BUSINESS ANALYTICS INTERNSHIP

TASK-1 SCORE PREDICTION using supervised learning

Author- Gummadi Vinila

In [37]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
```

In [38]:

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

In [39]:

```
data.head(5)
```

Out[39]:

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

In [40]:

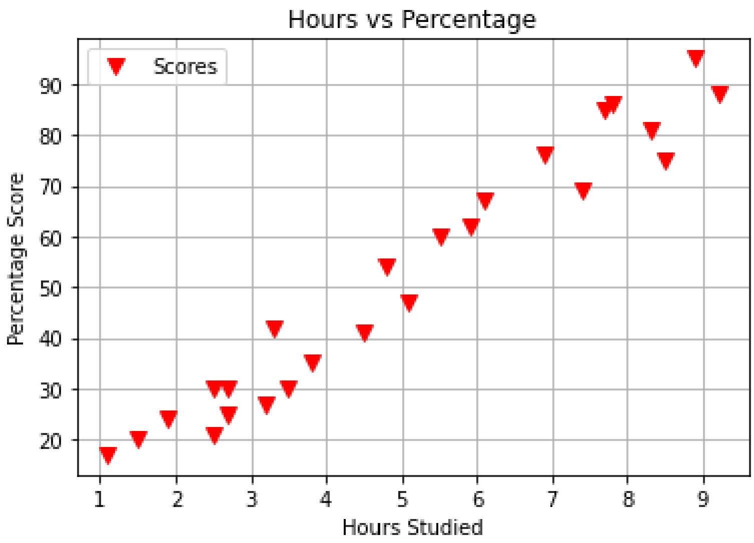
```
data.tail(5)
```

Out[40]:

	Hours	Scores
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

In [41]:

```
data.plot(x='Hours', y='Scores', style='v', color='red', markersize=8)
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.grid()
plt.show()
```



In [42]:

```
from sklearn.model_selection import train_test_split
x=data.iloc[:,:-1]
y=data.iloc[:,1]
x_train ,x_test ,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=0)
```

data fitting into a model: linear regression of two variables(independent variable-x and dependent variable-y)

In [43]:

```
from sklearn.linear_model import LinearRegression

model= LinearRegression()
model.fit(x_train, y_train)
```

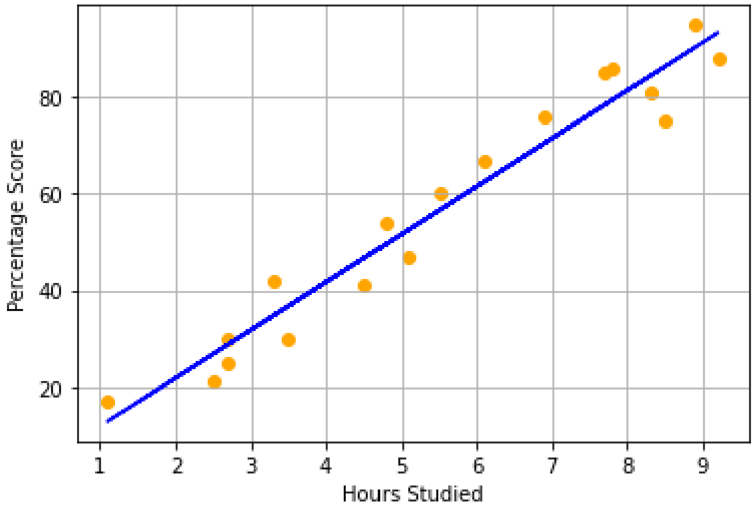
Out[43]:

```
LinearRegression()
```

In [61]:

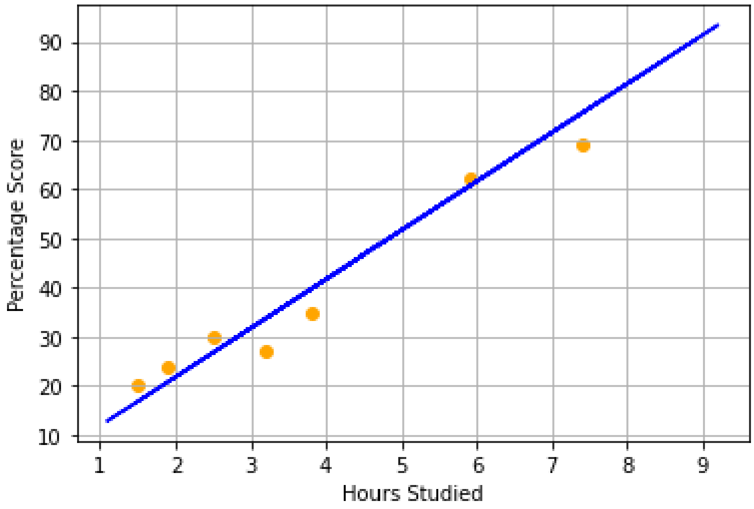
```
line = model.coef_*x + model.intercept_

# visualizing of training data
plt.scatter(x_train, y_train, color='orange')
plt.plot(x, line, color='blue');
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.grid()
plt.show()
```



In [45]:

```
# visualizing of testing data
plt.scatter(x_test, y_test, color='orange')
plt.plot(x, line, color='blue');
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.grid()
plt.show()
```



In [46]:

```
# predicting the test_score(y)
y_predicted = model.predict(x_test)

# data representation of the actual score and predicted scores
df = pd.DataFrame({'Actual score': y_test, 'Predicted score': y_predicted})
df
```

Out[46]:

	Actual score	Predicted score
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 [53]:

```
# knowing the error of the model and the good fit or accuracy score by r2_score func
from sklearn import metrics

print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_predicted))

print('the good_fit score is:', metrics.r2_score(y_test, y_predicted))
```

```
Mean Absolute Error: 4.130879918502482
the good_fit score is: 0.9367661043365056
```

In [59]:

```
#give any input(hours of study) to predict the score with the trained model

hrs = float(input('please enter the number of hours:'))
```

```
own_prediction = model.predict([[hrs]])
print("The predicted score if a person studies for",hrs,"hours is",own_prediction[0]
```

```
please enter the number of hours:9.6
The predicted score if a person studies for 9.6 hours is 97.37231631223736
```

In [ ]: