Name: Diya Gandhi

The Sparks Foundation GRIPJUNE22

#TASK 1: Pridiction Using Supervised ML

Predict The percentage of an student based on the no. of study hours

Import The Libraries

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

In [2]:
```
data = pd.read_csv("data.csv")
print("Importing data Succesfully")
```

```
Importing data Succesfully
```

=> check wheather data imported successfully or not

In [3]:
```
print("For this we print first 10 data of our data set")
data.head(10)
```

```
For this we print first 10 data of our data set
```

Out[3]:

	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 [6]:
```
#Check For any missing values
data.isnull().sum()
```

Out[6]:
```
Hours     0
Scores    0
dtype: int64
```

In [7]:
```
data.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

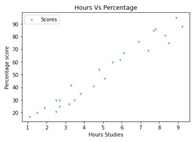
In [8]:
```
# check the Correlation between Hours and scores
data.corr()
```

Out[8]:

	Hours	Scores
Hours	1.000000	0.976191
Scores	0.976191	1.000000

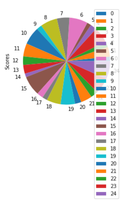
Plot The graph for detailed analysis of our dataset

In [9]:
```
data.plot(x="Hours",y="Scores",style="1")
plt.title("Hours Vs Percentage")
plt.xlabel("Hours Studies")
plt.ylabel("Percentage score")
plt.show()
```



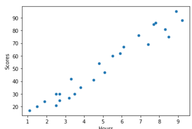
In [10]:
```
data.plot.pie(x='Hours',y='Scores')
```

Out[10]:
```
<AxesSubplot:ylabel='Scores'>
```



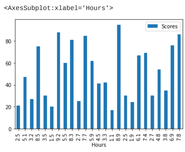
In [11]:
```
data.plot.scatter(x='Hours',y="Scores")
```

Out[11]:
```
<AxesSubplot:xlabel='Hours', ylabel='Scores'>
```



In [12]:
```
data.plot.bar(x="Hours",y="Scores")
```

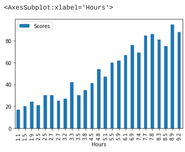
Out[12]:
```
<AxesSubplot:xlabel='Hours'>
```



=> Now plot The data in Ascending order

In [13]:
```
data.sort_values(["Hours"],axis=0,
                 ascending=[True],inplace=True)
data.head(10)
data.plot.bar(x="Hours",y="Scores")
```

Out[13]:
```
<AxesSubplot:xlabel='Hours'>
```



After ploting different graphs, we can say that as study hours increases score also increases.So, it is a good sign of a correct data.

In [14]:
```
print("So now we have to prepare our data according to the model we have ")
```

```
So now we have to prepare our data according to the model we have
```

In [15]:
```
X = data.iloc[:, :-1].values
y = data.iloc[:, 1].values
# print(X)
```

In [16]:
```
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)
```

Train The Algorithm

In [17]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()


# from sklearn.ensemble import RandomForestRegressor
# regressor = RandomForestRegressor(n_estimators = 1000, random_state = 42)


regressor.fit(X_train, y_train)
print("Training complete.")
```

```
Training complete.
```

=> Now our model is ready. Its Time to Test it

In [18]:
```
print(X_test)
print("Predection of Score")
y_pred = regressor.predict(X_test)
print(y_pred)
```

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

=> Now check the accuracy of our model

In [19]:
```
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})

df
```

Out[19]:

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

In [23]:
```
hours = [[9.25]]
pred = regressor.predict(hours)
print("Predicted score= [91.90447898]", pred)
```

```
Predicted score= [91.90447898]
```

Evaluate The Model

In [21]:
```
from sklearn import metrics
print('Mean Absolute Error:',
      metrics.mean_absolute_error(y_test, y_pred))
```

```
Mean Absolute Error: 4.621333622532769
```

In [ ]:
```

```