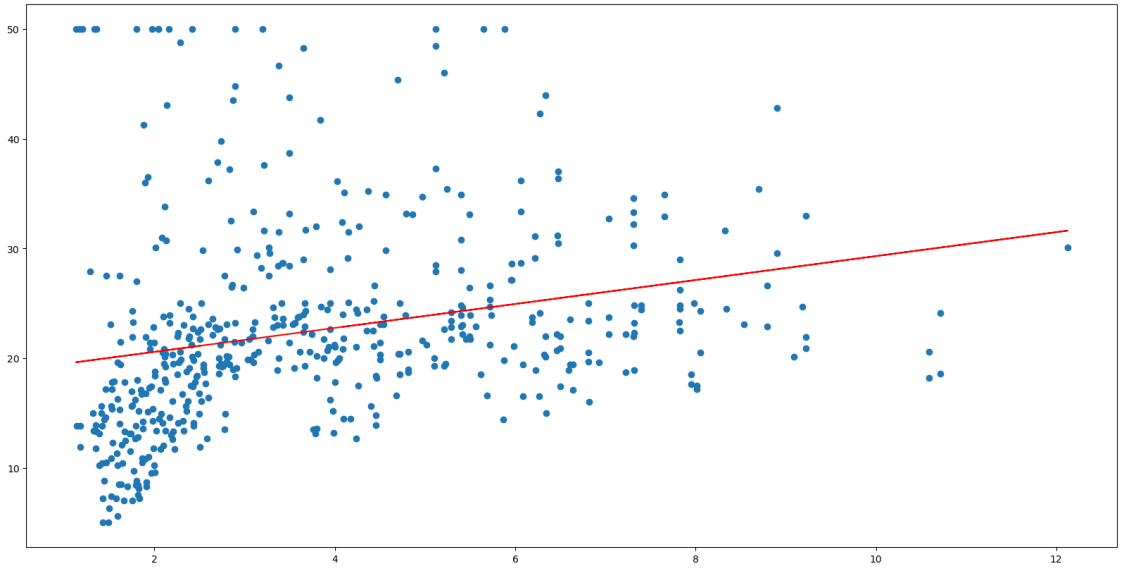
[10]:
```
# Calculating R2 Score
ss_tot = 0
ss_res = 0
for i in range(n):
    y_pred = b0 + b1 * X[i]
    ss_tot += (Y[i] - mean_y) ** 2
    ss_res += (Y[i] - y_pred) ** 2
r2 = 1 - (ss_res/ss_tot)
print("R2 Score")
print(r2)
```

```
R2 Score
0.06246437212178291
```

[11]:
```
data=pd.read_csv('BostonHousing.csv')
X = data.iloc[:,7].values.reshape(-1,1) #converts it into numpy array
Y = data.iloc[:,13].values.reshape(-1,1)
linear_regressor=LinearRegression() # create obect for class
linear_regressor.fit(X, Y) # perform Linear regression
y_pred=linear_regressor.predict(X) # make prediction
```

[12]:
```
plt.scatter(X,Y)
plt.plot(X, y_pred, color =  'red')
```

[12]: [<matplotlib.lines.Line2D at 0x2e079998890>]



[13]:
```
# The coefficients
print(f"Coefficients:\n{linear_regressor.coef_}")
```

```
Coefficients:
[[1.09161302]]
```

[14]:
```
print("Coefficient of determination: %.2f" % r2_score(Y, y_pred))
```

```
Coefficient of determination: 0.06
```

[ ]: