

# Mohammed Kayser

## GRIPSEPT'22 @The Sparks Foundation

Data Science and Business Analytics Intern

### Task 1: Prediction using supervised ML

Predict the percentage of students scores based on number of study hours

```
In [1]: !pip install jupyterthemes

Unable to create process using 'C:\Users\Mohammed_Kayser\anaconda3\python.exe "C:\Users\Mohammed_Kayser\anaconda3\scripts\pip-script.py" install jupyterthemes'
```

### Summary

```
In [2]: #1.importing the necessary libraries
#2.importing the data
#3.understanding the data structure
#4.visualizing the data
#5.splitting the data into inputs & outputs
#6.splitting the data into train & test
#7.training the algorithm
#8.prediction
```

### 1.Import necessary libraries

```
In [3]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

### 2.Importing data

```
In [4]: url = "http://bit.ly/w-data"
data = pd.read_csv(url)
```

### 3.Info,Shape,Describe,Nulls

```
In [5]: data.head(10) #view the data
```

Out[5]:

	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

Check the number of rows and columns

```
In [6]: data.shape
```

Out[6]: (25, 2)

```
In [7]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype
---  -
0   Hours   25 non-null    float64
1   Scores  25 non-null    int64
dtypes: float64(1), int64(1)
memory usage: 528.0 bytes
```

```
In [8]: data.describe(include="all")
```

Out[8]:

	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

Check for duplicates

```
In [9]: dups = data.duplicated()
dups.sum()
```

Out[9]: 0

Observation: There are no duplicates

```
In [10]: data.isnull().sum()
```

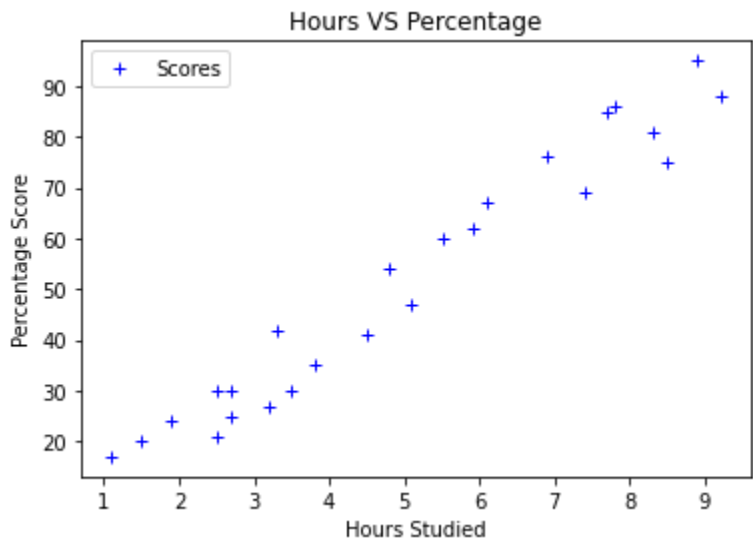
Out[10]: Hours 0
Scores 0
dtype: int64

Observation: There are no null values.

### 4.Data Visualization

Plot the data to see whether there is a relation between the data.

```
In [11]: data.plot(x = "Hours", y = "Scores", style = "+", color='blue')
plt.title("Hours VS Percentage")
plt.xlabel("Hours Studied")
plt.ylabel("Percentage Score")
plt.show()
```



Observation : There is a positive linear relation between hours studied and percentage score.

### 5.Splitting the data into inputs and outputs

```
In [12]: X = data.drop(["Scores"], axis = 1)
y = data["Scores"]
```

```
In [13]: X.head()
```

Out[13]:

	Hours
0	2.5
1	5.1
2	3.2
3	8.5
4	3.5

```
In [14]: y.head()
```

Out[14]: 0 21
1 47
2 27
3 75
4 30
Name: Scores, dtype: int64

### 6.Splitting data into train and test sets

```
In [15]: from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 0)
```

### 7.Training Algorithm

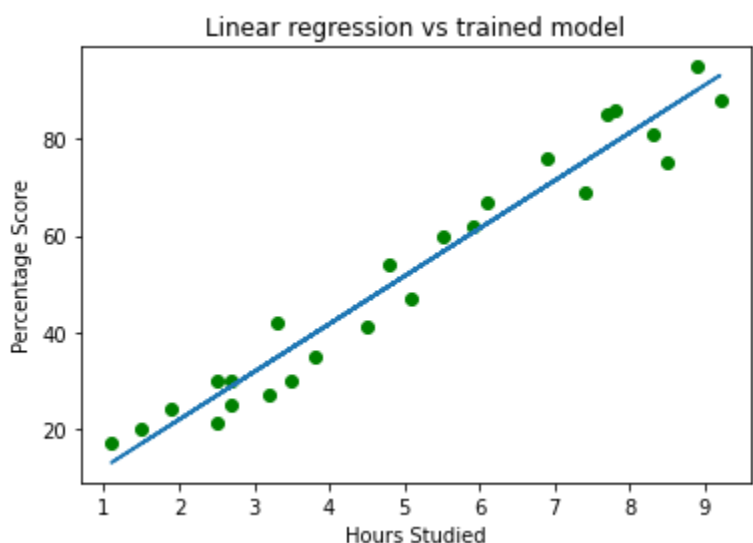
```
In [16]: from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[16]: LinearRegression()

```
In [17]: print(model.coef_, model.intercept_)
```

Out[17]: [9.81865648] 2.018160041434683

```
In [18]: line = model.coef_*X+model.intercept_
plt.title("Linear regression vs trained model")
plt.scatter(X, y, color='green')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.plot(X, line)
plt.show()
```



```
In [19]: #check the accuracy scores for training and test set
print("Training Score ", model.score(X_train, y_train))
print("Testing Score ", model.score(X_test, y_test))
```

Training Score 0.9515510725211552
Testing Score 0.9454906892105356

### 8.Predicting Scores for 9.25 hours of study

```
In [20]: hours = 9.25
test = np.array([hours])
test = test.reshape(-1,1)
pred = model.predict([test])
print("NO. of hours = {}".format(hours))
print("Predicted Score = {}".format(pred[0]))

NO. of hours = 9.25
Predicted Score = 96.16939660753593
```

INFERENCE: For 9.25 hours of study the predicted percentage scores is 96.169

```
In [ ]:
```

```
In [ ]:
```