Spark task 1

March 20, 2022

1 Spark intern - Task 1

2 Predict the percentage of a student based on the no.of.study hours.This is a simple linear regression task as it involves just 2 variables.What will be predicted score if a student studies for 9.25hrs/day?

[1]:

```
#Importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

[2]:

```
#Reading data
url="http://bit.ly/w-data"
data=pd.read_csv(url)
print("Data imported")
```

```
Data imported
```

[3]:

```
print(data)
```

```
    Hours  Scores
0     2.5      21
1     5.1      47
2     3.2      27
3     8.5      75
4     3.5      30
5     1.5      20
6     9.2      88
7     5.5      60
8     8.3      81
9     2.7      25
10    7.7      85
11    5.9      62
12    4.5      41
13    3.3      42
14    1.1      17
```

```
15    8.9      95
16    2.5      30
17    1.9      24
18    6.1      67
19    7.4      69
20    2.7      30
21    4.8      54
22    3.8      35
23    6.9      76
24    7.8      86
```

[4]:

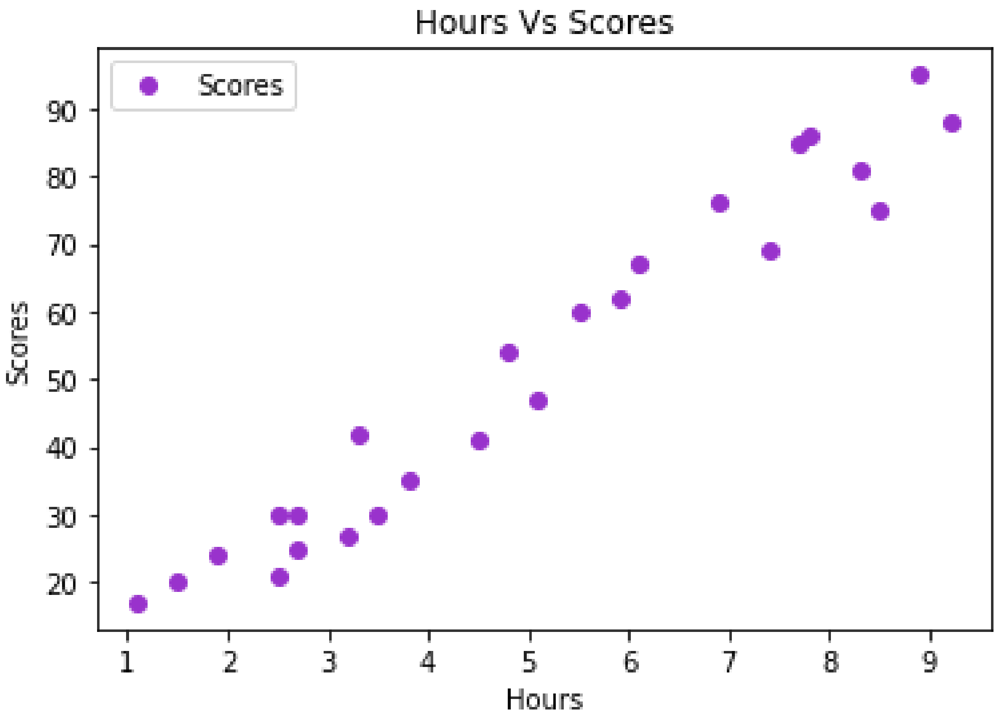
```
data.head(10)
```

[4]:

```
   Hours  Scores
0    2.5      21
1    5.1      47
2    3.2      27
3    8.5      75
4    3.5      30
5    1.5      20
6    9.2      88
7    5.5      60
8    8.3      81
9    2.7      25
```

[6]:

```
#Plotting dataset
data.plot(x="Hours",y="Scores",style="o",color="darkorchid")
plt.title("Hours Vs Scores")
plt.xlabel("Hours")
plt.ylabel("Scores")
plt.show()
```



```
[7]: #Preparing data
     x = data.iloc[:,:-1].values
     y = data.iloc[:,1].values
```

```
[8]: from sklearn.model_selection import train_test_split
     x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

```
[9]: #Training the algorithm
     from sklearn.linear_model import LinearRegression
     regression = LinearRegression()
     regression.fit(x_train,y_train)
     print("Training complete")
```

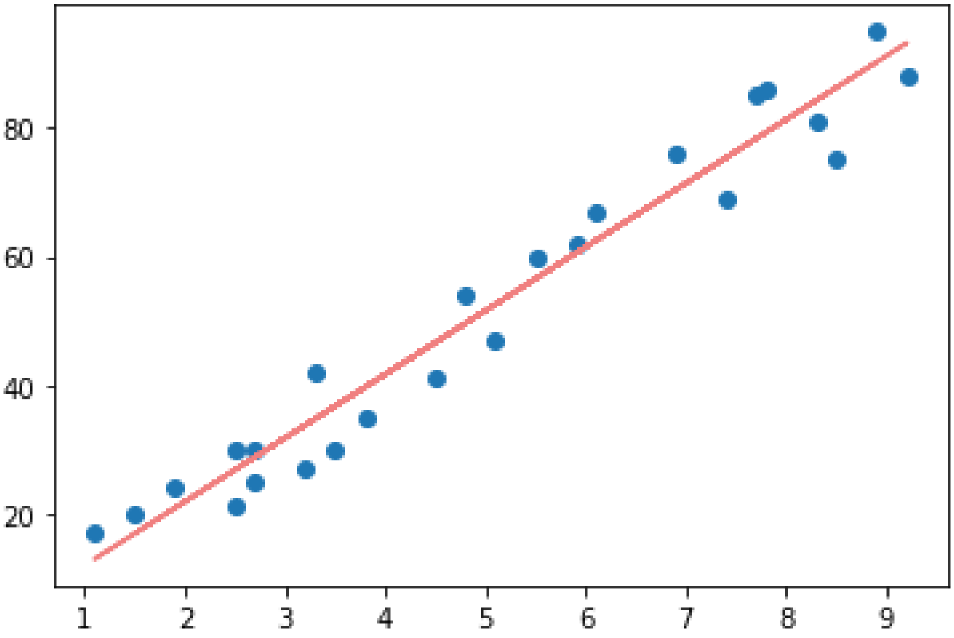
```
     Training complete
```

```
[10]: regression.coef_
```

```
[10]: array([9.91065648])
```

```
[12]: #Plotting the regression
      line = regression.coef_*x+regression.intercept_
```

```
[13]: #Plotting the test data using previously trained test data
      plt.scatter(x,y)
      plt.plot(x,line,color="lightcoral");
      plt.show()
```



[16]:
```
# Predicting the scores
print(x_test)
y_pred=regression.predict(x_test)
```

```
[[1.5]
 [3.2]
 [7.4]
 [2.5]
 [5.9]]
```

[19]:
```
#Comparing actual model vs Predicted model
data=pd.DataFrame({'Actual':y_test,'Predicted':y_pred})
data
```

[19]:
```
   Actual  Predicted
0      20  16.884145
1      27  33.732261
2      69  75.357018
3      30  26.794801
4      62  60.491033
```

[21]:
```
#Prediction for 9.25 hrs
Hours=[[9.25]]
```

```
own_pred=regression.predict(Hours)
print("No.of.Hours = {}".format(Hours))
print("Prediction Score = {}".format(own_pred[0]))
```

```
No.of.Hours = [[9.25]]
Prediction Score = 93.69173248737538
```

[22]:

```
#Evaluate the data
from sklearn import metrics
print('Mean Absolute Error:',metrics.mean_absolute_error(y_test,y_pred))
```

```
Mean Absolute Error: 4.183859899002975
```