Linear Regression Project

You just got some contract work with an Ecommerce company based in New York City that sells clothing online but they also have in-store style and clothing advice sessions. Customers come in to the store, have sessions/meetings with a personal stylist, then they can go home and order either on a mobile app or website for the clothes they want.

The company is trying to decide whether to focus their efforts on their mobile app experience or their website. They've hired you on contract to help them figure it out! Let's get started!

Imports

** Import pandas, numpy, matplotlib,and seaborn. Then set %matplotlib inline (You'll import sklearn as you need it.)**

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

Get the Data

We'll work with the Ecommerce Customers csv file from the company. It has Customer info, suchas Email, Address, and their color Avatar. Then it also has numerical value columns:

- Avg. Session Length: Average session of in-store style advice sessions.
- Time on App: Average time spent on App in minutes
- Time on Website: Average time spent on Website in minutes
- Length of Membership: How many years the customer has been a member.

** Read in the Ecommerce Customers csv file as a DataFrame called customers.**

In [ ]:
```
df = pd.read_csv("Ecommerce Customers")
```

Check the head of customers, and check out its info() and describe() methods.

In [ ]:
```
df
```

Out[54]:

	Email	Address	Avatar	Avg. Session Length	Time on App
0	mstephenson@fernandez.com	835 Frank Tunnel\nWrightmouth, MI 82180-9605	Violet	34.497268	12.655651
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.109461
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.330278
3	riverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13.717514
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12.795189
...	...	...	...	...	...
495	lewisjessica@craig-evans.com	4483 Jones Motorway Suite 872\nLake Jamiefurt,...	Tan	33.237660	13.566160
496	katrina56@gmail.com	172 Owen Divide Suite 497\nWest Richard, CA 19320	PaleVioletRed	34.702529	11.695736
497	dale88@hotmail.com	0787 Andrews Ranch Apt. 633\nSouth Chadburgh, ...	Cornsilk	32.646777	11.499409
498	cwilson@hotmail.com	680 Jennifer Lodge Apt. 808\nBrendachester, TX...	Teal	33.322501	12.391423
499	hannahwilson@davidson.com	49791 Rachel Heights Apt. 898\nEast Drewboroug...	DarkMagenta	33.715981	12.418808

500 rows × 8 columns

In [ ]:
```
df.corr()
```

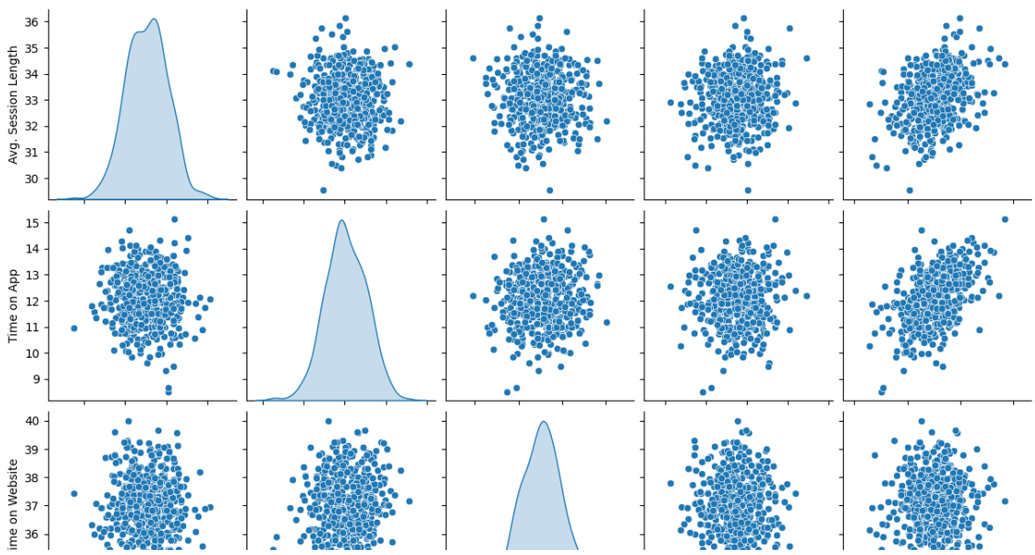
```
<ipython-input-4-2f6f6606aa2c>:1: FutureWarning: The default value of numeric
_only in DataFrame.corr is deprecated. In a future version, it will default t
o False. Select only valid columns or specify the value of numeric_only to si
lence this warning.
  df.corr()
```

Out[4]:

	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
Avg. Session Length	1.000000	-0.027826	-0.034987	0.060247	0.355088
Time on App	-0.027826	1.000000	0.082388	0.029143	0.499328
Time on Website	-0.034987	0.082388	1.000000	-0.047582	-0.002641
Length of Membership	0.060247	0.029143	-0.047582	1.000000	0.809084
Yearly Amount Spent	0.355088	0.499328	-0.002641	0.809084	1.000000

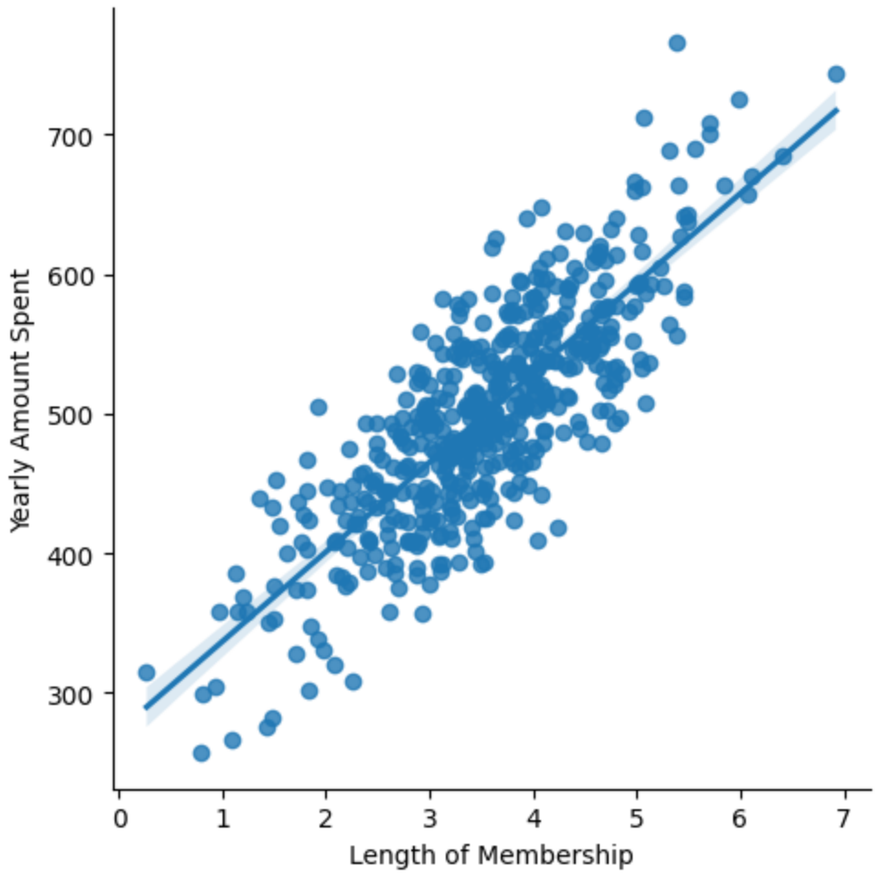
In [ ]:
```
sns.pairplot(df,diag_kind='kde')
```

Out[5]: <seaborn.axisgrid.PairGrid at 0x7aff4dd77970>



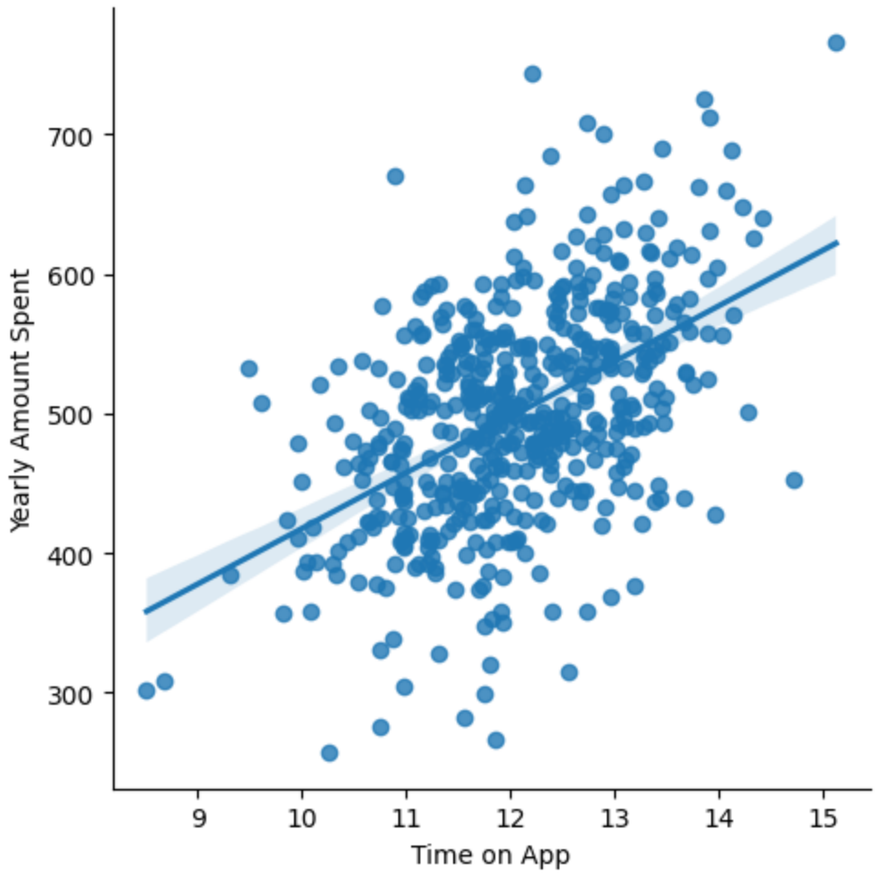
In [ ]:
```
sns.lmplot(x='Length of Membership',y='Yearly Amount Spent',data=df)
```

Out[12]: <seaborn.axisgrid.FacetGrid at 0x7aff4523e200>



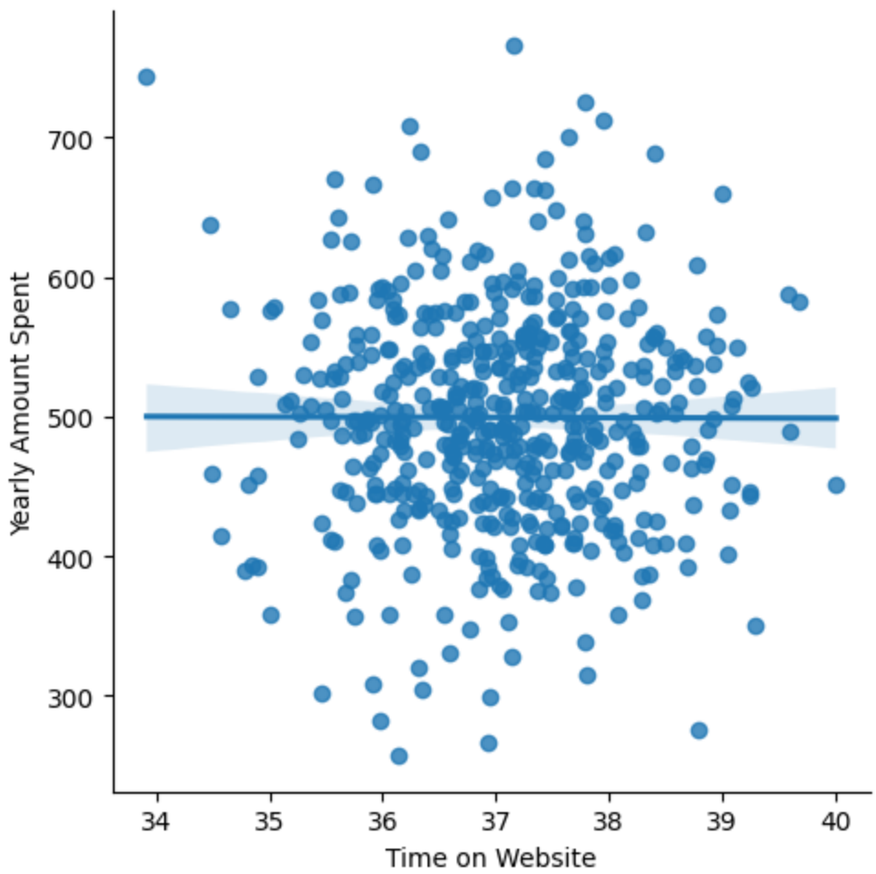
In [ ]:
```
sns.lmplot(x='Time on App',y='Yearly Amount Spent',data=df)
```

Out[13]: <seaborn.axisgrid.FacetGrid at 0x7aff4500cd00>



In [ ]:
```
sns.lmplot(x='Time on Website',y='Yearly Amount Spent',data=df)
```

Out[14]: <seaborn.axisgrid.FacetGrid at 0x7aff48adfe20>



In [ ]:
```
y = df['Yearly Amount Spent']
X = df[['Avg. Session Length', 'Time on App']]
```

In [ ]:
```
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, rando
```

In [ ]:
```
X_train.shape
```

Out[43]:
```
(400, 2)
```

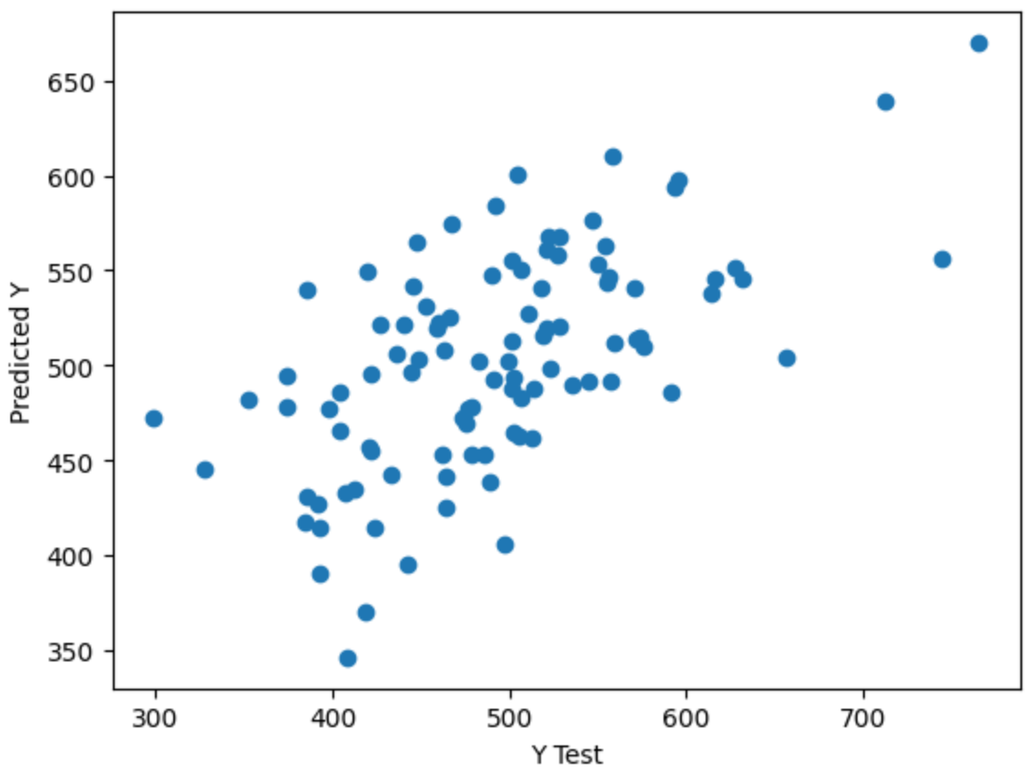
In [ ]:
```
from sklearn.linear_model import LinearRegression

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

Out[44]:
```
LinearRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [ ]:
```
y_pred = regressor.predict(X_test)
```

In [ ]:
```
plt.scatter(y_test,y_pred)
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
```

Out[47]:
```
Text(0, 0.5, 'Predicted Y')
```



In [ ]:
```
from sklearn import metrics

print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
print('MSE:', metrics.mean_squared_error(y_test, y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
MAE: 51.97826128658433
MSE: 4381.538601641748
RMSE: 66.19319150518237
```

In [ ]:
```
coeffecients = pd.DataFrame(regressor.coef_,X.columns)
coeffecients.columns = ['Coeffecient']
coeffecients
```

Out[49]:

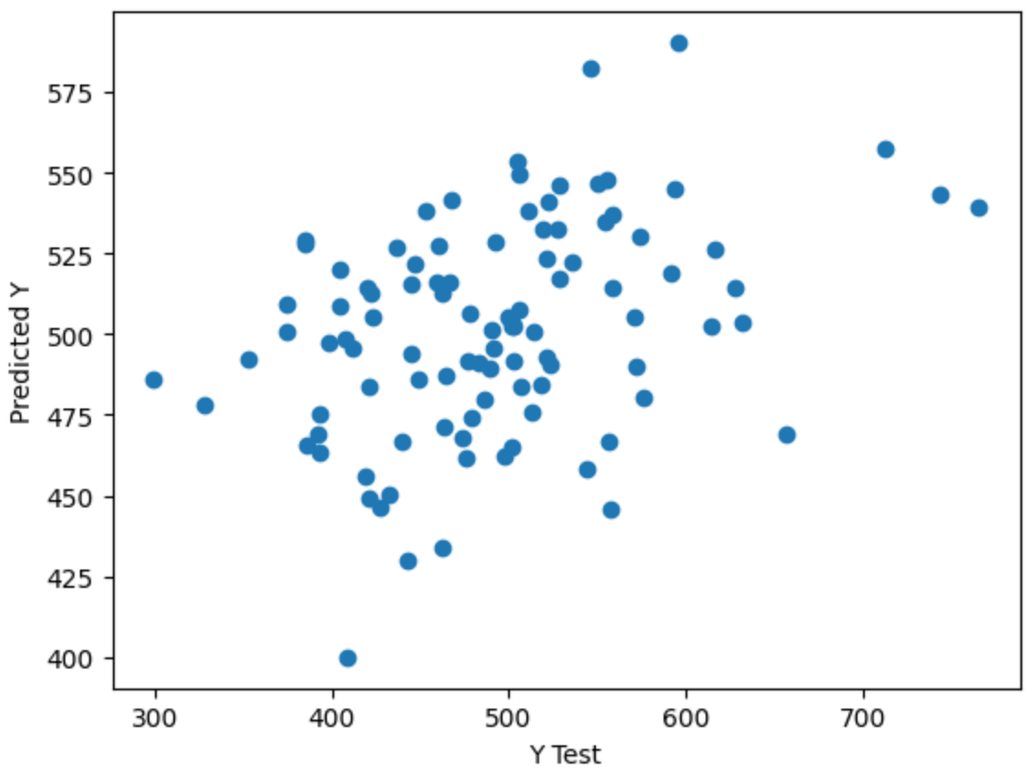
	Coeffecient
Avg. Session Length	31.305614
Time on App	41.398402

In [ ]:
```
```

In [ ]:
```
y = df['Yearly Amount Spent']
X = df[['Avg. Session Length', 'Time on Website']]

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, rando

from sklearn.linear_model import LinearRegression

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

y_pred = regressor.predict(X_test)
```

In [ ]:
```
plt.scatter(y_test,y_pred)
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
```

Out[51]:
```
Text(0, 0.5, 'Predicted Y')
```



In [ ]:
```
from sklearn import metrics

print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
print('MSE:', metrics.mean_squared_error(y_test, y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
MAE: 57.231266605573694
MSE: 5872.306873054435
RMSE: 76.63097854689339
```

In [ ]:
```
coeffecients = pd.DataFrame(regressor.coef_,X.columns)
coeffecients.columns = ['Coeffecient']
coeffecients
```

Out[53]:

	Coeffecient
Avg. Session Length	28.89477
Time on Website	0.92860

Conclusion

Comparing these scores, it appears that the model trained with time on the app performs better according to all three metrics. A lower MAE, MSE, and RMSE indicate better predictive accuracy, suggesting that the model trained with time on the app provides more accurate predictions of the yearly amount spent by customers.

It seems advisable for the company to focus its efforts on analyzing and improving the mobile app experience rather than the website, as the model trained with mobile app data demonstrates better predictive performance.

However we noticed that length of membership has more correlation to the yearly amount spent than the time on either the app or website. This shows the longer the customer has been a member, the more they spend.

In [ ]:
```

```