E-commerce list

November 6, 2023

```
[8]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as mp
     import seaborn as sb
```

1 Data Importing & Exporting

```
[9]: df=pd.read_csv('Ecommerce Customers.csv')
     df
```

```
[9]:                              Email  \
     0        mstephenson@fernandez.com
     1                hduke@hotmail.com
     2                 pallen@yahoo.com
     3          riverarebecca@gmail.com
     4    mstephens@davidson-herman.com
     ..                             …
     495   lewisjessica@craig-evans.com
     496            katrina56@gmail.com
     497             dale88@hotmail.com
     498            cwilson@hotmail.com
     499      hannahwilson@davidson.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
     ..                                                 …                 …
     495  4483 Jones Motorway Suite 872\nLake Jamiefurt,…               Tan
     496  172 Owen Divide Suite 497\nWest Richard, CA 19320     PaleVioletRed
     497  0787 Andrews Ranch Apt. 633\nSouth Chadburgh, …          Cornsilk
     498  680 Jennifer Lodge Apt. 808\nBrendachester, TX…              Teal
     499  49791 Rachel Heights Apt. 898\nEast Drewboroug…       DarkMagenta

          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
     ..                   …          …              …                   …
     495            33.237660    13.566160        36.417985              3.746573
     496            34.702529    11.695736        37.190268              3.576526
     497            32.646777    11.499409        38.332576              4.958264
     498            33.322501    12.391423        36.840086              2.336485
     499            33.715981    12.418808        35.771016              2.735160

          Yearly Amount Spent
     0             587.951054
     1             392.204933
     2             487.547505
     3             581.852344
     4             599.406092
     ..                   …
     495           573.847438
     496           529.049004
     497           551.620145
     498           456.469510
     499           497.778642

     [500 rows x 8 columns]
```

[10]:
```
df.head()
```

[10]:
```
                           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
```

[11]:

```
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
```

[12]:

```
df.describe().round(2)
```

[12]:

```
       Avg. Session Length  Time on App  Time on Website  \
count               500.00       500.00           500.00   
mean                 33.05        12.05            37.06   
std                   0.99         0.99             1.01   
min                  29.53         8.51            33.91   
25%                  32.34        11.39            36.35   
50%                  33.08        11.98            37.07   
75%                  33.71        12.75            37.72   
max                  36.14        15.13            40.01   

       Length of Membership  Yearly Amount Spent  
count                500.00               500.00  
mean                   3.53               499.31  
std                    1.00                79.31  
min                    0.27               256.67  
25%                    2.93               445.04  
50%                    3.53               498.89  
75%                    4.13               549.31  
```

```
max                      6.92               765.52
```

```
[29]: #df.sb.heatmap(df.corr(),annot=true)
```

```
[ ]: #sb.jointplot(data=df , x =['Length of Membership'] , y =['Yearly Amount␣
      ↪Spent'] )
```

```
[1]: #sb.jointplot(data=df, x = 'Length of Membership' , y = 'Yearly Amount Spent')
```

```
[20]: y=df['Yearly Amount Spent']
      y
```

```
[20]: 0      587.951054
      1      392.204933
      2      487.547505
      3      581.852344
      4      599.406092
                ...
      495    573.847438
      496    529.049004
      497    551.620145
      498    456.469510
      499    497.778642
      Name: Yearly Amount Spent, Length: 500, dtype: float64
```

```
[21]: x=df[['Avg. Session Length','Time on App','Time on Website','Length of␣
       ↪Membership']]
      x
```

```
[21]:      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
      ..                   …            …              …                    …
      495            33.237660    13.566160        36.417985              3.746573
      496            34.702529    11.695736        37.190268              3.576526
      497            32.646777    11.499409        38.332576              4.958264
      498            33.322501    12.391423        36.840086              2.336485
      499            33.715981    12.418808        35.771016              2.735160

      [500 rows x 4 columns]
```

```
[22]: 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=42)
```

[23]:
```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train , y_train)
```

[23]:
```
LinearRegression()
```

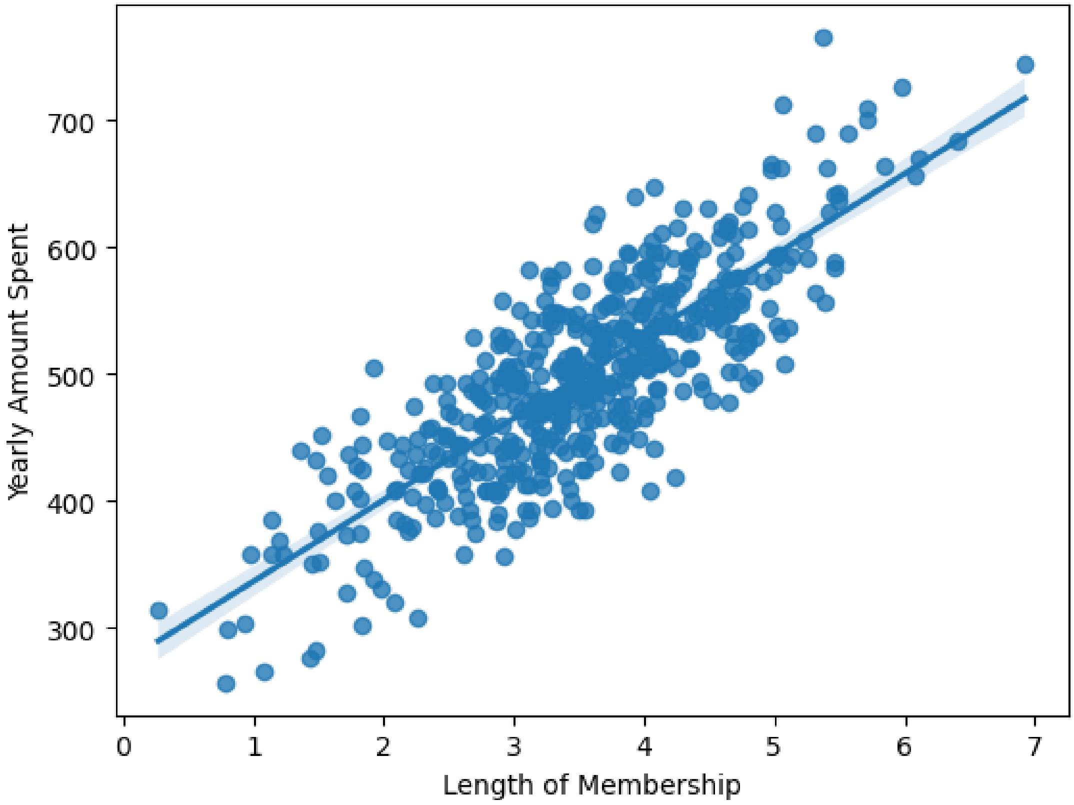
[24]:
```
model.coef_
```

[24]:
```
array([25.5962591 , 38.78534598,  0.31038593, 61.89682859])
```

[28]:
```
