By : Bhumi Shah

The Spark Foundation #GRIPJUNE2021

Task 1 : Prediction using Supervised ML

Objective : What will be predicted score if a student studies for 9.25 hrs/ day?

Dataset URL : http://bit.ly/w-data

In [1]:
```
# Importing all libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
# Reading the data
url='https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student_scores%20-%20student_scores.csv'
```

In [3]:
```
df=pd.read_csv(url)
print("Data imported successfully")
```

```
Data imported successfully
```

In [4]:
```
#to see first 5 rows of data
df.head()
```

Out[4]:

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30

In [6]:
```
#to find shape of data
df.shape
```

Out[6]: (25, 2)

In [7]:
```
#data description
df.describe()
```

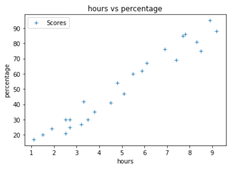
Out[7]:

	Hours	Scores
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

In [8]:
```
#info of dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Hours   25 non-null     float64
 1   Scores  25 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 528.0 bytes
```

In [9]:
```
# Plotting the relationship between hours and score
df.plot(x='Hours',y='Scores',style='+')
plt.title('hours vs percentage')
plt.xlabel('hours')
plt.ylabel('percentage')
plt.show()
```



From the above graph, we can see that there is a positive relationship between hours and score

In [10]:
```
# Divide the data into input and output
x=df.iloc[:,0:1]
y=df.iloc[:,1:]
```

Training the data

In [11]:
```
from sklearn.model_selection import train_test_split
```

In [12]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=0)
```

In [13]:
```
from sklearn.linear_model import LinearRegression
```

In [14]:
```
lr=LinearRegression()
```

In [15]:
```
lr.fit(x_train,y_train)
print("Data Trained!")
```

```
Data Trained!
```

In [16]:
```
lr.score(x_train,y_train)
```

Out[16]: 0.9484509249326872

In [17]:
```
lr.score(x_test,y_test)
```

Out[17]: 0.9367661043365056

In [18]:
```
pred=lr.predict(x_test)
```

In [20]:
```
from sklearn.metrics import mean_squared_error,mean_absolute_error
```

In [21]:
```
#finding mean square error
print(mean_squared_error(pred,y_test))
```

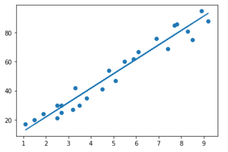
```
20.33292367497996
```

In [22]:
```
print(np.sqrt(mean_squared_error(pred,y_test)))
```

```
4.509204328368805
```

In [23]:
```
#plotting the best fit line
line = lr.coef_*x+lr.intercept_

plt.scatter(x,y)
plt.plot(x,line)
plt.show()
```



Making the prediction

In [24]:
```
df2=pd.DataFrame(y_test)
df2
```

Out[24]:

	Scores
5	20
2	27
19	69
16	30
11	62
22	35
17	24

In [25]:
```
df2['prediction']=pred
```

In [26]:
```
# Comparison between actual and predicted
df2
```

Out[26]:

	Scores	prediction
5	20	16.844722
2	27	33.745575
19	69	75.500624
16	30	26.786400
11	62	60.588106
22	35	39.710582
17	24	20.821393

In [27]:
```
#Test with your own data
hours= [[9.25]]
```

In [28]:
```
pred2=lr.predict(hours)
```

In [29]:
```
pred2
```

Out[29]: array([[93.89272889]])

If a student studies for 9.25 hours/day then he/she will score 93.89