```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("Ecommerce Customers.txt")
df.head()

                           Email  \
0      mstephenson@fernandez.com
1              hduke@hotmail.com
2               pallen@yahoo.com
3        riverarebecca@gmail.com
4  mstephens@davidson-herman.com

                                             Address            Avatar
\
0       835 Frank Tunnel\nWrightmouth, MI 82180-9605            Violet

1     4547 Archer Common\nDiazchester, CA 06566-8576         DarkGreen

2  24645 Valerie Unions Suite 582\nCobbborough, D...            Bisque

3   1414 David Throughway\nPort Jason, OH 22070-1220       SaddleBrown

4  14023 Rodriguez Passage\nPort Jacobville, PR 3...  MediumAquaMarine


   Avg. Session Length  Time on App  Time on Website  Length of
Membership  \
0            34.497268    12.655651        39.577668
4.082621
1            31.926272    11.109461        37.268959
2.664034
2            33.000915    11.330278        37.110597
4.104543
3            34.305557    13.717514        36.721283
3.120179
4            33.330673    12.795189        37.536653
4.446308

   Yearly Amount Spent
0           587.951054
1           392.204933
2           487.547505
3           581.852344
4           599.406092

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Email                 500 non-null    object 
 1   Address               500 non-null    object 
 2   Avatar                500 non-null    object 
 3   Avg. Session Length   500 non-null    float64
 4   Time on App           500 non-null    float64
 5   Time on Website       500 non-null    float64
 6   Length of Membership  500 non-null    float64
 7   Yearly Amount Spent   500 non-null    float64
dtypes: float64(5), object(3)
memory usage: 31.4+ KB
```

```
df.duplicated().sum()
```

```
0
```

```
df.isnull().sum()
```

```
Email                   0
Address                 0
Avatar                  0
Avg. Session Length     0
Time on App             0
Time on Website         0
Length of Membership    0
Yearly Amount Spent     0
dtype: int64
```

```
df.shape
```

```
(500, 8)
```

```
df1 = df.drop(['Email','Address','Avatar'],axis = 'columns')
df1.head()
```

```
   Avg. Session Length  Time on App  Time on Website  Length of 
Membership  \
0            34.497268    12.655651        39.577668      
4.082621   
1            31.926272    11.109461        37.268959      
2.664034   
2            33.000915    11.330278        37.110597      
4.104543   
3            34.305557    13.717514        36.721283      
3.120179   
4            33.330673    12.795189        37.536653      
4.446308
```

```
   Yearly Amount Spent
0           587.951054
1           392.204933
2           487.547505
3           581.852344
4           599.406092
```

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Avg. Session Length   500 non-null    float64
 1   Time on App           500 non-null    float64
 2   Time on Website       500 non-null    float64
 3   Length of Membership  500 non-null    float64
 4   Yearly Amount Spent   500 non-null    float64
dtypes: float64(5)
memory usage: 19.7 KB
```

```
x = df1.drop(['Yearly Amount Spent'],axis=1)
y = df1[['Yearly Amount Spent']]
```

```
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=101)
```

```
x_train
```

```
     Avg. Session Length  Time on App  Time on Website  Length of
Membership
202            31.525752    11.340036        37.039514
3.811248
428            31.862741    14.039867        37.022269
3.738225
392            33.258238    11.514949        37.128039
4.662845
86             33.877779    12.517666        37.151921
2.669942
443            33.025020    12.504220        37.645839
4.051382
..                   ...          ...              ...
...
63             32.789773    11.670066        37.408748
3.414688
326            33.217188    10.999684        38.442767
4.243813
337            31.827979    12.461147        37.428997
```

```
2.974737
11             33.879361    11.584783        37.087926
3.713209
351            32.189845    11.386776        38.197483
4.808320

[350 rows x 4 columns]

x_test

     Avg. Session Length  Time on App  Time on Website  Length of
Membership
18             32.187812    14.715388        38.244115
1.516576
361            32.077590    10.347877        39.045156
3.434560
104            31.389585    10.994224        38.074452
3.428860
4              33.330673    12.795189        37.536653
4.446308
156            32.294642    12.443048        37.327848
5.084861
..                   ...          ...              ...
...
147            32.255901    10.480507        37.338670
4.514122
346            32.765665    12.506548        35.823467
3.126509
423            33.128693    10.398458        36.683393
3.859818
17             32.338899    12.013195        38.385137
2.420806
259            32.096109    10.804891        37.372762
2.699562

[150 rows x 4 columns]

y_train

     Yearly Amount Spent
202           443.965627
428           556.298141
392           549.131573
86            487.379306
443           561.516532
..                   ...
63            483.159721
326           505.230068
337           440.002748
11            522.337405
```

```
351              533.396554

[350 rows x 1 columns]
```

```python
y_test
```

```
     Yearly Amount Spent
18            452.315675
361           401.033135
104           410.069611
4             599.406092
156           586.155870
..                   ...
147           479.731938
346           488.387526
423           461.112248
17            407.704548
259           375.398455

[150 rows x 1 columns]
```

```python
from sklearn.linear_model import LinearRegression
```

```python
lr = LinearRegression()
```

```python
lr.fit(x_train,y_train)
```

```
LinearRegression()
```

```python
lr.score(x_train,y_train)
```

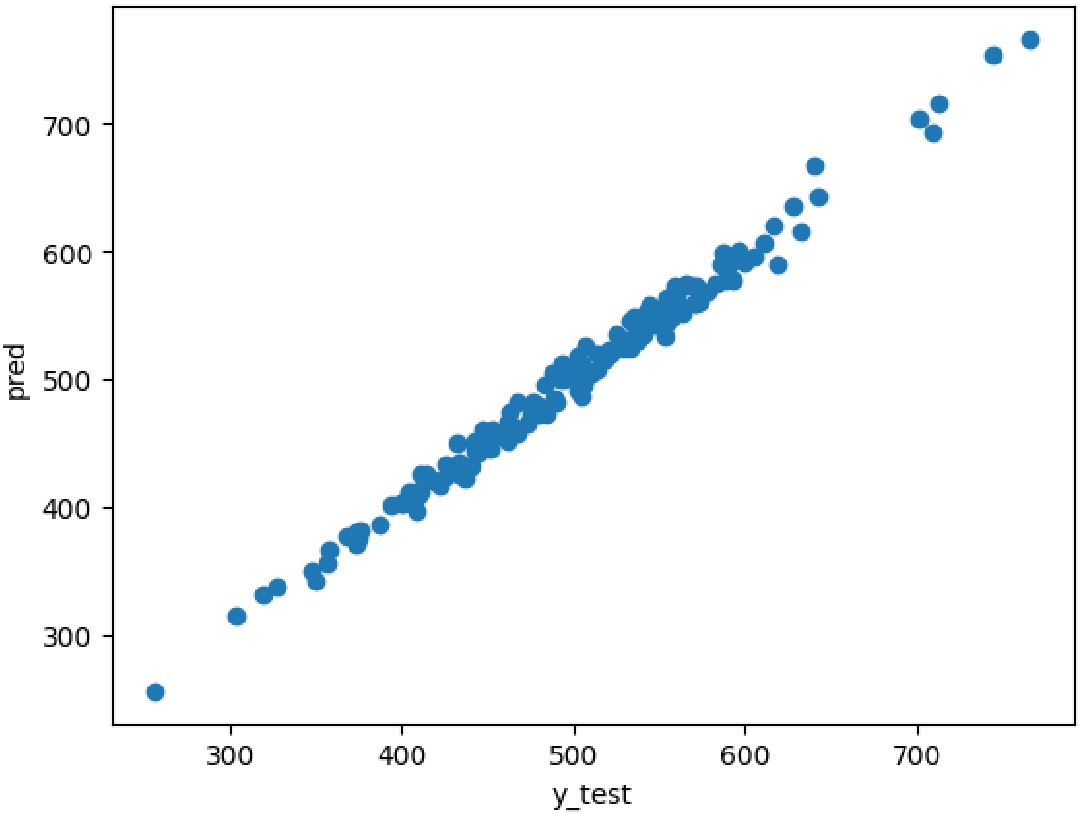
```
0.9817562058732432
```

```python
lr.score(x_test,y_test)
```

```
0.9890046246741234
```

```python
pred = lr.predict(x_test)
```

```python
plt.scatter(y_test,pred)
plt.xlabel('y_test')
plt.ylabel('pred')
plt.show()
```


700
600
500
400
300
pred
300
400
500
600
700
y_test