

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

```
In [2]: customers=pd.read_csv("Ecommerce Customers")
customers.head()
```

Out[2]:

	Email	Address	Avatar	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
0	mstephenson@fernandez.com	835 Frank Tunneh\nWrightmouth, MI 82180-9605	Violet	34.497268	12.655651	39.577668	4.082621	587.951054
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.109461	37.268959	2.664034	392.204933
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.330278	37.110597	4.104543	487.547505
3	rverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13.717514	36.721283	3.120179	581.852344
4	mstephens@davids-on-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12.795189	37.536653	4.464308	599.406092

```
In [3]: customers.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

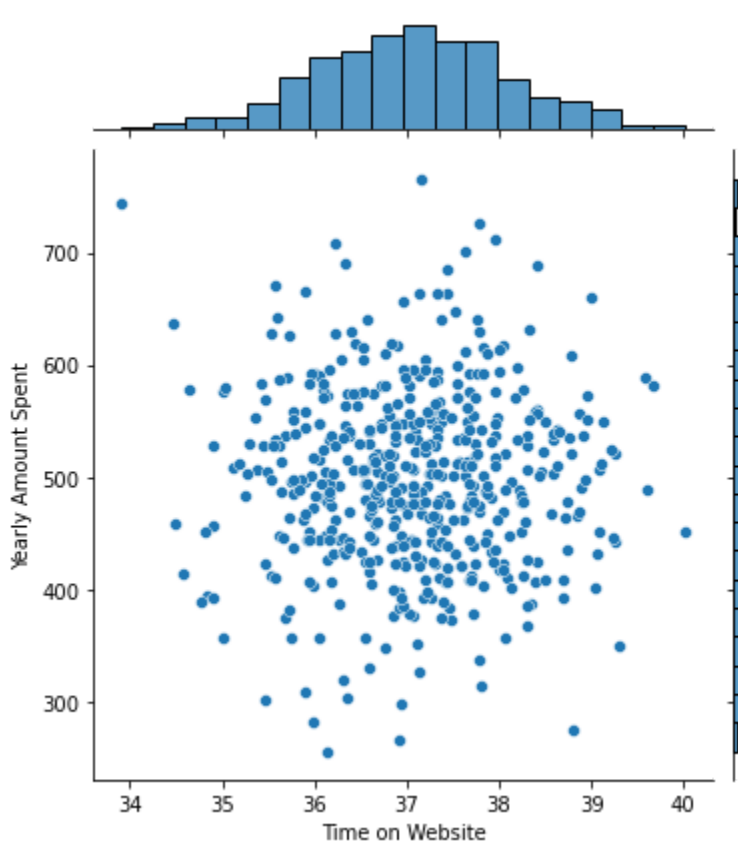
In [4]: customers.describe()
```

Out[4]:

	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
count	500.000000	500.000000	500.000000	500.000000	500.000000
mean	33.053194	12.052488	37.060445	3.533462	499.314038
std	0.992563	0.994216	1.010489	0.999278	79.314782
min	29.532429	8.508152	33.913847	0.269901	256.670582
25%	32.341822	11.388153	36.349257	2.930450	445.038277
50%	33.082008	11.983231	37.069367	3.533975	498.887875
75%	33.711985	12.753850	37.716432	4.126502	549.313828
max	36.139662	15.126994	40.005182	6.922689	765.518462

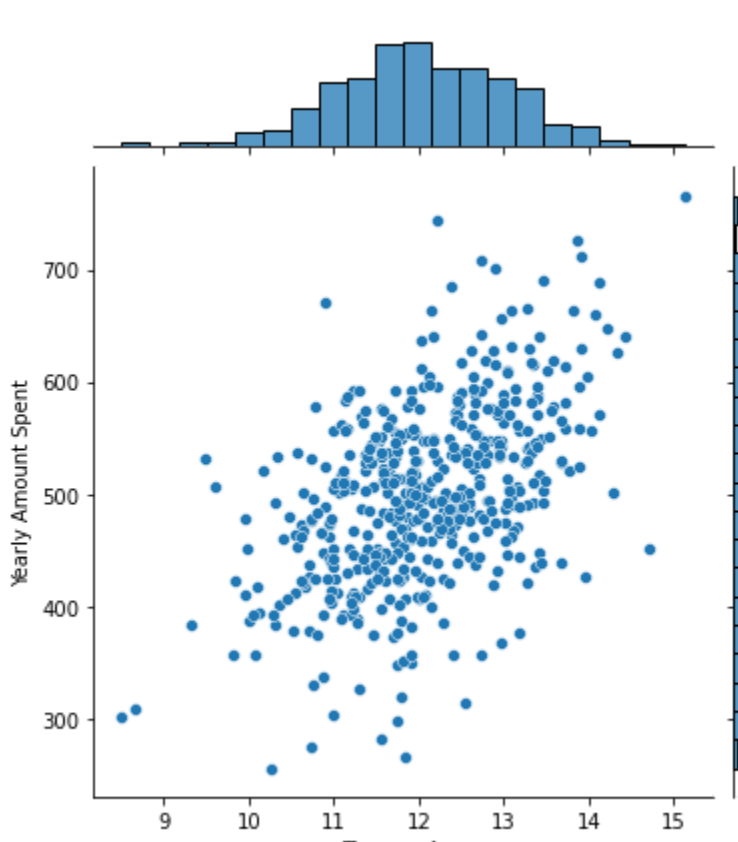
```
In [5]: import seaborn as sns
sns.jointplot(data=customers,x="Time on Website",y="Yearly Amount Spent")
```

Out[6]: <seaborn.axisgrid.JointGrid at 0x1fb540f2520>



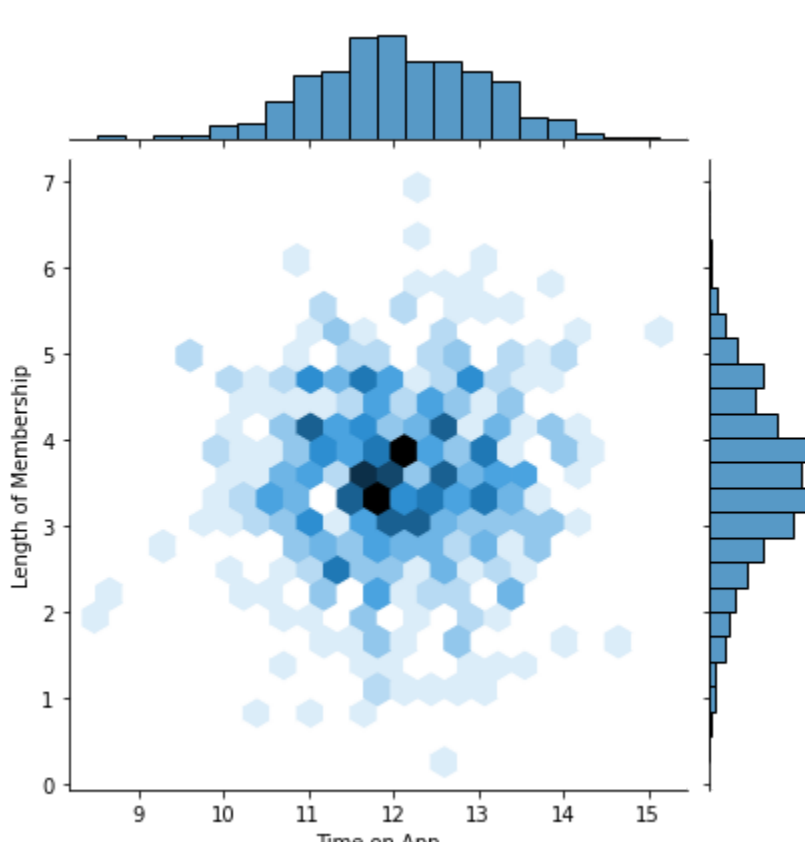
```
In [7]: sns.jointplot(x="Time on App",y="Yearly Amount Spent",data=customers)
```

Out[7]: <seaborn.axisgrid.JointGrid at 0x1fb550753d0>



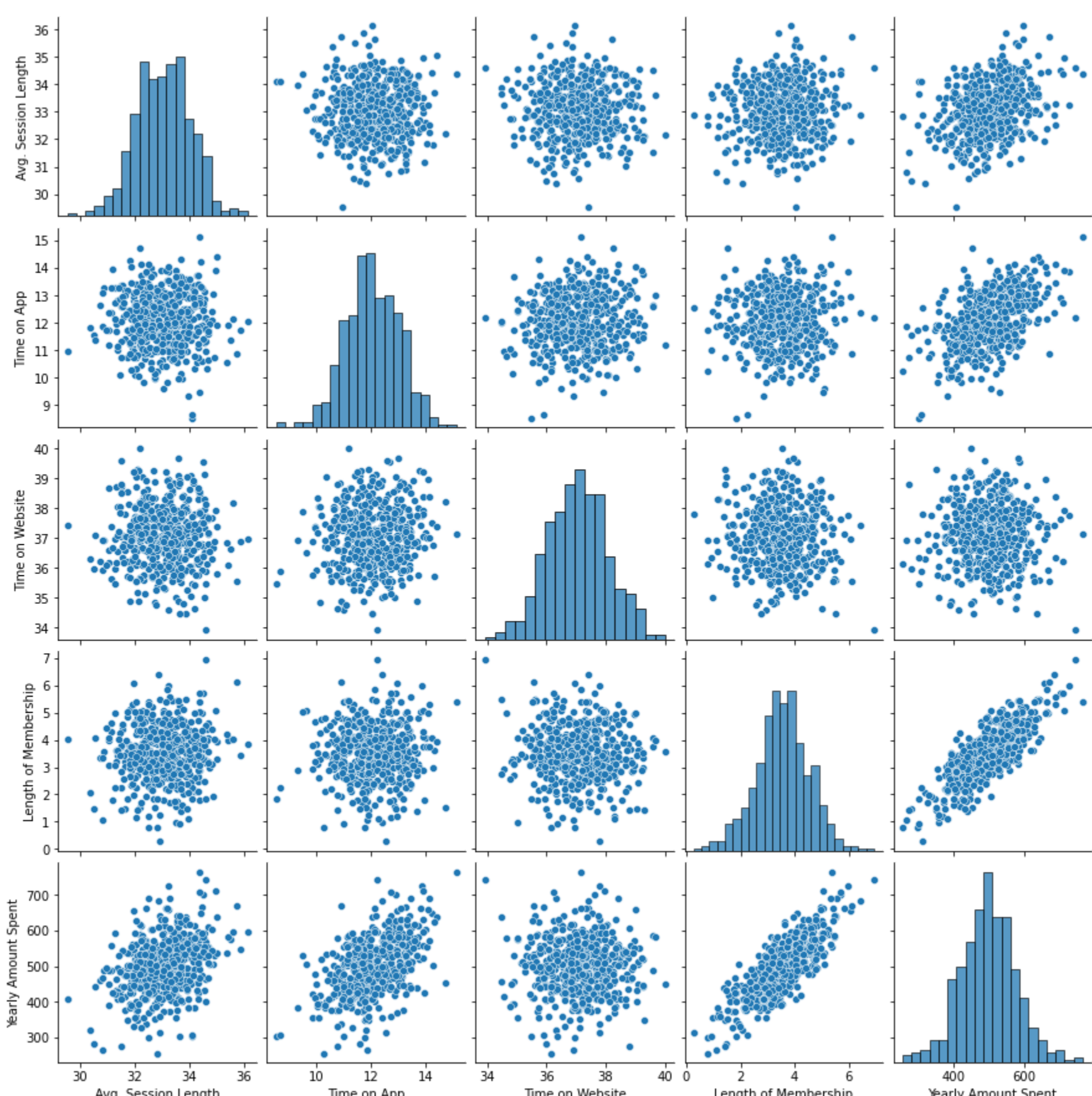
```
In [8]: sns.jointplot(x="Time on App",y="Length of Membership",data=customers,kind="hex")
```

Out[8]: <seaborn.axisgrid.JointGrid at 0x1fb55b0bbe0>



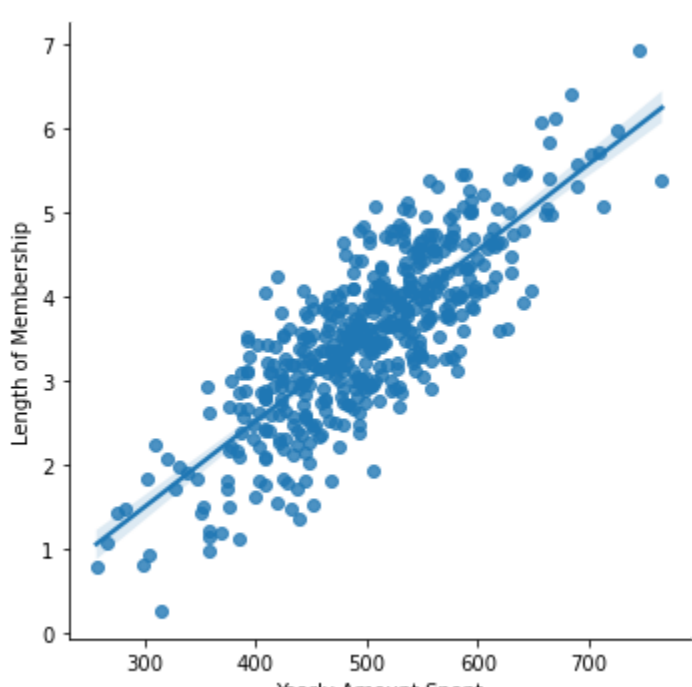
```
In [9]: sns.pairplot(data=customers)
```

Out[9]: <seaborn.axisgrid.PairGrid at 0x1fb55c42a00>



```
In [10]: sns.lmplot(x="Yearly Amount Spent",y="Length of Membership",data=customers)
```

Out[10]: <seaborn.axisgrid.FacetGrid at 0x1fb4ffed00>



```
In [11]: customers.columns
```

Out[11]: Index(['Email', 'Address', 'Avatar', 'Avg. Session Length', 'Time on App', 'Time on Website', 'Length of Membership', 'Yearly Amount Spent'], dtype='object')

```
In [12]: X= customers[['Avg. Session Length', 'Time on App','Time on Website', 'Length of Membership']]
y= customers["Yearly Amount Spent"]
```

```
In [13]: from sklearn.model_selection import train_test_split
```

```
In [14]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=101)
```

```
In [15]: from sklearn.linear_model import LinearRegression
```

```
In [16]: lm=LinearRegression()
```

```
In [17]: lm.fit(X_train,y_train)
```

Out[17]: LinearRegression()

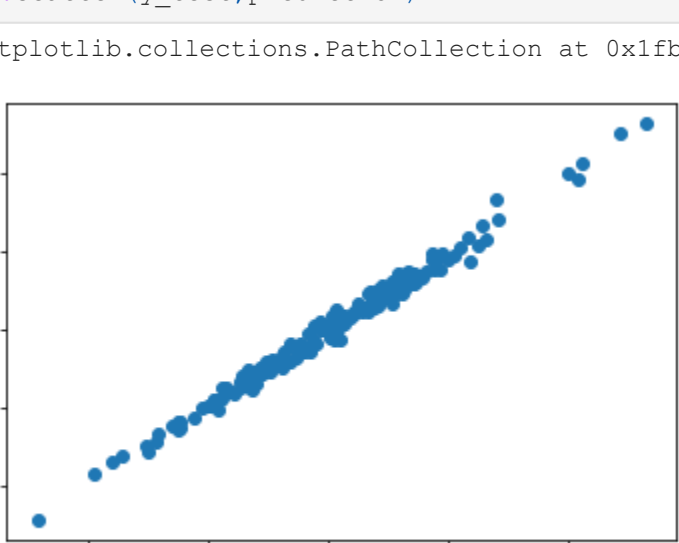
```
In [18]: print("coeff:",lm.coef_)
```

coeff: [25.91225933 38.50812563 0.28871629 61.16161032]

```
In [19]: prediction=lm.predict(X_test)
```

```
In [20]: plt.scatter(y_test,prediction)
```

Out[20]: <matplotlib.collections.PathCollection at 0x1fb583bcf70>



```
In [21]: from sklearn import metrics
```

```
In [22]: print("MAE:",metrics.mean_absolute_error(y_test,prediction))
print("MSE:",metrics.mean_squared_error(y_test,prediction))
print("RMSE:",np.sqrt(metrics.mean_squared_error(y_test,prediction)))
```

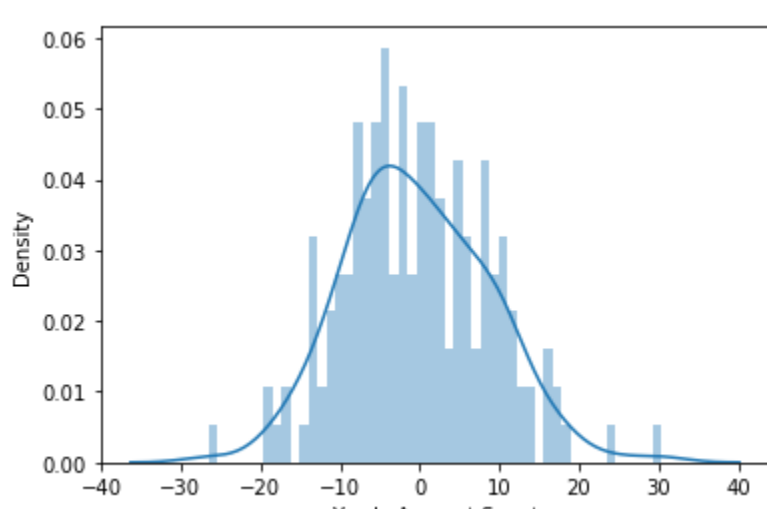
MAE: 7.29454658831306
MSE: 81.90726984520916
RMSE: 9.0502635235229

```
In [23]: sns.distplot((y_test-prediction),bins=50)
```

C:\Users\lenovo\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version. Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for histograms).

warnings.warn(msg, FutureWarning)

Out[23]: <AxesSubplot: xlabel='Yearly Amount Spent', ylabel='Density'>



```
In [24]: lm.intercept_
```

-1047.975964821279

```
In [25]: coef=mpd.DataFrame(lm.coef_,X.columns,columns=["coeff"])
coef
```

Out[25]:

	coeff
Avg. Session Length	25.912259
Time on App	38.508126
Time on Website	0.288716
Length of Membership	61.161610

