**Machine Learning (ML):** ML is an application of Artificial Intelligence (AI) that provides systems the ability to automatically learn themselves and improve from the experience without being explicitly programmed. ML focuses on the development of computer programs that can access data and use it to learn themselves.

**Data Set:** A collection of related sets of information that is composed of separate elements but can be manipulated as a unit by a computer.

**Data Visualisation:** It is a representation of data or information in a graph, chart, or other visual formats which is helpful to conduct analyses such as predictive analysis which can serve as helpful Visualisation to present.

**Data Cleaning:** It is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.

**Supervised Learning:** The model is trained using ‘labeled data’. Datasets are said to contain labels that contain both input and output parameters. To simplify – ‘Data is already tagged with the correct answer’.

**Simple Linear Regression:** It is a Regression Model that estimates the relationship between the independent variable and the dependent variable using a straight line .

# [y = mx + c]

**where both the variables should be quantitative.**

**Models:** Those are output by algorithms and are comprised of model data and a prediction algorithm.

**Training Model:** In supervised learning, an ML Algorithm builds a model by examining many examples and attempting to find a model that minimizes loss and improves prediction accuracy.

### These are the few terms used in machine learning while creating a model and to get familiar with. Now let’s get started with the analysis and prediction of the model.

### This is about a fictional e-commerce 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.

### Data is fictional, including Email id's data and other personally identifiable information.

### In this mini-project, I am going to use supervised data and simple linear regression for analysis and prediction. The Ultimate goal is the predict the company whether it has to focus on application or website development using the trained model to the highest achievable accuracy using available data.

### ****This is an pratical application project to generalize them any one can change data set and some parameters to get the desired output****

### ****The steps involved are:****

1. Loading the dataset.
2. Visualising the Data or exploring the data
3. Build the Model and Train it.
4. Evaluating the model.
5. Making a decision on the given data.

### ****======================Importing libraries===========================****

In [ ]:

**import** **numpy** **as** **np**

**import** **pandas** **as** **pd**

**import** **matplotlib.pyplot** **as** **plt**

**import** **seaborn** **as** **sns**

### ****====================Loading the dataset=====================****

**The 'Ecommerce Customers' dataset 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.

In [ ]:

df=pd.read\_csv("/content/Ecommerce Customers.txt",sep=",")

print("Data frame created successfuly")

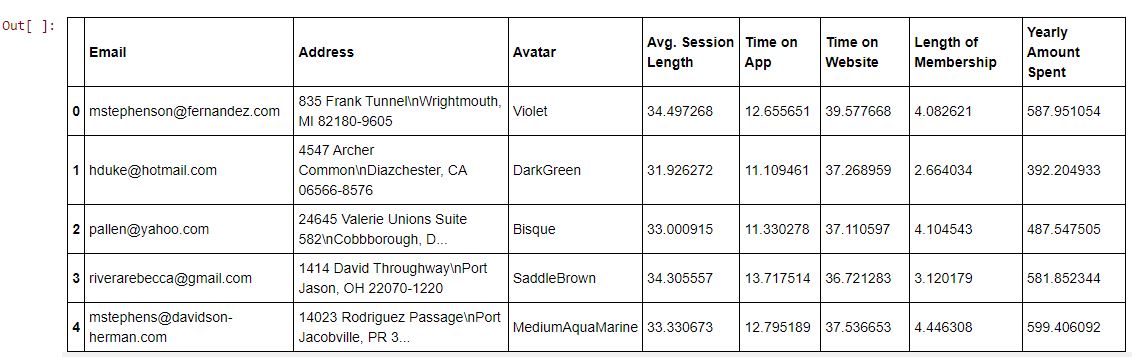
Data frame created successfuly

In [ ]:

print("top 5 rows**\n**")

df.head()

top 5 rows

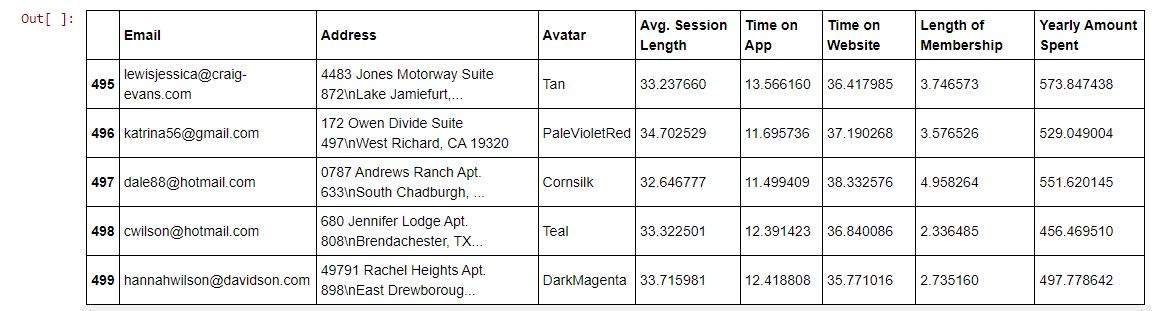


In [ ]:

print("bottom 5 rows **\n**")

df.tail()

bottom 5 rows



In [ ]:

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

In [ ]:

df.columns

Out[ ]:

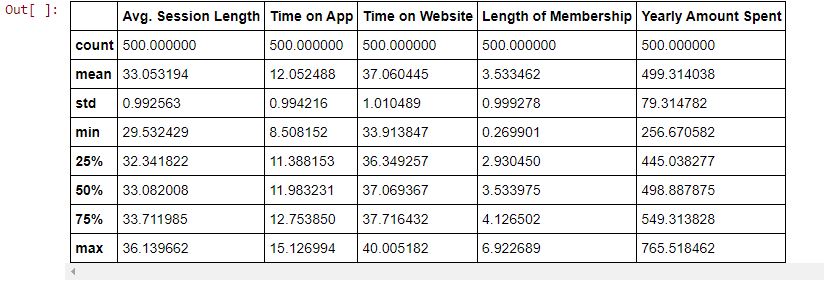
Index(['Email', 'Address', 'Avatar', 'Avg. Session Length', 'Time on App',

'Time on Website', 'Length of Membership', 'Yearly Amount Spent'],

dtype='object')

In [ ]:

df.describe()



### ==================Exploratory Data Analysis====================

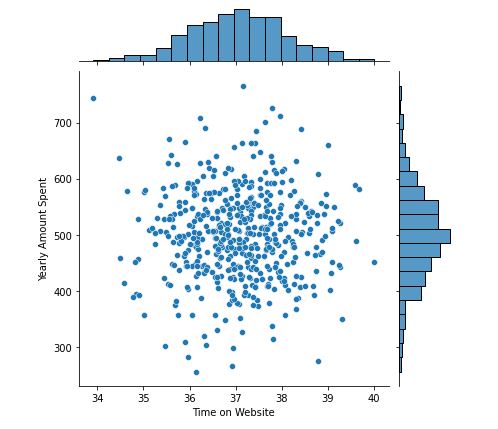
**Here we are comparing different parameters to find the strong relationship between parameters.**

In [ ]:

sns.jointplot(x='Time on Website', y='Yearly Amount Spent', data=df)

Out[ ]:

<seaborn.axisgrid.JointGrid at 0x7f7a33500f90>

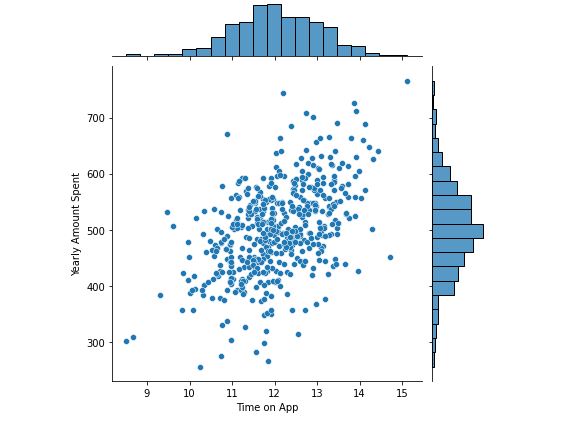


In [ ]:

sns.jointplot(x='Time on App', y='Yearly Amount Spent', data=df)

Out[ ]:

<seaborn.axisgrid.JointGrid at 0x7f7a2a022190>



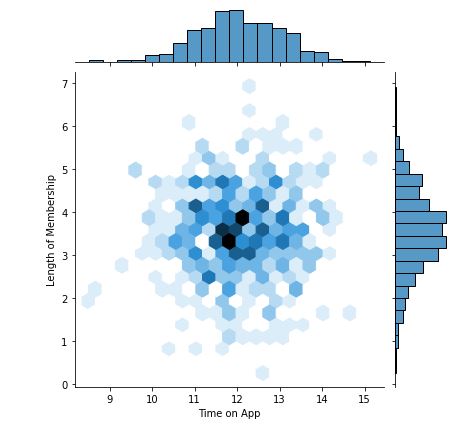
**From the plot,it seems like that there is some relationship between Yearly Amount Spent and Time on App as most of points are some how near to each other**

