Tanvi Bhosle

Data Science And Business Analytics Intern @ TSF GRIP JULY2021

Task 1: Pridiction using Supervised ML

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

In [ ]:

Import Libraries

In [168…

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from sklearn.metrics import r2_score
```

Import Dataset

In [169…

```
data=pd.read_csv("http://bit.ly/w-data")
data.head(5)
```

Out[169…

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

In [170…

```
data.shape
```

Out[170… (25, 2)

In [171…

```
data.describe()
```

Out[171…

	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 [172…

```
data.isnull().sum()
```

Out[172…

```
Hours     0
Scores    0
dtype: int64
```

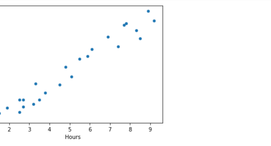
In [173…

```
data.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 [174…

```
data.plot(kind="scatter",x="Hours",y="Scores")
plt.show()
```



In [175…

```
data.corr()
```

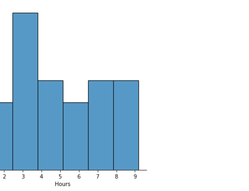
Out[175…

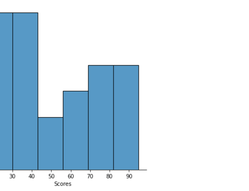
	Hours	Scores
Hours	1.000000	0.976191
Scores	0.976191	1.000000

In [176…

```
Hours=data["Hours"]
sns.displot(Hours)
Scores=data["Scores"]
sns.displot(Scores)
```

Out[176… <seaborn.axisgrid.FacetGrid at 0x1ef52d06d30>





Linear Regression

In [177…

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

In [178…

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [179…

```
reg=LinearRegression()
reg.fit(x_train,y_train)
y_predicted=reg.predict(x_test)
y_predicted
```

Out[179… array([84.74048856, 77.18873785, 18.66266991, 37.54204667, 88.51636391,
40.37395318, 34.71014016, 69.63698715])

In [180…

```
actual=pd.DataFrame({"Target":y_test,"Predicted":y_predicted})
actual
```

Out[180…

	Target	Predicted
0	75	84.740489
1	85	77.188738
2	20	18.662670
3	30	37.542047
4	95	88.516364
5	35	40.373953
6	27	34.710140
7	76	69.636987

In [181…

```
m=reg.coef_
m
```

Out[181… array([9.43968838])

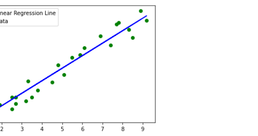
In [182…

```
b=reg.intercept_
b
```

Out[182… 4.503137347481427

In [183…

```
l=m*x+b
plt.scatter(x,y,color="green",label="Data")
plt.plot(x,l,color="blue",label="Linear Regression Line")
plt.legend(["Linear Regression Line","Data"])
plt.show()
```



In [ ]:

Trial

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

In [185…

```
print(" If a student studies for 9.25hrs/day, the prediction score is:")
reg.predict([[9.25]])
```

```
 If a student studies for 9.25hrs/day, the prediction score is:
```

Out[185… array([91.82025484])

In [186…

```
Mean_Absolute_Error=metrics.mean_absolute_error(y_test,y_predicted)
Mean_Absolute_Error
```

Out[186… 6.545233716855944

In [187…

```
r2_score=r2_score(y_test,y_predicted)
r2_score
```

Out[187… 0.9395320810589967

In [ ]: