E- Commerce project

Problem Statement- A Company is trying to decide whether to focus their efforts on their mobile app experience or their website.

Method: Linear Regression/source of Data: Udemy/Tool: Python

Library: - Pandas, Matplotlib, sklearn, seaborn, Numpy

Dataset: head of dataset is as below and Target: 'Yearly Amount spent'

```
df.head()
```

	Email	Address	Avatar	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
0	mstephenson@fernandez.com	835 Frank Tunnel\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	riverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13.717514	36.721283	3.120179	581.852344
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12.795189	37.536653	4.446308	599.406092

Description of data snap:- it's have 500 row

```
df.describe()
```

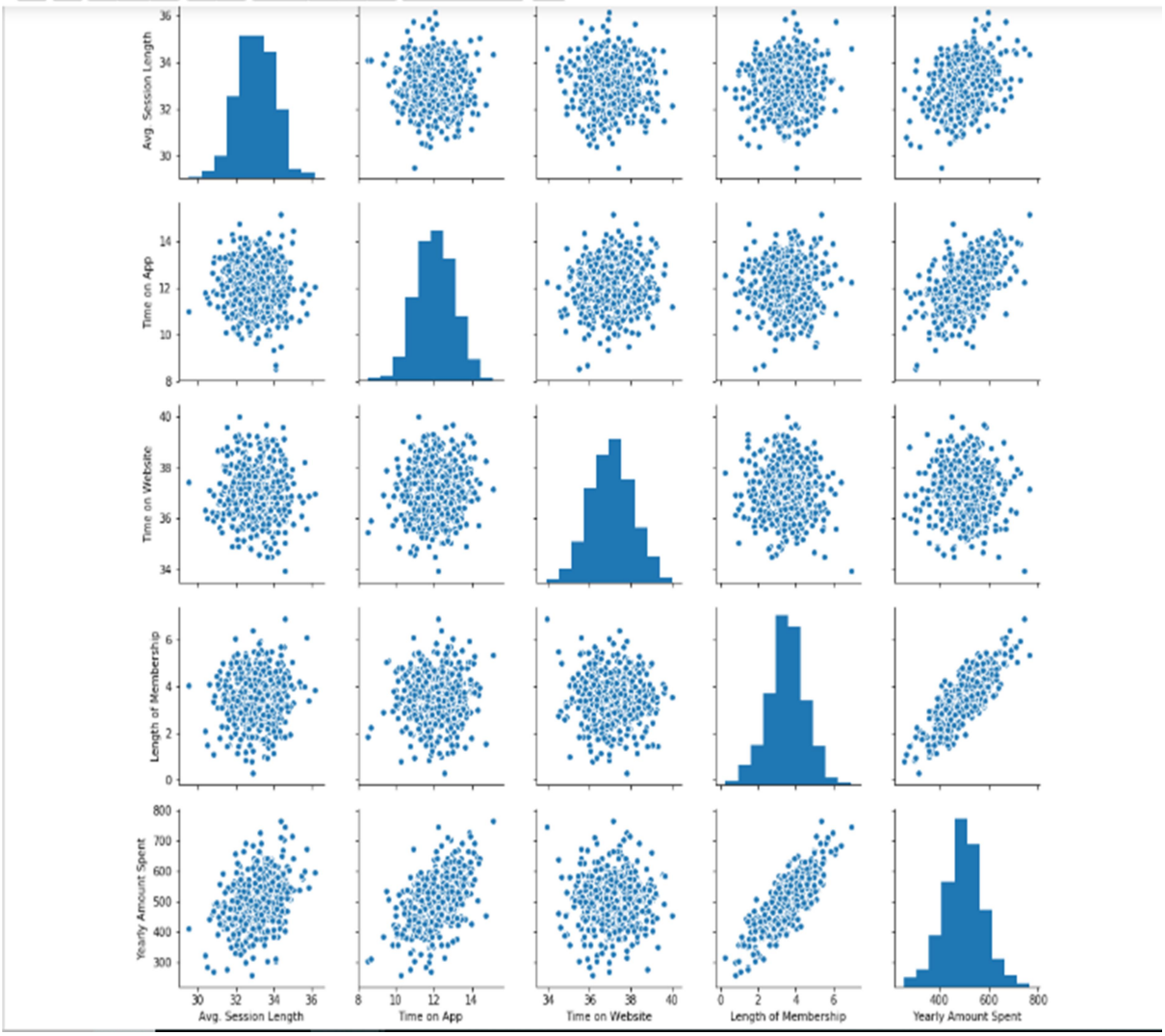
	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
count	500.000000	500.000000	500.000000	500.000000	500.000000
mean	33.053194	12.052488	37.060445	3.533462	499.314038
std	0.992563	0.994216	1.010489	0.999278	79.314782
min	29.532429	8.508152	33.913847	0.269901	256.670582
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

Exploratory Data Analysis:- only few important snap added

For the exploratory data analysis will try to creat 'Pair Plot'. Since seborn gives power to creat multiple plot in single shot and also its show multiple corelation with all numerical data inside the dataframe.

However will try to understand which are the best co-relation among and try to go more deep.

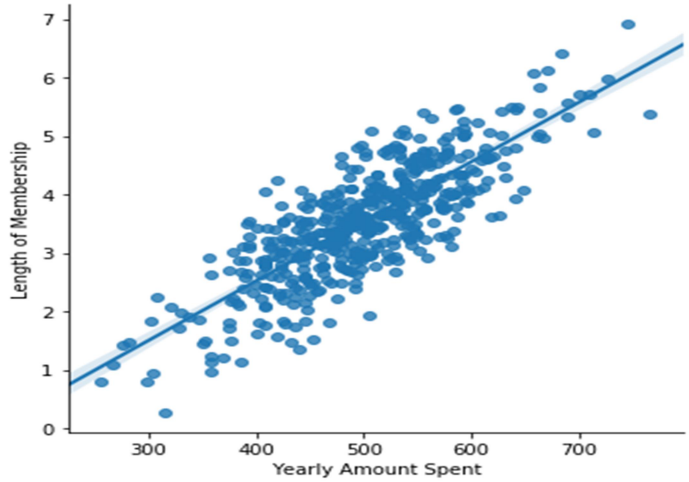
So lets see :



As per above plots we can see there are linear relationship between “Length of Membership” and “Yearly Amount Spent” so now will try to check if it have perfect line inside which goes via or nearly all dots.

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

```
<seaborn.axisgrid.FacetGrid at 0x15d02088c18>
```



➢ Training and Testing Data

Now that we've explored the data a bit, let's go ahead and split the data into training and testing sets. Set a variable X equal to the numerical features of the customers and a variable y equal to the "Yearly Amount Spent" column.

```
y = customers['Yearly Amount Spent']
```

```
X = customers[['Avg. Session Length', 'Time on App','Time on Website', 'Length of Membership']]
```

Use model_selection.train_test_split from sklearn to split the data into training and testing sets. Set test_size=0.3 and random_state=101

```
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)
```

➢ Training the Model

Now its time to train our model on our training data!

Import LinearRegression from sklearn.linear_model

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

```
lm = LinearRegression()
```

➢ Train/fit lm on the training data.

```
lm.fit(X_train,y_train)
```

➢ Predicting Test Data

```
predictions = lm.predict( X_test)
```

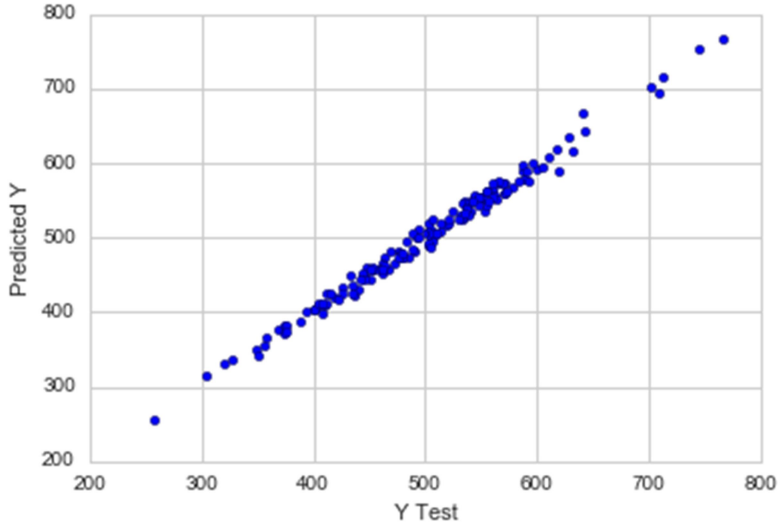
➢ Create a scatterplot of the real test values versus the predicted value

```
plt.scatter(y_test,predictions)
```

```
plt.xlabel('Y Test')
```

```
plt.ylabel('Predicted Y')
```

```
Out[296]: <matplotlib.text.Text at 0x135546320>
```



Comment on model: as we can see model has performed well and shows linear relationship with data(train and test) let check how good model via statics

➢ Evaluating the Model

from sklearn import metric

print('MAE:', metrics.mean_absolute_error(y_test, predictions))

print('MSE:', metrics.mean_squared_error(y_test, predictions))

print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))

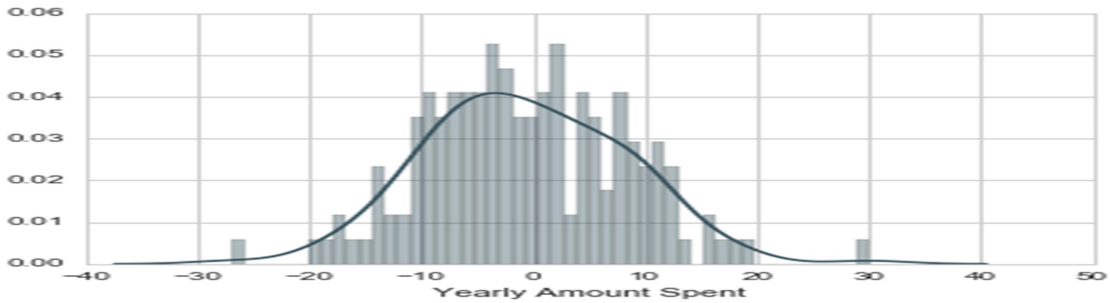
MAE: 7.22814865343
MSE: 79.813051651
RMSE: 8.93381506698

➢ Residuals

Let's quickly explore the residuals to make sure everything was okay with our data.

We can plot below histogram of the residuals and make sure it looks normally distributed.

```
sns.distplot((y_test-predictions),bins=50);
```



As model were perfromed good hence result of residuals also shows very well curves to make und erstand good data fit .

➢ Conclusion

We still want to figure out the answer to the original question, do we focus our efforst on mobile ap p or website development? Or maybe that doesn't even really matter, and Membership Time is what is really important. Let's see if we can interpret the coefficients at all to get an idea.

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

	Coeffecient
Avg. Session Length	25.981550
Time on App	38.590159
Time on Website	0.190405
Length of Membership	61.279097

➢ Interpreting the coefficients:

- Holding all other features fixed, a 1 unit increase in Avg. Session Length is associated with an increase of 25.98 total dollars spent.
- Holding all other features fixed, a 1 unit increase in Time on App is associated with an increase of 38.59 total dollars spent.
- Holding all other features fixed, a 1 unit increase in Time on Website is associated with an increase of 0.19 total dollars spent.
- Holding all other features fixed, a 1 unit increase in Length of Membership is associated with an increase of 61.27 total dollars spent.

Final comment :- as we got answer through cofficient as company need to focus more about their "LENGTH OF MEMBERSHIP".

However there are multiple factor can be affected their cost but for this data above is resolution as of now.