Linear Regression with python

Prediction analysis using supervised machine learning

In [1]:
```
import warnings
warnings.simplefilter('ignore')
```

Import numpy and pandas

In [2]:
```
import numpy as np
import pandas as pd
```

Import data visualization library

In [12]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

Import the dataset

In [14]:
```
dataset = pd.read_csv('Marathondata.csv')
```

In [15]:
```
dataset
```

Out[15]:

	id	Marathon	Name	Category	km4week	sp4week	CrossTraining	Wall21	MarathonTime	CATEGORY
0	1	Prague17	Blair MORGAN	MAM	132.8	14.434783	NaN	1.16	2.37	A
1	2	Prague17	Robert Heczko	MAM	68.6	13.674419	NaN	1.23	2.59	A
2	3	Prague17	Michon Jerome	MAM	82.7	13.520436	NaN	1.30	2.66	A
3	4	Prague17	Daniel Or lek	M45	137.5	12.258544	NaN	1.32	2.68	A
4	5	Prague17	Luk ? Mr zek	MAM	84.6	13.945055	NaN	1.36	2.74	A
...	...	...	...	...	...	...	...	...	...	...
82	83	Prague17	Stefano Vegliani	M55	50.0	10.830325	NaN	2.02	3.93	D
83	84	Prague17	Andrej Madliak	M40	33.6	10.130653	ciclista 3h	1.94	3.93	D
84	85	Prague17	Yoi Ohsako	M40	55.4	11.043189	NaN	1.94	3.94	D
85	86	Prague17	Simon Dunn	M45	33.2	11.066667	NaN	2.05	3.95	D
86	87	Prague17	Pavel ?imek	M40	17.9	10.848485	ciclista 5h	2.05	3.98	D

87 rows × 10 columns

Understanding the dataset

In [16]:
```
dataset.shape
```

Out[16]: (87, 10)

In [17]:
```
dataset.head()
```

Out[17]:

	id	Marathon	Name	Category	km4week	sp4week	CrossTraining	Wall21	MarathonTime	CATEGORY
0	1	Prague17	Blair MORGAN	MAM	132.8	14.434783	NaN	1.16	2.37	A
1	2	Prague17	Robert Heczko	MAM	68.6	13.674419	NaN	1.23	2.59	A
2	3	Prague17	Michon Jerome	MAM	82.7	13.520436	NaN	1.30	2.66	A
3	4	Prague17	Daniel Or lek	M45	137.5	12.258544	NaN	1.32	2.68	A
4	5	Prague17	Luk ? Mr zek	MAM	84.6	13.945055	NaN	1.36	2.74	A

Slicing the dataset

In [44]:
```
dataset =dataset.drop(['id','Marathon','Name','Category','sp4week','CrossTraining','Wall21','CATEGORY'],axis =1)
```

In [45]:
```
dataset
```

Out[45]:

	km4week	MarathonTime
0	132.8	2.37
1	68.6	2.59
2	82.7	2.66
3	137.5	2.68
4	84.6	2.74
...	...	...
82	50.0	3.93
83	33.6	3.93
84	55.4	3.94
85	33.2	3.95
86	17.9	3.98

87 rows × 2 columns

In [46]:
```
x = dataset.iloc[:,0]
```

In [47]:
```
x.shape
```

Out[47]: (87,)

In [48]:
```
x = dataset.iloc[:,0].values.reshape(-1,1)
```

In [49]:
```
x.shape
```

Out[49]: (87, 1)

In [50]:
```
x
```

Out[50]:
```
array([[132.8],
       [ 68.6],
       [ 82.7],
       [137.5],
       [ 84.6],
       [ 42.2],
       [ 89. ],
       [106. ],
       [ 70. ],
       [ 84.2],
       [ 93.5],
       [ 65.7],
       [ 53.5],
       [ 84.4],
       [ 76.8],
       [ 76.1],
       [112.3],
       [ 49.7],
       [ 84.5],
       [ 76.7],
       [ 94.5],
       [ 67.3],
       [ 59.4],
       [ 66.1],
       [ 67.4],
       [ 23.8],
       [129.6],
       [ 82.4],
       [ 51.6],
       [104.9],
       [ 52.5],
       [ 79.4],
       [ 65.6],
       [112.4],
       [112.2],
       [ 50.1],
       [ 50.1],
       [ 52.1],
       [ 64.7],
       [ 69.2],
       [ 61.3],
       [ 58.8],
       [ 82.7],
       [ 32.2],
       [ 27.7],
       [ 68. ],
       [ 48.6],
       [ 39.6],
       [ 60.1],
       [ 78.2],
       [ 50.3],
       [ 70.7],
       [121.7],
       [ 51.1],
       [ 70.7],
       [ 26.9],
       [ 56.2],
       [ 36.3],
       [ 22.7],
       [ 45.2],
       [ 43.2],
       [ 54.1],
       [ 48.8],
       [ 20.7],
       [ 54.2],
       [ 60.3],
       [ 48.5],
       [ 34.3],
       [ 59.1],
       [ 41.6],
       [ 87. ],
       [ 24.2],
       [ 52.3],
       [ 53.6],
       [ 66.7],
       [ 23.9],
       [ 40.3],
       [ 30.6],
       [ 28. ],
       [ 53.9],
       [ 38.1],
       [ 35.6],
       [ 50. ],
       [ 33.6],
       [ 55.4],
       [ 33.2],
       [ 17.9]])
```

