Data Science & Bussiness Analytics @ TSF

Name : Sarthak Kumar Rath

Task - 1 : Prediction using Supervised ML

Problem statement : What will be The Predicted score if a student studies for 9.25 hours/day

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

import libraries

In [7]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

reading data

In [9]:
```
url = "http://bit.ly/w-data"
df = pd.read_csv(url)
```

In [10]:
```
df.head(5)  # display the top 5 data
```

Out[10]:

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

Exploratory Data Analysis

In [11]:
```
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 [13]:
```
df.shape
```

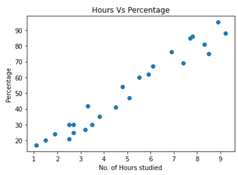
Out[13]:
```
(25, 2)
```

In [14]:
```
df.isnull().sum() #to check missing values
```

Out[14]:
```
Hours     0
Scores    0
dtype: int64
```

In [43]:
```
plt.scatter(x='Hours',y='Scores',data=df)
plt.xlabel("No. of Hours studied")
plt.ylabel("Percentage")
plt.title("Hours Vs Percentage")
```

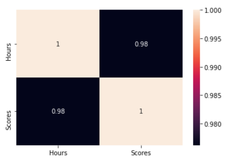
Out[43]:
```
Text(0.5, 1.0, 'Hours Vs Percentage')
```



A linear relationship between no. of hours studied and percentage secured is shown

In [19]:
```
sns.heatmap(df.corr(),annot=True)
```

Out[19]:
```
<AxesSubplot:>
```



No.of Hours studied is highly correlated with the percentage score

In [20]:
```
X= df.drop("Scores",axis = 1) #input
```

In [21]:
```
y=df["Scores"]   #output
```

In [23]:
```
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)
```

Splitting 80% of the dataset into training data and 20% into test data

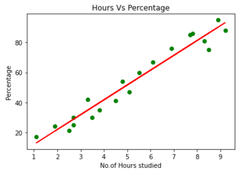
Training Model

In [29]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(X_train,y_train)
```

Out[29]:
```
LinearRegression()
```

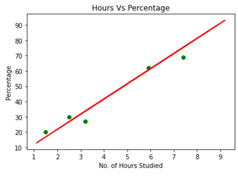
In [33]:
```
line=lr.coef_*X_train+lr.intercept_
plt.scatter(X_train,y_train,color="green")
plt.plot(X_train,line,color="red")
plt.xlabel('No.of Hours studied')
plt.ylabel("Percentage")
plt.title("Hours Vs Percentage")
```

Out[33]:
```
Text(0.5, 1.0, 'Hours Vs Percentage')
```



In [34]:
```
plt.scatter(X_test,y_test,color="green")
plt.plot(X_train,line,color="red")
plt.xlabel("No. of Hours Studied")
plt.ylabel("Percentage")
plt.title("Hours Vs Percentage")
```

Out[34]:
```
Text(0.5, 1.0, 'Hours Vs Percentage')
```



Making Prediction

In [35]:
```
y_pred=lr.predict(X_test)  #predicting the score
```

In [36]:
```
data1 = pd.DataFrame({"Actual":y_test,"Predicted": y_pred})
data1.head()
#comparing actual vs predicted scores
```

Out[36]:

	Actual	Predicted
5	20	16.884145
2	27	33.732261
19	69	75.357018
16	30	26.794801
11	62	60.491033

In [37]:
```
hrs=[9.25]
my_pred = lr.predict([hrs])
print("No. of Hours = {}".format(hrs))
print("Predicted Score = {}".format(my_pred))
```

```
No. of Hours = [9.25]
Predicted Score = [93.69173249]
```

Model

In [41]:
```
from sklearn import metrics
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error

print("r2 error :", r2_score(y_test,y_pred))
print("MSE:", mean_squared_error(y_test,y_pred))
print("MAE:", mean_absolute_error(y_test,y_pred))
```

```
r2 error : 0.9454906892105356
MSE: 21.598769307217474
MAE: 4.183859899002975
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: