jupyter data science task Last Checkpoint: Yesterday at 12:08 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Markdown

GRIP: The Spark Foundation-MARCH 2023
NAME- Manju Thakur

**Data Science and Business Analytics Intern
Task-1:Prediction using Supervised ML
Predict the percentage of a student based on the number of study hours.

**IMPORTING LIBRARIES

In [10]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [11]:
```
data=pd.read_csv("C:/Users/USER - HP/Documents/.ipynb_checkpoints/data-st.csv")
print("Importing Data Successfully")
```

```
Importing Data Successfully
```

In [12]:
```
print("For this we print first 10 data of Data-set")
data.head(10)
```

```
For this we print first 10 data of Data-set
```

Out[12]:

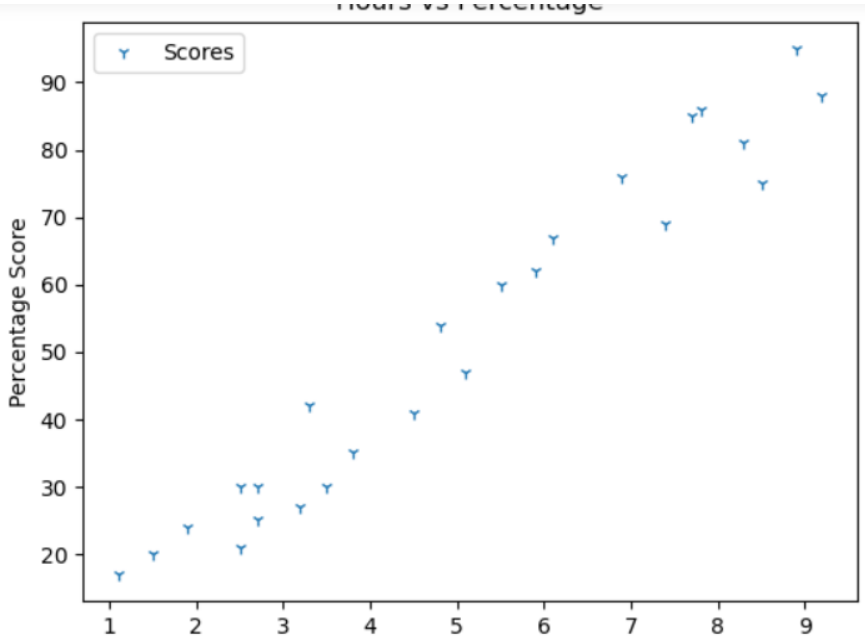
	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20
6	9.2	88
7	5.5	60
8	8.3	81
9	2.7	25

In [13]:
```
print("We have successfully imported Data-set")
```

```
We have successfully imported Data-set
```

**Now,let's plot the points on graph to see the relationship between the data points.

In [15]:
```
data.plot(x='Hours',y='Scores',style='1')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



**we can see that there is a linear relationship between the hours studied and scores.

**Now for dividing the data into attributes and labels, attributes will be "Hours" and labels will be "Scores".

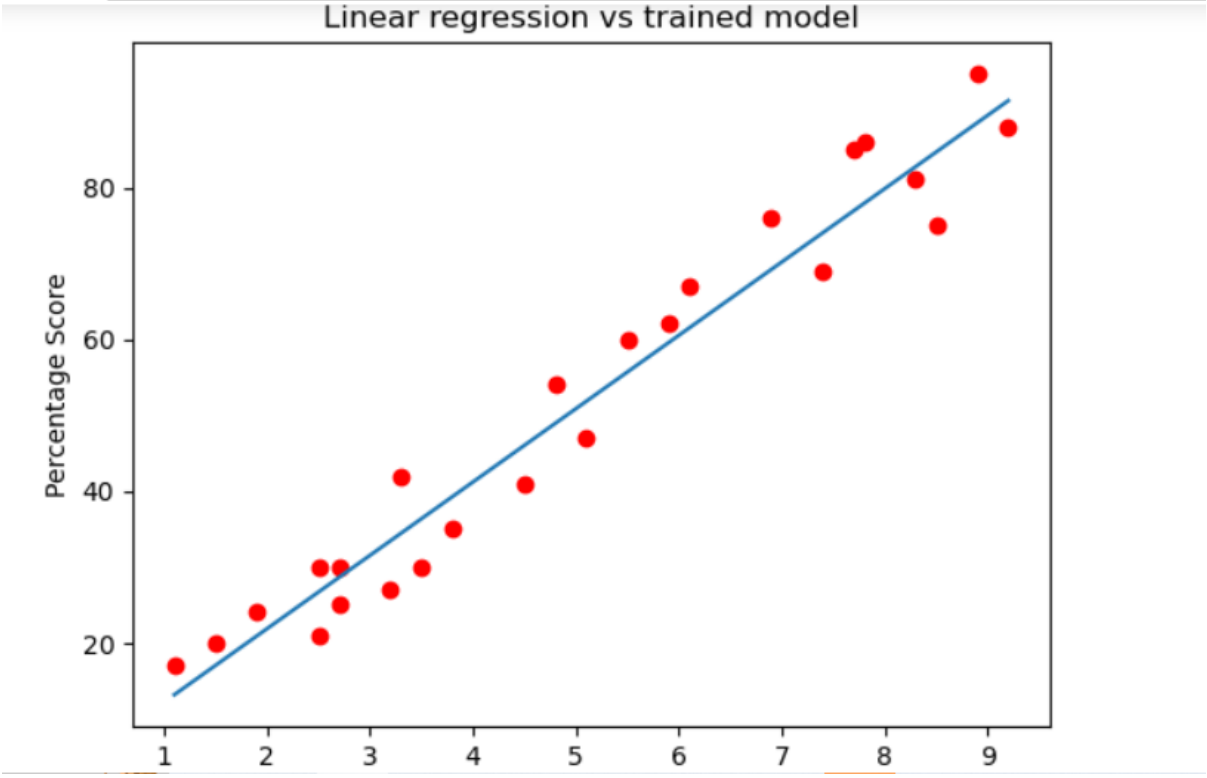
In [20]:
```
x=data.iloc[:,:-1].values
y=data.iloc[:,1].values
```

In [35]:
```
x
```

Out[35]:
```
array([[1.1],
       [1.5],
       [1.9],
       [2.5],
       [2.5],
       [2.7],
       [2.7],
       [3.2],
       [3.3],
       [3.5],
       [3.8],
       [4.5],
       [4.8],
       [5.1],
       [5.5],
       [5.9],
       [6.1],
       [6.9],
       [7.4],
       [7.7],
       [7.8],
       [8.3],
       [8.5],
       [8.9],
       [9.2]])
```

In [36]:
```
y
```

In [40]:
```
line=regressor.coef_*x+regressor.intercept_
plt.title("Linear regression vs trained model")
plt.scatter(x,y,color='red')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.plot(x,line)
plt.show()
```



**Predictions

In [41]:
```
print(x_test)
print("Prediction of Score")
y_pred=regressor.predict(x_test)
print(y_pred)
```

```
[[2.7]
 [1.9]
 [7.7]
 [6.1]
 [4.5]]
Prediction of Score
[28.6177145  20.88803334 76.92822173 61.46885942 46.0094971 ]
```

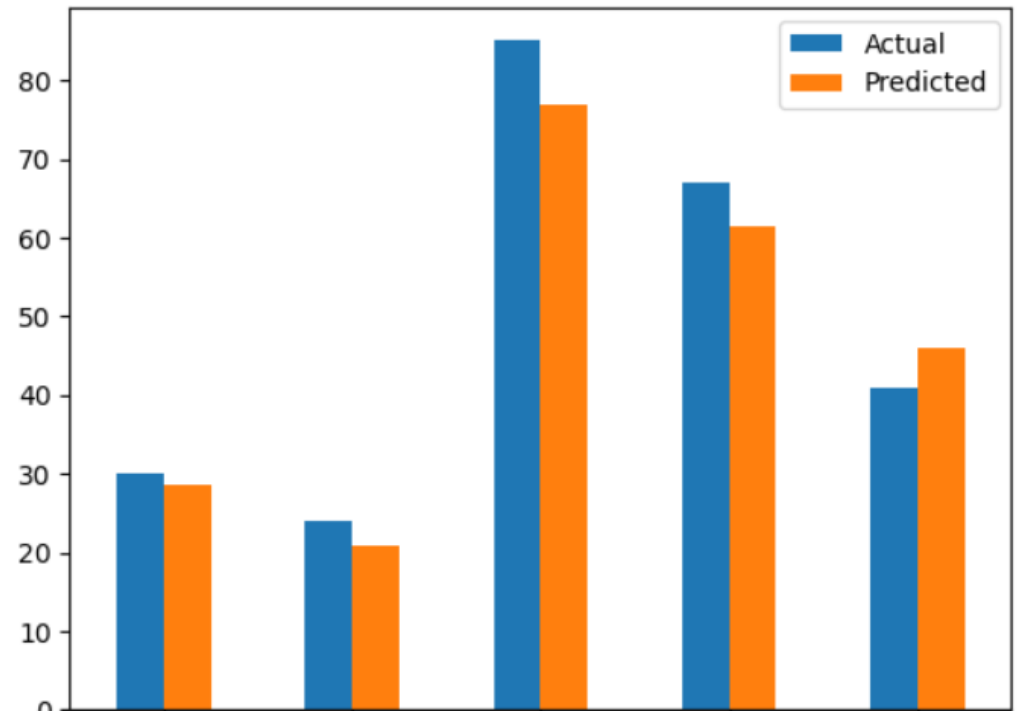
In [30]:
```
df=pd.DataFrame({'Actual':y_test,'Predicted':y_pred})
df
```

Out[30]:

	Actual	Predicted
0	30	28.617714
1	24	20.888033
2	85	76.928222
3	67	61.468859
4	41	46.009497

In [42]:
```
df.plot(kind='bar')
```

Out[42]: <AxesSubplot:>



In [43]:
```
hours=[[9.25]]
pred=regressor.predict(hours)
print(pred)
```

```
[91.90447898]
```

In [51]:
```
hours=9.25
test=np.array([hours])
test=test.reshape(-1,1)
pred=regressor.predict([[9.5]])
print("No. of Hours={}".format(hours))
print("Predicted score={}".format(pred[0]))
```

```
No. of Hours=9.25
Predicted score=94.32000433799152
```

****Final step: Evaluating the Model.**
This step is important to evaluate the performance of our algorithm

In [47]:
```
from sklearn import metrics
```

In [49]:
```
print('Mean Absolute Error is:',metrics.mean_absolute_error(y_test,y_pred))
print('Mean squared Error is:',metrics.mean_squared_error(y_test,y_pred))
print('Root Mean squared Error is:',np.sqrt(metrics.mean_absolute_error(y_test,y_pred)))
```

```
Mean Absolute Error is: 4.621333622532769
Mean squared Error is: 26.487446288873382
Root Mean squared Error is: 2.1497287323131653
```

In [50]:
```
print("Slope of Regression Line",regressor.coef_)
print("Y-intercept of Regression Line",regressor.intercept_)
```

```
Slope of Regression Line [9.66210145]
Y-intercept of Regression Line 2.5300405912062587
```