In [51]:
```
y = dataset.iloc[:,-1].values.reshape(-1,1)
```

In [52]:
```
y.shape
```

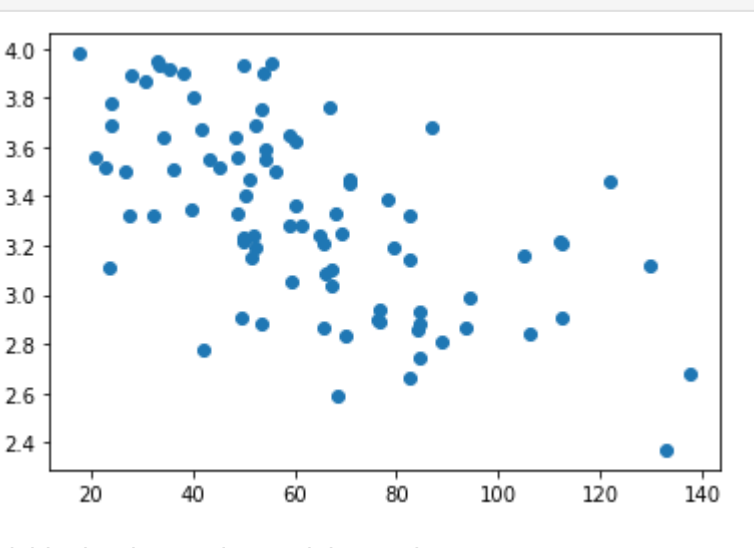
Out[52]: (87, 1)

In [53]:
```
y
```

Out[53]:
```
array([[2.37],
       [2.59],
       [2.66],
       [2.68],
       [2.74],
       [2.78],
       [2.81],
       [2.84],
       [2.83],
       [2.86],
       [2.87],
       [2.87],
       [2.88],
       [2.88],
       [2.89],
       [2.9 ],
       [2.91],
       [2.91],
       [2.93],
       [2.94],
       [2.99],
       [3.04],
       [3.05],
       [3.09],
       [3.1 ],
       [3.11],
       [3.12],
       [3.14],
       [3.15],
       [3.16],
       [3.19],
       [3.19],
       [3.21],
       [3.21],
       [3.22],
       [3.22],
       [3.23],
       [3.24],
       [3.24],
       [3.25],
       [3.28],
       [3.28],
       [3.32],
       [3.32],
       [3.32],
       [3.33],
       [3.33],
       [3.35],
       [3.36],
       [3.39],
       [3.4 ],
       [3.45],
       [3.46],
       [3.47],
       [3.47],
       [3.5 ],
       [3.5 ],
       [3.51],
       [3.52],
       [3.52],
       [3.55],
       [3.55],
       [3.56],
       [3.56],
       [3.59],
       [3.62],
       [3.64],
       [3.64],
       [3.65],
       [3.67],
       [3.68],
       [3.69],
       [3.69],
       [3.75],
       [3.76],
       [3.78],
       [3.8 ],
       [3.87],
       [3.89],
       [3.9 ],
       [3.9 ],
       [3.92],
       [3.93],
       [3.93],
       [3.94],
       [3.95],
       [3.98]])
```

