jupyter Linear Regresssion assignment Last Checkpoint: an hour ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

In [ ]:
```
K.Mohith Saran
Simple Linear regression
5-7-2021
```

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

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

In [3]:
```
dataset
```

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
10	7.7	85
11	5.9	62
12	4.5	41
13	3.3	42
14	1.1	17
15	8.9	95
16	2.5	30
17	1.9	24
18	6.1	67
19	7.4	69
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

In [4]:
```
dataset.isnull().any() #no null values
```

Out[4]:
```
Hours     False
Scores    False
dtype: bool
```

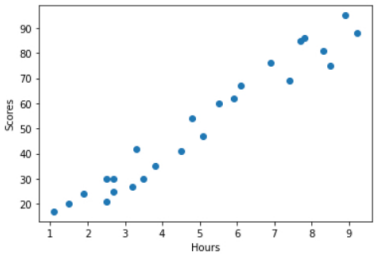
In [5]:
```
dataset.describe()
```

Out[5]:

	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 [6]:
```
plt.scatter(dataset['Hours'],dataset['Scores'])
plt.xlabel('Hours')
plt.ylabel('Scores')
```

Out[6]:
```
Text(0, 0.5, 'Scores')
```



In [7]:
```
x=dataset.iloc[:,0:1]
```

In [8]:
```
y=dataset.iloc[:,1:2]
```

In [9]:
```
x.head()
```

Out[9]:

	Hours
0	2.5
1	5.1

2	3.2
3	8.5
4	3.5

In [10]:
```
y.head()
```

Out[10]:

	Scores
0	21
1	47
2	27
3	75
4	30

In [12]:
```
x=np.array(x)
```

In [13]:
```
x
```

Out[13]:
```
array([[2.5],
       [5.1],
       [3.2],
       [8.5],
       [3.5],
       [1.5],
       [9.2],
       [5.5],
       [8.3],
       [2.7],
       [7.7],
       [5.9],
       [4.5],
       [3.3],
       [1.1],
       [8.9],
       [2.5],
       [1.9],
       [6.1],
       [7.4],
       [2.7],
       [4.8],
       [3.8],
       [6.9],
       [7.8]])
```

In [15]:
```
x.shape
```

Out[15]: (25, 1)

In [16]:
```
y.shape
```

Out[16]: (25, 1)

In [17]:
```
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)
```

In [18]:
```
x_train.shape
```

Out[18]: (20, 1)

In [19]:
```
y_train.shape
```

Out[19]: (20, 1)

In [20]:
```
x_test.shape
```

Out[20]: (5, 1)

In [21]:
```
y_test.shape
```

Out[21]: (5, 1)

In [22]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
```

In [23]:
```
lr.fit(x_train,y_train)
```

Out[23]: LinearRegression()

In [24]:
```
y_pred=lr.predict(x_test)
```

In [25]:
```
y_pred
```

Out[25]:
```
array([[16.88414476],
       [33.73226078],
       [75.357018  ],
       [26.79480124],
       [60.49103328]])
```

In [26]:
```
y_test
```

Out[26]:

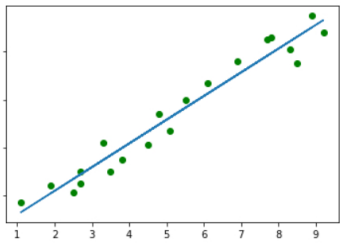
	Scores
5	20
2	27
19	69
16	30
11	62

In [27]:
```
from sklearn.metrics import r2_score
accuracy=r2_score(y_test,y_pred)
accuracy
```

Out[27]: 0.9454906892105356

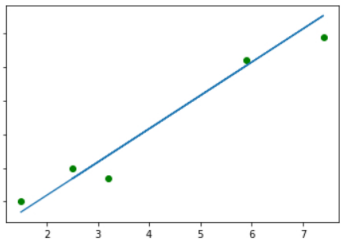
In [28]:
```
#best fit line for train data
plt.scatter(x_train,y_train,color='green')
plt.plot(x_train,lr.predict(x_train))
```

Out[28]: [<matplotlib.lines.Line2D at 0x1f14f51c550>]



In [29]:
```
plt.scatter(x_test,y_test,color='green')
plt.plot(x_test,lr.predict(x_test))
```

Out[29]: [<matplotlib.lines.Line2D at 0x1f14fbc1a00>]



In [30]:
```
lr.intercept_
```

Out[30]: array([2.01816004])

In [31]:
```
lr.coef_
```

Out[31]: array([[9.91065648]])

In [ ]: