ML_Lab_4_LinearRegression_One_Variable

November 8, 2021

0.0.1 ML LAB - Dr Neeraj Gupta (neeraj.gupta@gla.ac.in)

0.1 Linear Regression with One Variable

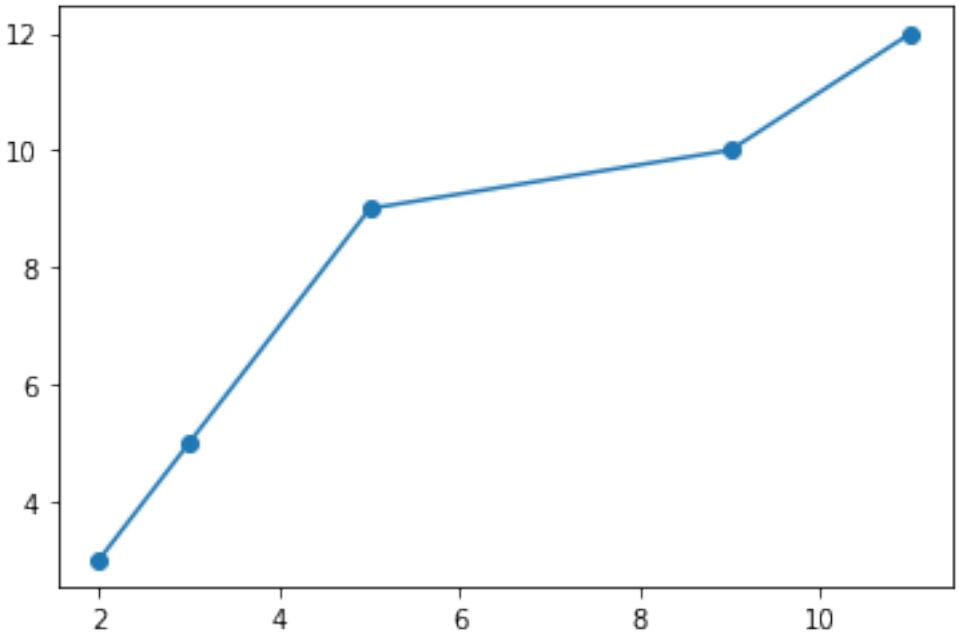
[9]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

[10]:
```
#Two Points are given (3, 5) and (9,10)
#Find equation of line ?
def slope_intercept(x1,y1,x2,y2):
    a = (y2 - y1) / (x2 - x1)
    b = y1 - a * x1
    return a,b
m,c = slope_intercept(3,5,9,10)
print(slope_intercept(3,5,9,10))
print("Equation of line : y = {0:.2f}.x + {1}".format(m,c))
```

```
(0.8333333333333334, 2.5)
Equation of line : y = 0.83.x + 2.5
```

[11]:
```
# X : 2, 3, 5, 9, 7, 11, 10.5
# Y : 3, 5, 9, 10, 6,5, 11.8, ?
X =[ 2, 3, 5, 9, 11]
Y =[ 3, 5, 9, 10, 12 ]

# Ploting Line
plt.scatter(X,Y,label='Scatter plot')
plt.plot(X,Y,label='Line')
```

[11]:
```
[<matplotlib.lines.Line2D at 0x24004cc2448>]
```



[12]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Collecting X and Y
X = [171,151,124,134,156] #height
Y = [80,60,45,50,65] #weight
```

[13]:
```
# Calculating coefficient

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

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

[14]:
```
# Using the formula to calculate b1 and b0
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)
```

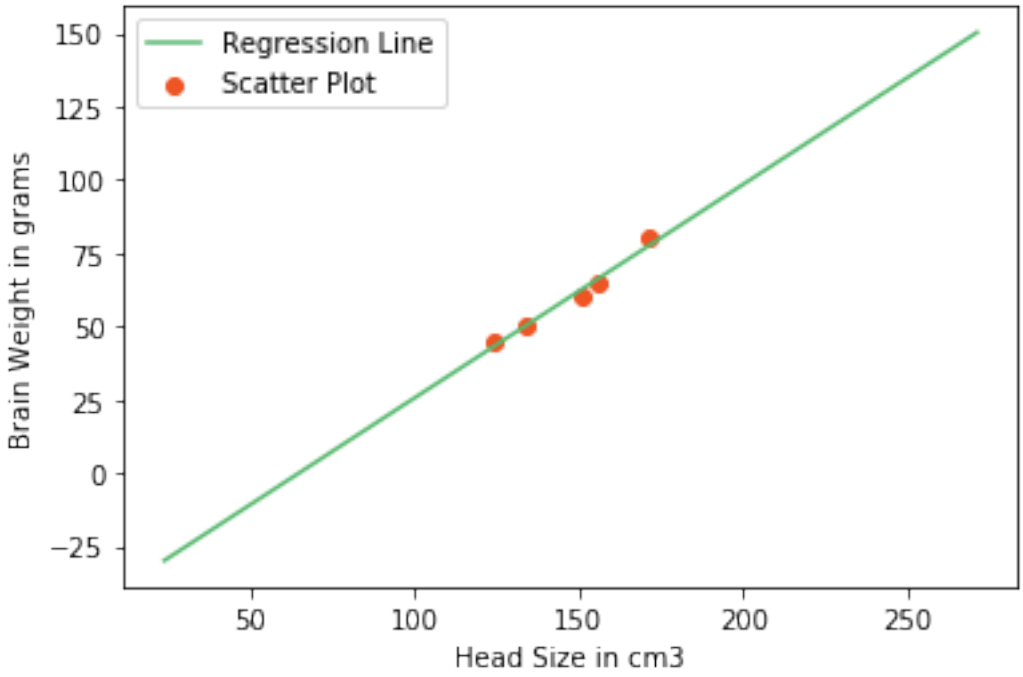
```
# Printing coefficients
print("Coefficients")
print(b1, b0)
```

```
Coefficients
0.7295010213014298 -47.38255033557046
```

[15]:
```
#predict the weight of person having height= 130?
#b1 is slope
#b0 intercept
#x= 130
print(b1)
print(b0)
y = b0 + b1*171
print(y)
```

```
0.7295010213014298
-47.38255033557046
77.36212430697404
```

[16]:
```
# Plotting Values and Regression Line
max_x = np.max(X) + 100
min_x = np.min(X) - 100

# Calculating line values x and y
x = np.linspace(min_x, max_x,100)
y =[]
for i in range(100):
    y.append(b0 + b1 * x[i])
#print(y)

# Ploting Line
plt.plot(x, y, color='#58b970', label='Regression Line')
# Ploting Scatter Points
plt.scatter(X, Y, color='#ef5423', label='Scatter Plot')

plt.xlabel('Head Size in cm3')
plt.ylabel('Brain Weight in grams')
plt.legend()
plt.show()
```



[17]:
```
%%html
<img src='image/pd_rmse.jpg', width=300, height=300>
```

```
<IPython.core.display.HTML object>
```

[18]:
```
# Calculating Root Mean Squares Error
rmse = 0
for i in range(n):
    y_pred = b0 + b1 * X[i]
    rmse += (Y[i] - y_pred) ** 2
rmse = np.sqrt(rmse/n)
print("RMSE")
print(rmse)
```

```
RMSE
2.02479523402097
```

[19]:
```
%%html
<img src='image/pd_r2score.jpg', width=300, height=300>
```

```
<IPython.core.display.HTML object>
```

[20]:
```
# 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
0.9726680284019065
```

0.1.1 ML LAB - Dr Neeraj Gupta (neeraj.gupta@gla.ac.in)

0.2 Linear Regression with One Variable (Sklearn Library)

[1]:
```
#using sklearn library
import numpy as np
from sklearn.linear_model import LinearRegression

X = np.array([171,151,124,134,156])
y = np.array([80,60,45,50,65])

clf = LinearRegression()

#print(X)
#print(X.reshape(-1,1))

clf.fit(X.reshape(-1,1),y) # you want a COLUMN vector (many samples, 1 feature)

a=clf.predict([[200]])

print(a)
```

```
[98.51765392]
```

[22]:
```
import numpy as np
z = np.array([[1, 2, 3, 4],
              [5, 6, 7, 8],
              [9, 10, 11, 12]])
print(z.shape)
print(z.reshape(-1))
print(z.reshape(-1,1))
```

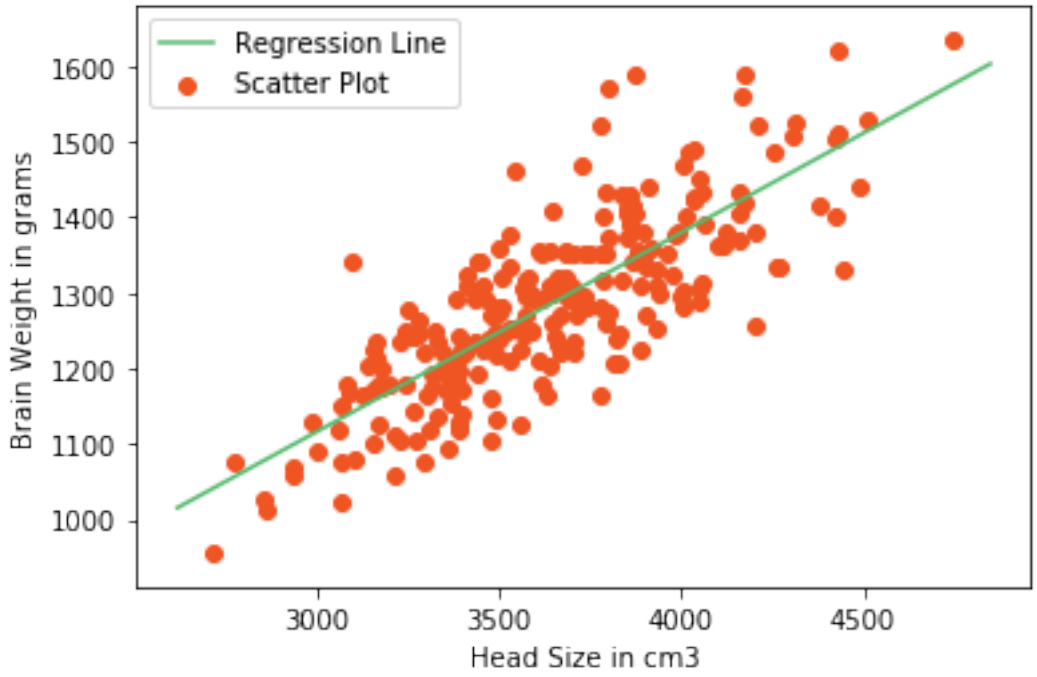
```
(3, 4)
[ 1  2  3  4  5  6  7  8  9 10 11 12]
```

```
[[ 1]
 [ 2]
 [ 3]
 [ 4]
 [ 5]
 [ 6]
 [ 7]
 [ 8]
 [ 9]
 [10]
 [11]
 [12]]
```

[23]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

# Reading Data
data = pd.read_csv('image/headbrain.csv')
data.head()

# Collecting X and Y
X = data['Head Size(cm^3)'].values
Y = data['Brain Weight(grams)'].values

# Calculating coefficient

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


# Total number of values
n = len(X)

# Using the formula to calculate b0 and b1
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)
```

```
# Printing coefficients
print("Coefficients")
print(b1, b0)

# Plotting Values and Regression Line

max_x = np.max(X) + 100
min_x = np.min(X) - 100

# 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='#58b970', label='Regression Line')
# Ploting Scatter Points
plt.scatter(X, Y, c='#ef5423', label='Scatter Plot')

plt.xlabel('Head Size in cm3')
plt.ylabel('Brain Weight in grams')
plt.legend()
plt.show()

# Calculating Root Mean Squares Error
rmse = 0
for i in range(n):
    y_pred = b0 + b1 * X[i]
    rmse += (Y[i] - y_pred) ** 2
rmse = np.sqrt(rmse/n)
print("RMSE")
print(rmse)

# 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)
```

```
3633.9915611814345
1282.873417721519
Coefficients
0.26342933948939945 325.57342104944223
```




```
RMSE
72.1206213783709
R2 Score
0.6393117199570003
```

[24]:
```python
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Reading Data
data = pd.read_csv('image\headbrain.csv')
data.head()

# Collecting X and Y
X = data['Head Size(cm^3)'].values
Y = data['Brain Weight(grams)'].values
data.iloc[0:5,0:1]

m = len(X)

X = X.reshape((m, 1))

# Model Intialization
reg = LinearRegression()
```

```
# Data Fitting
reg = reg.fit(X, Y)
# Y Prediction
Y_pred = reg.predict(X)

print('Intercept: \n', reg.intercept_)
print('Coefficients: \n', reg.coef_)

# Model Evaluation
rmse = np.sqrt(mean_squared_error(Y, Y_pred))
r2 = reg.score(X, Y)


print("RMSE")
print(rmse)
print("R2 Score")
print(r2)
```

```
Intercept:
 325.5734210494428
Coefficients:
 [0.26342934]
RMSE
72.1206213783709
R2 Score
0.639311719957
```

[25]:
```
data.head()
```

[25]:

	Gender	Age Range	Head Size(cm^3)	Brain Weight(grams)
0	1	1	4512	1530
1	1	1	3738	1297
2	1	1	4261	1335
3	1	1	3777	1282
4	1	1	4177	1590

[ ]:

[ ]:

[ ]:

[ ]: