Prediction using Supervised ML

By : Nidisha Mandlik

In this task we are going to use simple Linear Regression Alogrithm to predict score by Students based on Study hours.

statement- What will be predicted score if a student studies for 9.25 hrs/ day?

```
Data source : http://bit.ly/w-data
```

Importing all libraries that required for this project.

In [32]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

Reading data from the link.

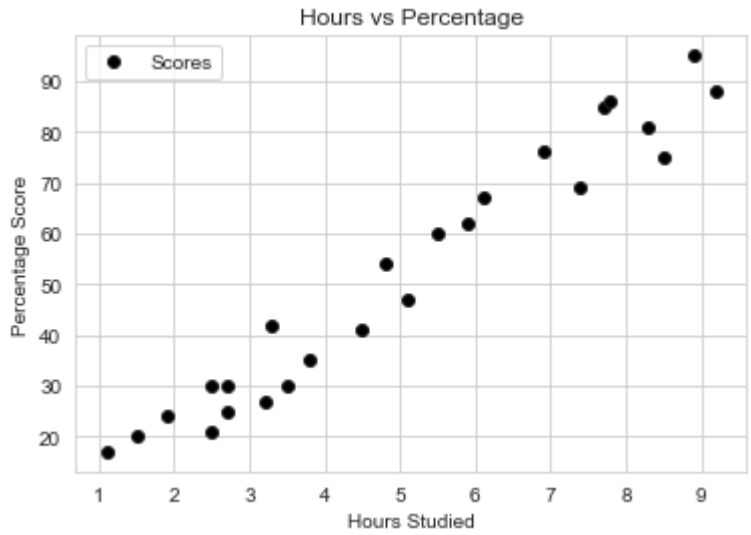
In [8]:
```
data = pd.read_csv(r'http://bit.ly/w-data')
data.head(10)
```

Out[8]:

	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

Plotting dataset to find the relationship between the data.

In [49]:
```
data.plot(x='Hours', y='Scores',style='o',markerfacecolor='k')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



From the graph above, we clearly see that there is positive linear relation between number of Hours studied (x-axis) and percentage score (y-axis).

Preparing the data

The next step is to divide the data into "attributes" (inputs) and "labels" (outputs).

In [23]:
```
X = data.iloc[:, :-1].values
y = data.iloc[:, 1].values
```

Now that we have our attributes and labels, the next step is to split this data into training and test sets. We'll do this by using Scikit-Learn's built-in train_test_split() method:

Training the Algorithm

We've split our data into training and testing sets, and now it's time to train our algorithm.

In [27]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_s
```

In [28]:
```
from sklearn.linear_model import LinearRegression
lm=LinearRegression()
lm.fit(X_train,y_train)
```

Out[28]:
```
LinearRegression()
```

It's time to make some Predictions.

In [30]:
```
predictions = lm.predict(X_test)
```

In [86]:
```
df.corr()
df
```

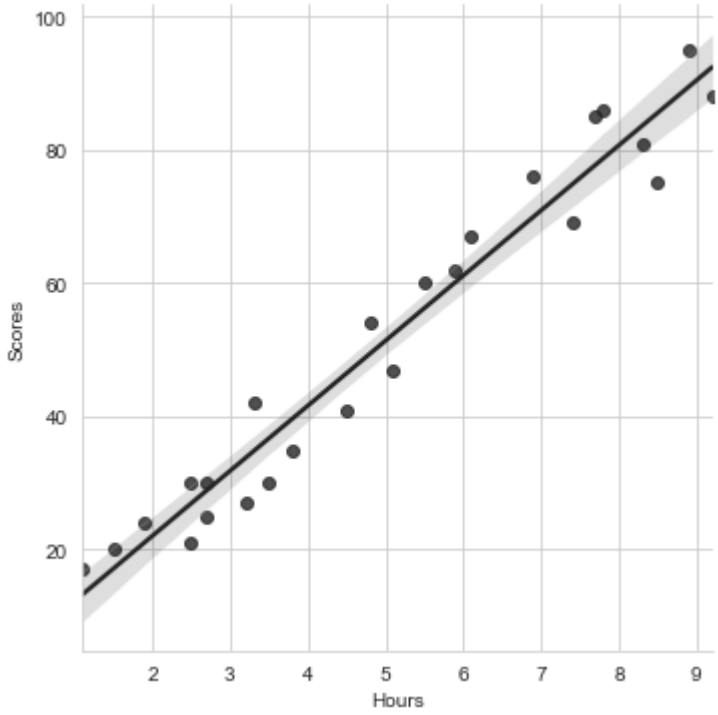
Out[86]:

	Actual	Predicted
0	30	26.753460
1	85	79.392992
2	35	39.913343
3	69	76.356096
4	60	57.122421
5	54	50.036330
6	27	33.839551
7	75	87.491382
8	47	53.073226
9	17	12.581278

Plotting the Regression Line

In [48]:
```
sns.set_palette("gray")
sns.set_style('whitegrid')
sns.lmplot(x='Hours',y ='Scores',data=data)
```

Out[48]: <seaborn.axisgrid.FacetGrid at 0x228f64c2b08>



Evaluating the model

In [64]:
```
from sklearn import metrics

print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pr
```

```
Mean Absolute Error: 5.778711665327945
Mean Squared Error: 40.3446038605835
Root Mean Squared Error: 6.351740223008455
```

Predicting the score if the student studies for 9.25 hrs/day

In [75]:
```
lm.predict([[9.25]])
```

Out[75]:
```
array([95.08362193])
```

Conclusion

If a student studies for 9.25 hrs/day then there is possibility of getting score of 95.083...

In [ ]:
```

```