

```
In [1]: #Used "Student score predictor based on class hours" data set from Kaggle  
#https://www.kaggle.com/datasets/shubham47/students-score-dataset-linear-regre  
  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
from sklearn.linear_model import LinearRegression  
from sklearn.metrics import mean_squared_error  
from sklearn.model_selection import train_test_split
```

```
In [2]: student_data=pd.read_csv('student_scores.csv')
```

```
In [3]: student_data.tail(5)
```

```
Out[3]:
```

	Hours	Scores
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

```
In [4]: student_data.describe()
```

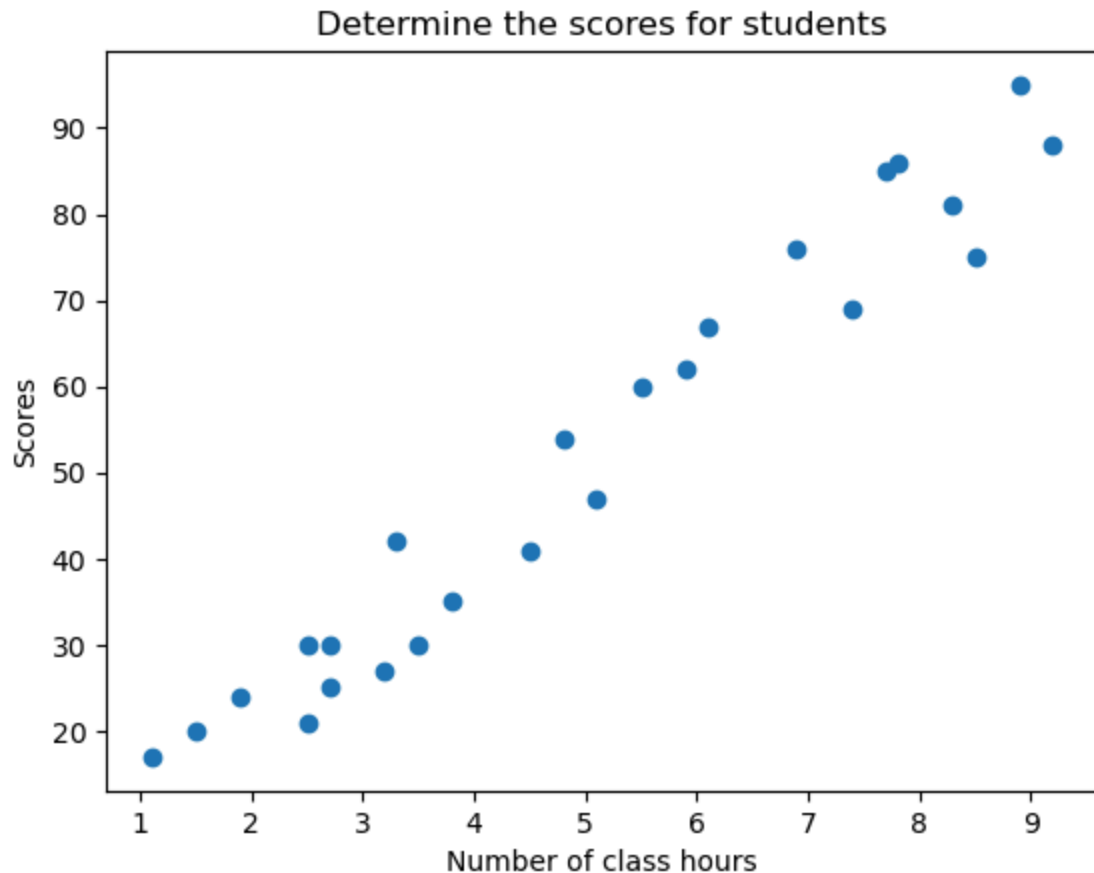
```
Out[4]:
```

	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 [5]: indep_vars=student_data['Hours'].values.reshape(-1,1)  
dep_var=student_data['Scores'].values
```

```
In [6]: plt.scatter(indp_vars,dep_var)
plt.xlabel("Number of class hours")
plt.ylabel("Scores")
plt.title("Determine the scores for students")
```

Out[6]: Text(0.5, 1.0, 'Determine the scores for students')



```
In [7]: train_x,test_x,train_y,test_y=train_test_split(indp_vars,dep_var,test_size=0.3)
```

```
In [8]: lr_model=LinearRegression()
```

```
In [9]: lr_model.fit(train_x,train_y)
```

Out[9]: LinearRegression()

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.

On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [10]: pred=lr_model.predict(test_x)
pred
```

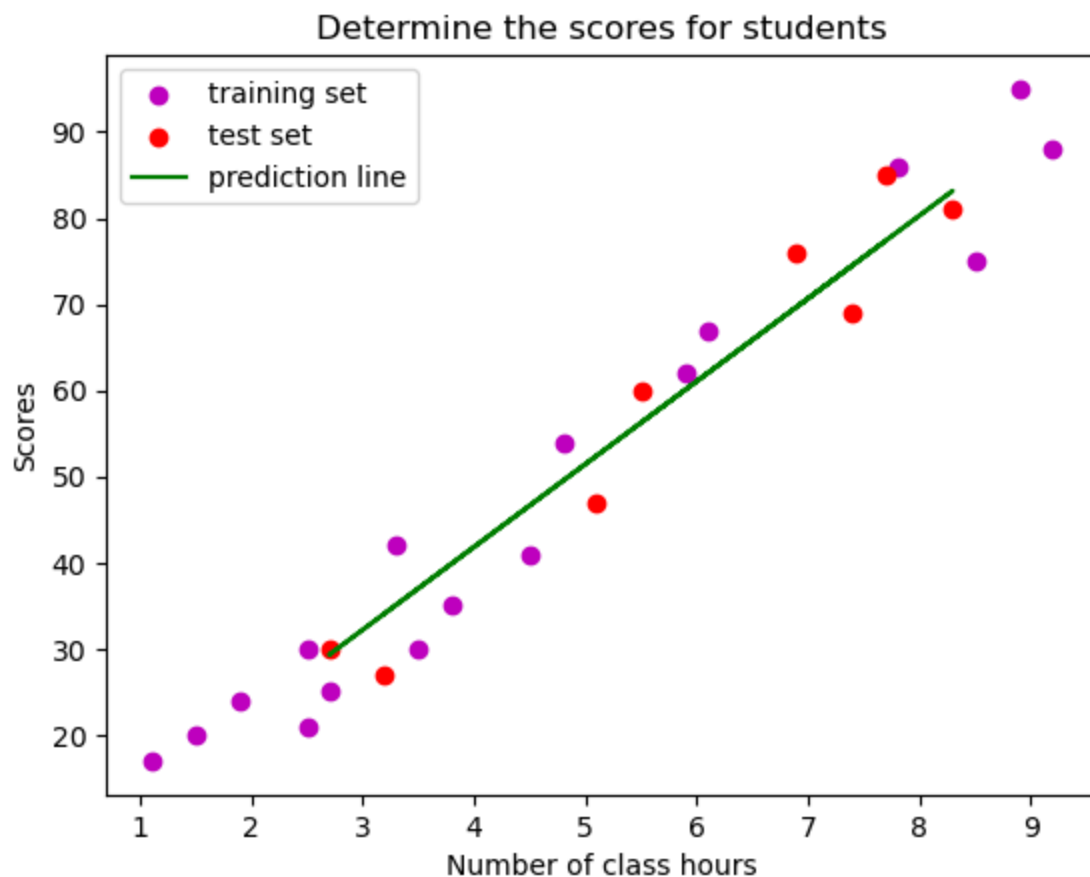
Out[10]: array([56.24769763, 77.39862373, 69.70737788, 83.16705812, 29.32833714,
52.40207471, 74.51440654, 34.1353658])

```
In [11]: mse=mean_squared_error(pred,test_y)
mse
```

Out[11]: 28.388699379556982

```
In [12]: plt.scatter(train_x,train_y,color='m',label='training set')
plt.scatter(test_x,test_y,color='red',label='test set')
plt.plot(test_x,pred,color="green",label="prediction line")
plt.legend()
plt.xlabel("Number of class hours")
plt.ylabel("Scores")
plt.title("Determine the scores for students")
```

Out[12]: Text(0.5, 1.0, 'Determine the scores for students')



```
In [15]: new_entry=input("please enter the class hour :")
new_entry=[[float(new_entry)]]
pred_new=lr_model.predict(new_entry)
print(pred_new)
```

please enter the class hour :3.8
[39.90380019]

In []: