

TASK 1 : PREDICTION USING SUPERVISED ML

SUBMITTED BY RIDHANYA S

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

```
In [1]: # IMPORTED NECESSARY LIBRARIES

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error
```

```
In [2]: # LOADING THE DATASET

df = pd.read_csv("http://bit.ly/w-data")
print("----- Data Loaded Successfully -----")

----- Data Loaded Successfully -----
```

```
In [3]: # PREVIEW OF THE DATASET

df.head(10)
```

```
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 [4]: # DIMENSION OF THE DATASET

df.shape
```

```
Out[4]: (25, 2)
```

```
In [5]: #CHECK THE MISSING VALUES

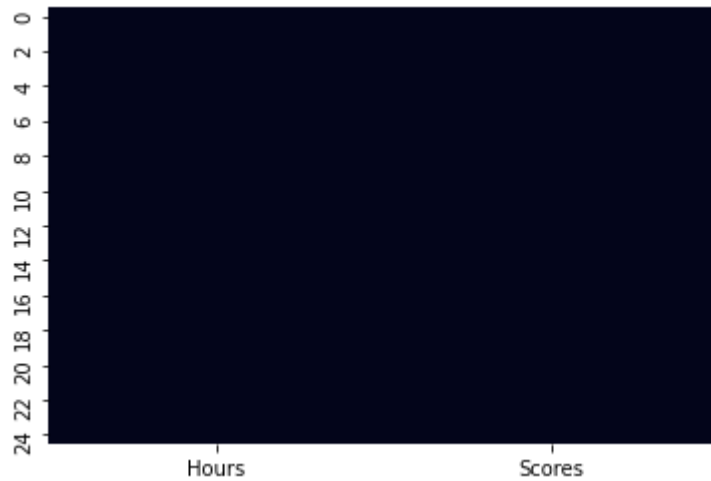
print(df.isnull().sum())

sns.heatmap(df.isnull(),cbar=False)
```

```
Hours      0
Scores     0
```

```
dtype: int64
```

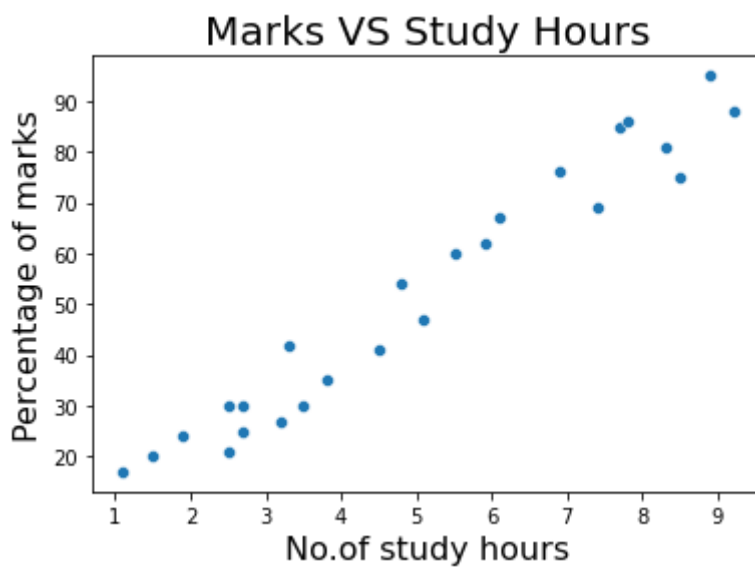
```
Out[5]: <AxesSubplot:>
```



It shows that the dataset does not contains any missing values

```
In [6]: # SCATTER PLOT FOR MARKS VS STUDY HOURS

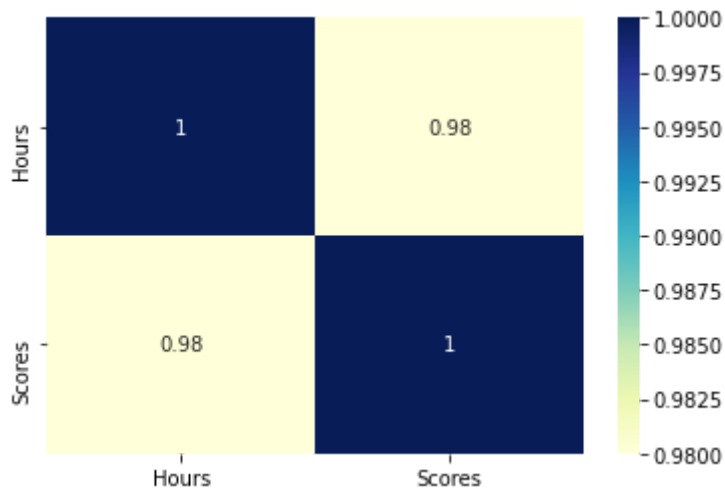
sns.scatterplot(data=df, x='Hours', y='Scores')
plt.title("Marks VS Study Hours",size=20)
plt.xlabel("No.of study hours",size=16)
plt.ylabel("Percentage of marks",size=16)
plt.show()
```



```
In [7]: corr= df.corr().round(2)
sns.heatmap(data=corr, annot=True,cmap="YlGnBu")
print(df.corr())
```

```

      Hours  Scores
Hours  1.000000  0.976191
Scores 0.976191  1.000000
```



