

DATA SCIENCE & BUSINESS ANALYTICS

THE SPARKS FOUNDATION

NAME:ANALA VENKATA SATHWIKA

Prediction using Supervised ML

TASK1

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

Importing the dataset

```
url = "https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student_scores%20-%20"
df = pd.read_csv(url)
df
```



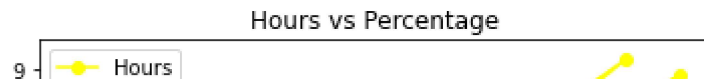
	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20
6	9.2	88
7	5.5	60

```
df.head()
```

	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

Visualizing the dataset

```
df.plot(x='Scores', y='Hours', marker='o',color='yellow')
plt.title('Hours vs Percentage')
plt.xlabel('Percentage')
plt.ylabel('Hours studied')
plt.show()
```



Preparation of data

```
'1
X = df.iloc[:, :-1].values
y = df.iloc[:, 1].values

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)

14
```

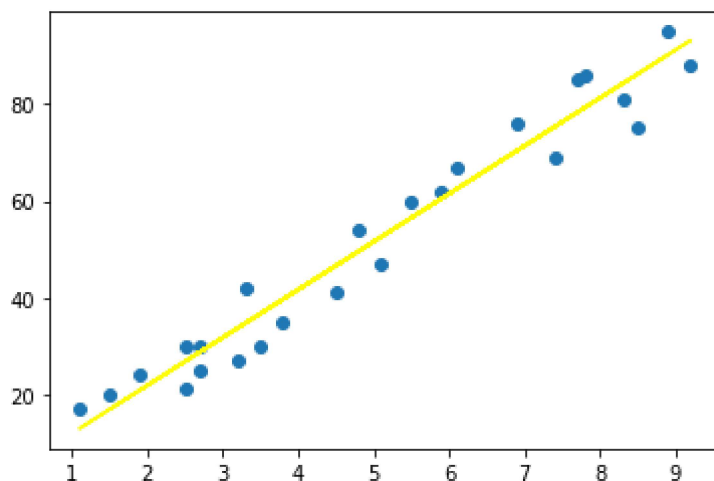
Training the Algorithm

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

print("Training complete.")
```

Training complete.

```
line = regressor.coef_*X+regressor.intercept_
plt.scatter(X, y)
plt.plot(X, line,color='yellow');
plt.show()
```



Making Predictions

```
print(X_test)
y_pred = regressor.predict(X_test)
```

```
[[1.5]
```

```
[3.2]
[7.4]
[2.5]
[5.9]]
```

```
df1 = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df1
```

	Actual	Predicted
0	20	16.884145
1	27	33.732261
2	69	75.357018
3	30	26.794801
4	62	60.491033

Testing with own data

What will be predicted score if a student studies for 9.25 hrs/ day?

```
hours = 9.25
own_predridiction = regressor.predict([[hours]])
print("No of Hours = {}".format(hours))
print("Predicted Score = {}".format(own_predridiction[0]))
```

```
No of Hours = 9.25
Predicted Score = 93.69173248737539
```

Evaluating the data

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

```
Mean Absolute Error: 4.183859899002982
```