Scatter plot

In [54]:
```
plt.scatter(x,y)
plt.show()
```



Divide the dataset into training and test set

In [60]:
```
from sklearn.model_selection import train_test_split
```

In [63]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.2,random_state=0)
```

In [64]:
```
x_train.shape
```

Out[64]: (69, 1)

In [65]:
```
x_test.shape
```

Out[65]: (18, 1)

In [66]:
```
y_test.shape
```

Out[66]: (18, 1)

Perform Linear regression

In [67]:
```
from sklearn.linear_model import LinearRegression
```

In [74]:
```
lm = LinearRegression()
```

Train the model Linear Regression

In [75]:
```
lm.fit(x_train,y_train)
```

Out[75]: LinearRegression()

Predict the chance

In [77]:
```
y_pred = lm.predict(x_test)
```

In [78]:
```
y_pred
```

Out[78]:
```
array([[3.1476571 ],
       [3.13291071],
       [3.42176631],
       [3.35497388],
       [3.4443196 ],
       [3.40962223],
       [3.27516993],
       [3.58571138],
       [3.51544921],
       [3.70975447],
       [2.94554492],
       [2.74083045],
       [2.89002913],
       [3.68546631],
       [3.25782124],
       [2.89089657],
       [3.28037453],
       [3.37752716]])
```

Check the prediction

In [80]:
```
check = pd.DataFrame(x_test,columns =['km4week'])
```

In [81]:
```
check['COA Actual']=y_test
```

In [82]:
```
check['COA Predicted']= y_pred
```

In [83]:
```
check
```

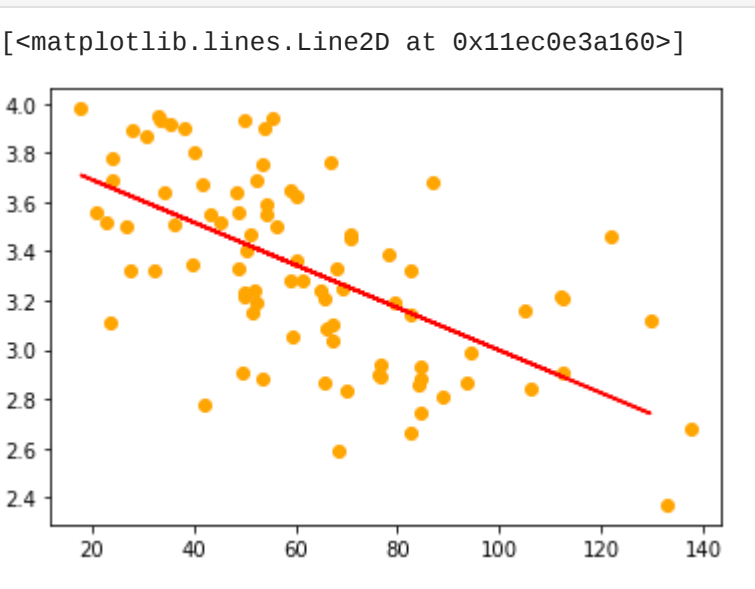
Out[83]:

	km4week	COA Actual	COA Predicted
0	82.7	2.66	3.147657
1	84.4	2.88	3.132911
2	51.1	3.47	3.421766
3	58.8	3.28	3.354974
4	48.5	3.64	3.444320
5	52.5	3.19	3.409622
6	68.0	3.33	3.275170
7	32.2	3.32	3.585711
8	40.3	3.80	3.515449
9	17.9	3.98	3.709754
10	106.0	2.84	2.945545
11	129.6	3.12	2.740830
12	112.4	3.21	2.890029
13	20.7	3.56	3.685466
14	70.0	2.83	3.257821
15	112.3	2.91	2.890897
16	67.4	3.10	3.280375
17	56.2	3.50	3.377527

Visualize the regressor line

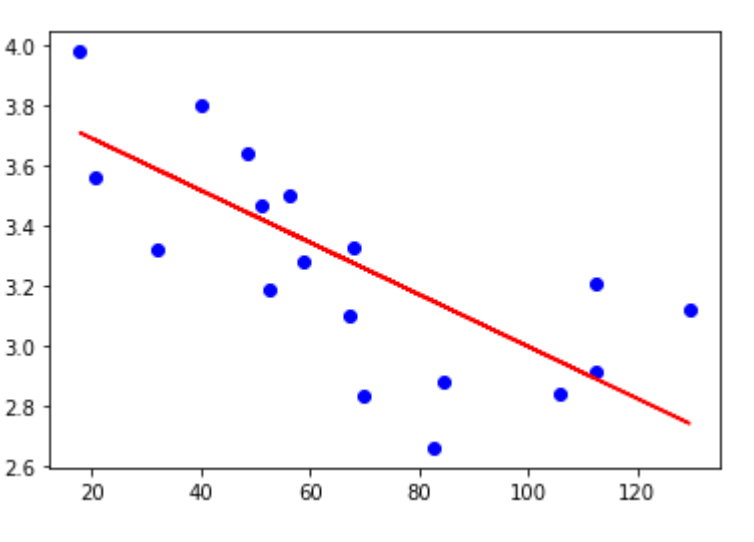
In [84]:
```
plt.scatter(x,y,color = 'orange')
plt.plot(x_test,y_pred,color ='red')
```

Out[84]: [<matplotlib.lines.Line2D at 0x11ec0e3a160>]



In [86]:
```
plt.scatter(x_test,y_test,color ='blue')
plt.plot(x_test,y_pred,color ='red')
```

Out[86]: [<matplotlib.lines.Line2D at 0x11ebe6e36a0>]



In [ ]: