Linear Regression with Python Scikit Learn:

Simple Linear Regression:

GRIP Task 1: Supervised Learning

+ Code + Text

Author:Chennuboina Nagababu

```
#importing  important libaries in python to perform task
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

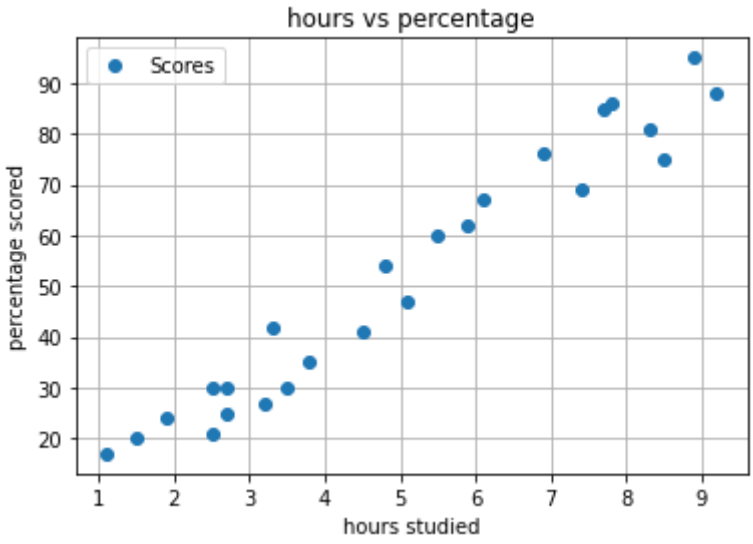
```
#reading data from given url link in task
data="http://bit.ly/w-data"
df=pd.read_csv(data)
print("data has been succesfully imported\n")
df
```

```
data has been succesfully imported
```

	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

```
#distribution of scores
x='Hours'
y='Scores'

df.plot(x,y,style='o')
plt.title('hours vs percentage')
plt.xlabel('hours studied')
plt.ylabel('percentage scored')
plt.grid()
plt.show
```

```
<function matplotlib.pyplot.show>
```



from the above graph,we can clearly see that there is a positive linear relationship between the number of hours studied and percentage ,they are directly proportional to each other

preparing the data

```
#next step is to divide the data into "attributes" (inputs) and "labels" (outputs).
X = df.iloc[:, :-1].values
y = df.iloc[:, 1].values
```

```
#divide into training and test sets by using attributes and labels.We'll do this by usi
```

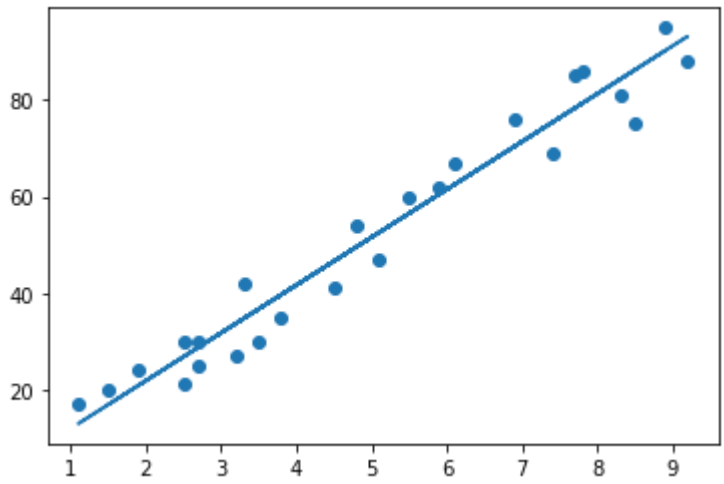
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0
```

training the algorithm

```
from sklearn.linear_model import LinearRegression

regressor = LinearRegression()
regressor.fit(X_train, y_train)

print("Model is trained successfully")
```

```
Model is trained successfully
```

```
#plotting the regreession line
# Plotting the regression line
line = regressor.coef_ * X + regressor.intercept_

# Plotting for the test data
plt.scatter(X, y)
plt.plot(X, line)
plt.show()
```



making predictions

```
#to make predictions
print(X_test)
y_pred = regressor.predict(X_test)
```

```
[[1.5]
 [3.2]
 [7.4]
 [2.5]
 [5.9]]
```

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

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

```
#What will be predicted score if a student studies for 9.25 hrs/ day?
#Predicting with dataset
hours = 9.25
own_pred = regressor.predict([[hours]])
print("No of Hours = {}".format(hours))
print("Predicted Score = {}".format(own_pred[0]))
```

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

Final step

evaluating the model

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

```
Mean Absolute Error: 4.183859899002982
```