#sb.heatmap(df.corr(),annot=True,cmap='Blues')
```

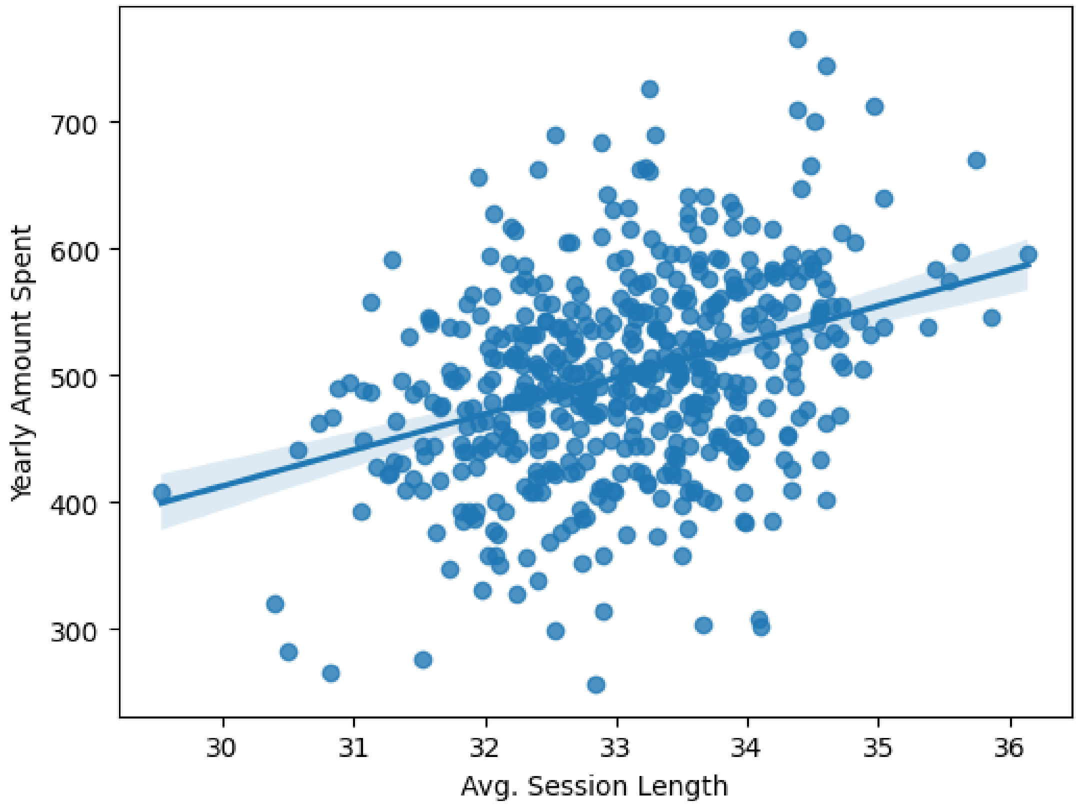
[27]:
```
sb.regplot(data=df, x = 'Length of Membership' , y = 'Yearly Amount Spent')
```

[27]:
```
<Axes: xlabel='Length of Membership', ylabel='Yearly Amount Spent'>
```



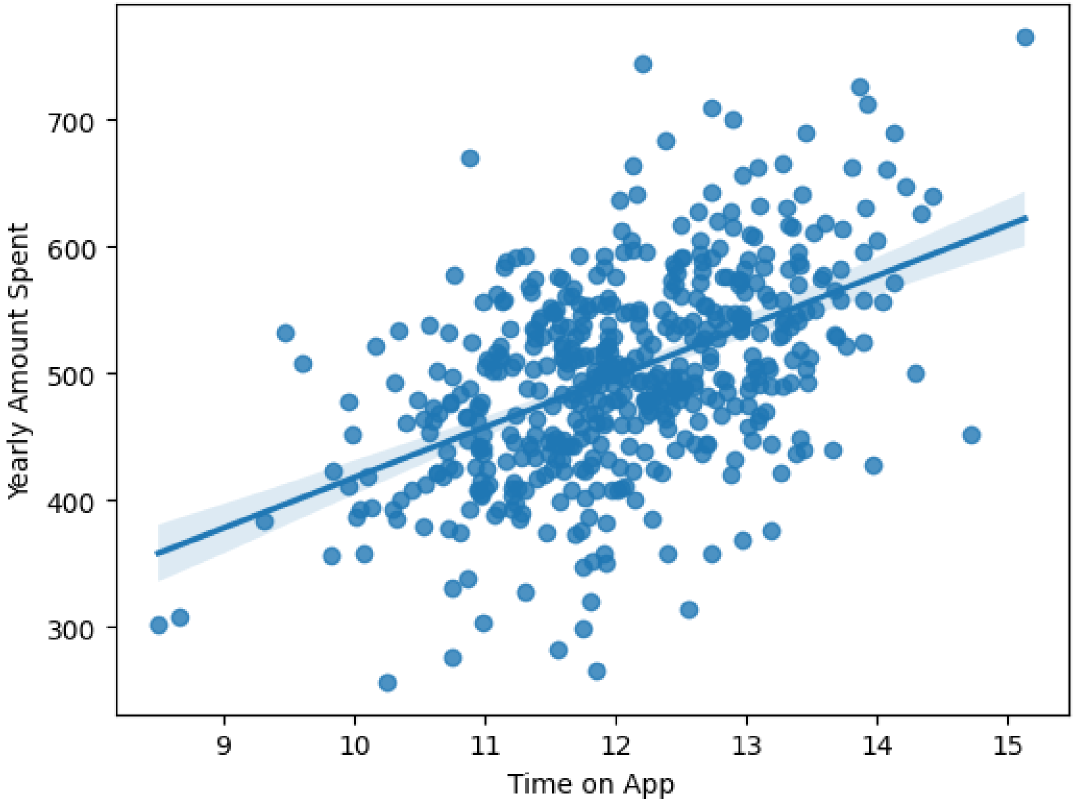
[42]:
```
sb.regplot(data=df, x = 'Avg. Session Length' , y = 'Yearly Amount Spent')
```

[42]:
```
<Axes: xlabel='Avg. Session Length', ylabel='Yearly Amount Spent'>
```



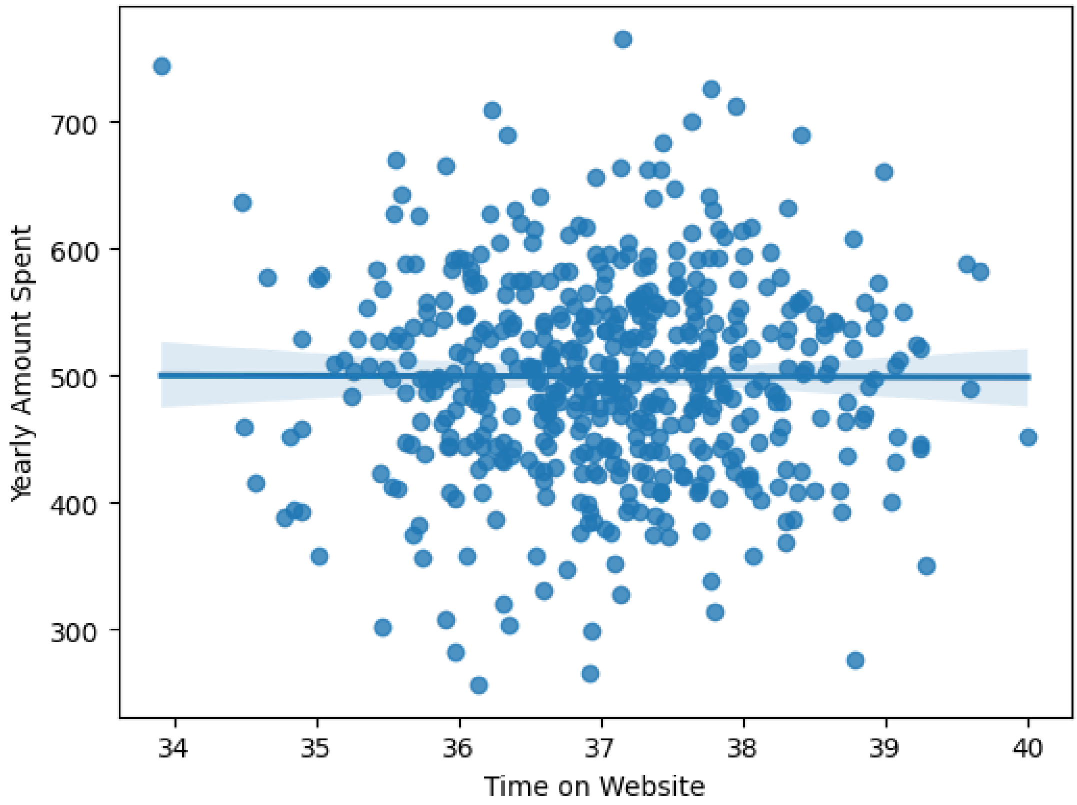
```
[43]: sb.regplot(data=df, x = 'Time on App' , y = 'Yearly Amount Spent')
```

```
[43]: <Axes: xlabel='Time on App', ylabel='Yearly Amount Spent'>
```



```
[44]: sb.regplot(data=df, x = 'Time on Website' , y = 'Yearly Amount Spent')
```

```
[44]: <Axes: xlabel='Time on Website', ylabel='Yearly Amount Spent'>
```



```
[55]: Y_pred=model.predict(X_test)
      Y_pred
```

```
[55]: array([402.86230051, 542.53325708, 426.62011918, 501.91386363,
             409.6666551 , 569.92155038, 531.50423529, 505.94309188,
             408.10378607, 473.45942928, 441.18668812, 424.52463471,
             424.83341694, 527.12061508, 430.87985533, 423.47062047,
             575.8751518 , 484.6563331 , 457.77896975, 481.58742311,
             501.56110993, 513.12815188, 507.49166899, 646.63377343,
             449.70050586, 496.26290484, 556.18523776, 554.78684161,
             399.1582784 , 325.16921284, 532.62732659, 477.73025415,
             500.76491535, 305.09971374, 505.46811902, 483.52069444,
             519.09464122, 437.75549737, 456.25005245, 470.63517876,
             494.11207805, 444.65549239, 508.57079732, 500.88197484,
             488.35128728, 535.34025218, 594.58301773, 513.59474408,
             279.69877702, 432.71590835, 421.06976164, 480.94327496,
             584.59481888, 608.61734059, 564.42312991, 494.47224504,
             393.95593318, 456.11321352, 572.92228417, 499.27385693,
             512.42973545, 391.56170305, 479.60705887, 481.05023229,
             474.71926117, 546.37716047, 430.11675694, 601.91418143,
             422.26508516, 493.11622454, 528.10614863, 581.06630842,
             620.60774498, 512.47838603, 411.2147464 , 498.07095351,
             461.44587681, 445.63453258, 447.63898998, 534.81030495,
             598.85091016, 619.46554961, 494.43362232, 672.2442837 ,
             532.15516513, 438.41740681, 514.80907179, 546.73893548,
             331.73069072, 510.33949236, 536.21660556, 499.50696031,
             375.86919792, 573.61952185, 479.18212334, 588.32862943,
             485.18137257, 455.93070091, 398.67820721, 451.70869105])
```

[59]:

```
#Y _test.values()
```

```
  Cell In[59], line 1
    Y _test.values()
      ^
SyntaxError: invalid syntax
```

[53]:

```
model.score(x,y)
```

[53]:

```
0.9842900924920522
```

[61]:

```
cof = pd.DataFrame(model.coef_,x.columns)
cof.columns = ['Coeffecient']
cof
```

[61]:

```
                      Coeffecient
Avg. Session Length     25.596259
Time on App             38.785346
Time on Website          0.310386
Length of Membership    61.896829
```

[67]:

```
from sklearn.metrics import mean_absolute_error,mean_squared_error
print(mean_absolute_error(y_test , y_pred))
```

```
8.558441885315235
```

[72]:

```
 print(mean_squared_error(y_test , y_pred))
```

```
109.86374118394022
```

[ ]:

```

```