Heat map shows that the data was highly correlated

```
In [8]: # DEFINING X AND Y FROM THE DATA
```

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

```
In [9]: # SPLITTING THE DATA INTO TRAIN AND TEST SETS
```

```
X_train,X_test,Y_train,Y_test=train_test_split(X,y,random_state=0)
```

```
In [10]: # SIMPLE LINEAR REGRESSION MODEL
```

```
linreg=LinearRegression()

# FITTING TRAINING DATA

linreg.fit(X_train,Y_train)
print("----- Model Trained -----")
```

```
----- Model Trained -----
```

```
In [11]: # PREDICTION ON TEST DATA
```

```
Y_pred=linreg.predict(X_test)
Y_pred
```

```
Out[11]: array([16.84472176, 33.74557494, 75.50062397, 26.7864001 , 60.58810646,
                39.71058194, 20.8213931 ])
```

```
In [12]: # COMPARING THE ACTUAL AND PREDICTED SCORES
```

```
compare_scores = pd.DataFrame({'Actual Marks': Y_test, 'Predicted Marks': Y_pred})
compare_scores
```

```
Out[12]:
```

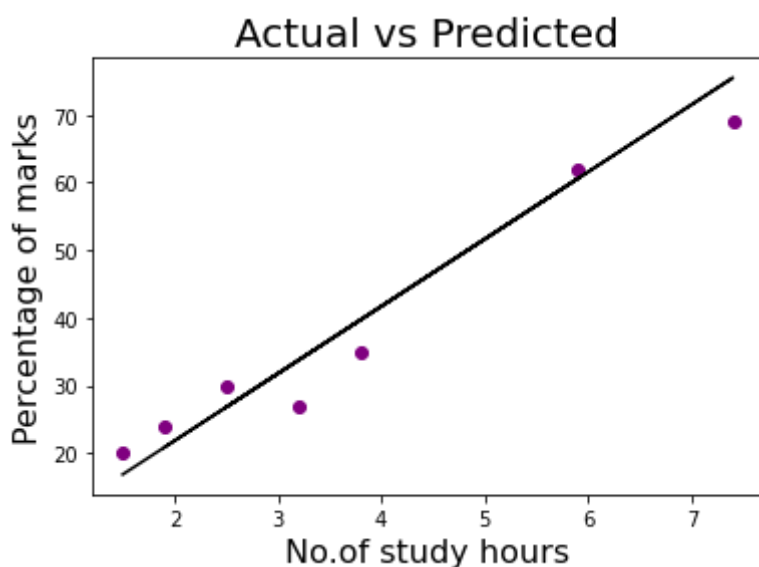
	Actual Marks	Predicted Marks
0	20	16.844722
1	27	33.745575
2	69	75.500624
3	30	26.786400
4	62	60.588106
5	35	39.710582

	Actual Marks	Predicted Marks
0	20	16.844722
1	27	33.745575
2	69	75.500624
3	30	26.786400
4	62	60.588106
5	35	39.710582

	Actual Marks	Predicted Marks
6	24	20.821393

```
In [13]: # VISUALIZE THE ACTUAL AND PREDICTED SCORES

plt.scatter(x=X_test, y=Y_test, color='purple')
plt.plot(X_test, Y_pred, color='black')
plt.title('Actual vs Predicted', size=20)
plt.xlabel("No.of study hours",size=16)
plt.ylabel("Percentage of marks",size=16)
plt.show()
```



```
In [14]: # EVALUATION OF METRICS

print("Mean absolute error =",mean_absolute_error(Y_test, Y_pred))
print("Mean squared error =",mean_squared_error(Y_test, Y_pred))
```

Mean absolute error = 4.130879918502486
Mean squared error = 20.33292367497997

What will be the predicted score of a student if he/she studies for 9.25 hrs/ day?

```
In [15]: hours = [9.25]
answer = linreg.predict([hours])
print("Score = {}".format(round(answer[0],3)))
```

Score = 93.893

According to the regression model if a student studies for 9.25 hours a day he/she is likely to score 93.89 marks.

THANK YOU :)

In []: