Supervised Machine Learning

Importing libraries

In [66]:

```
import numpy as np
from matplotlib import pyplot as plt
from sklearn import datasets,linear_model
from sklearn.metrics import mean_squared_error
```

Reading the data set

In [67]:

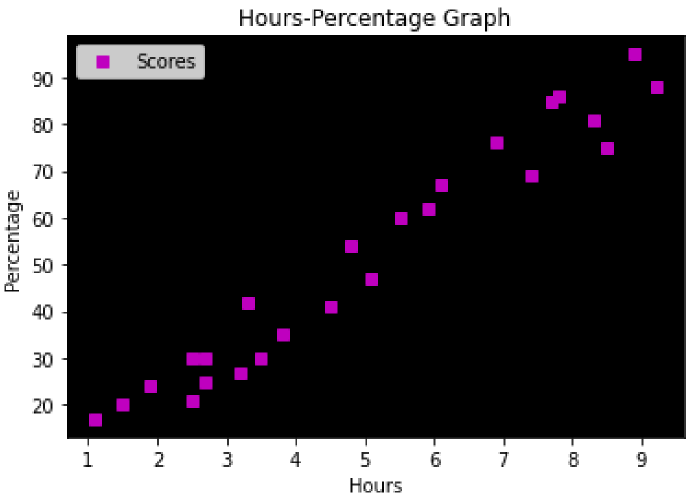
```
import pandas as pd
data_url='http://bit.ly/w-data'
data = pd.read_csv(data_url)
print(data.head(10))
```

```
   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
```

Visualising the relationship between the data

In [68]:

```
from matplotlib import pyplot as plt
a=data.plot(x="Hours", y="Scores", style="ms")
a.set_facecolor('black')
plt.title("Hours-Percentage Graph")
plt.xlabel("Hours")
plt.ylabel("Percentage")
plt.show()
```



the graph shows a positive linear relation between the number of hours studied and the percentage score

Prepairing data for testing and training

In [69]:

```
from sklearn.model_selection import train_test_split
x=data.iloc[:,:-1].values
y=data.iloc[:,1].values
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

Model fitting

In [70]:

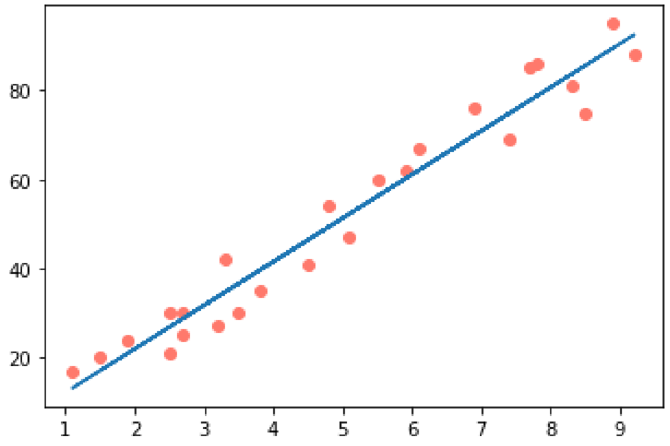
```
from sklearn.linear_model import LinearRegression
model=linear_model.LinearRegression()
model.fit(x_train,y_train)
print("model successfully fitted")
```

```
model successfully fitted
```

Plotting the Regression Line

In [71]:

```
from sklearn import linear_model
regression_line = model.coef_*x+model.intercept_
b=plt.scatter(x, y)
b.set_facecolor('xkcd:salmon')
plt.plot(x, regression_line)
plt.show()
```



Predicting the data

In [72]:

```
print("data for testing=", x_test)
data_pred=model.predict(x_test)
```

```
data for testing= [[1.5]
 [3.2]
 [7.4]
 [2.5]
 [5.9]
 [3.8]
 [1.9]
 [7.8]]
```

In [73]:

```
# Comparing the actual and predicted data
df = pd.DataFrame({'Actual': y_test, 'Predicted': data_pred})
print(df)
```

```
   Actual  Predicted
0      20  17.053665
1      27  33.694229
2      69  74.806209
3      30  26.842232
4      62  60.123359
5      35  39.567369
6      24  20.969092
7      86  78.721636
```

Testing our own data

In [74]:

```
hours_studied=[[9.25]]
prediction = model.predict(hours_studied)
print("no of hours studied={}".format(hours_studied))
print("predicted score={}".format(prediction[0]))
```

```
no of hours studied=[[9.25]]
predicted score=92.91505723477056
```

Evaluating the performance of the model

In [76]:

```
from sklearn import metrics
print('mean absolute error=', metrics.mean_absolute_error(y_test,data_pred))
```

```
mean absolute error= 4.419727808027652
```

In [79]:

```
print('mean squared error=', metrics.mean_squared_error(y_test,data_pred))
```

```
mean squared error= 22.96509721270043
```