In [ ]:

sns.jointplot(x='Time on App', y='Length of Membership', data=df, kind='hex')

Out[ ]:

<seaborn.axisgrid.JointGrid at 0x7f7a29a5f890>



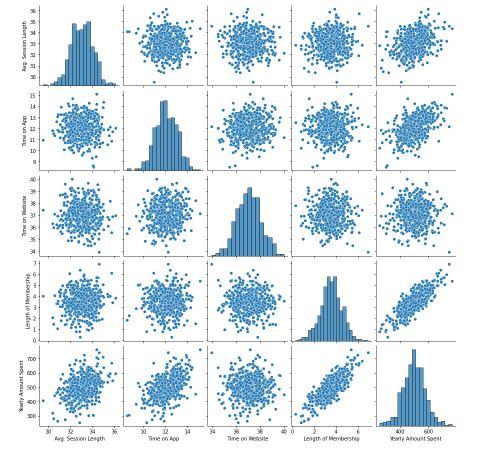
**From the plot,it seems like that there is some relationship between Yearly Amount Spent and Time on App as most of points are some how near to each other**

In [ ]:

sns.jointplot(x='Time on App', y='Length of Membership', data=df, kind='hex')

Out[ ]:

<seaborn.axisgrid.JointGrid at 0x7f7a29a5f890>



**from the above graphs we can say that there is a close relationship between Length of Membership and Yearly Amount spent**

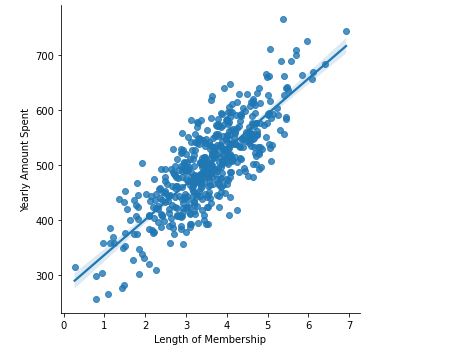
#### ****Checking the relationship of Yearly Amount Spent vs. Length of Membership using a linear model plot.****

In [ ]:

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

Out[ ]:

<seaborn.axisgrid.FacetGrid at 0x7f7a28c7db90>



### ****=================Training and Testing Data====================****

**Let's split the data into training and testing sets.**

**Setting a variable X equal to the numerical features of the customers and a variable y equal to the "Yearly Amount Spent" column.**

**Note: omitted categorical values as they have no impact on model**

In [ ]:

df.columns

Out[ ]:

Index(['Email', 'Address', 'Avatar', 'Avg. Session Length', 'Time on App',

'Time on Website', 'Length of Membership', 'Yearly Amount Spent'],

dtype='object')

In [ ]:

X = df[['Avg. Session Length', 'Time on App',

'Time on Website', 'Length of Membership']]

In [ ]:

y = df['Yearly Amount Spent']

### ****====================Building a model========================****

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, random\_state=0)

### ****=====================Traning model=========================****

In [ ]:

**from** **sklearn.linear\_model** **import** LinearRegression

In [ ]:

lm = LinearRegression()

In [ ]:

lm.fit(X\_train, y\_train)

Out[ ]:

LinearRegression(copy\_X=True, fit\_intercept=True, n\_jobs=None, normalize=False)

### ****============Printing out the coefficients of the model==============****

In [ ]:

lm.intercept\_

Out[ ]:

-1060.5508096198853

In [ ]:

lm.coef\_

Out[ ]:

array([25.88815047, 38.87046474, 0.47066154, 61.78369022])

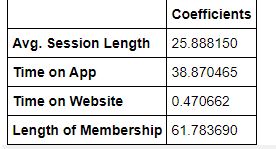
In [ ]:

cdf = pd.DataFrame(lm.coef\_, X\_train.columns, columns=['Coefficients'])

In [ ]:

Cdf

Out[ ]:



### ****=================Predicting Test Data==========================****

**Now that we have fit our model, let's evaluate its performance by predicting off the test values.**

In [ ]:

predictions = lm.predict(X\_test)

In [ ]:

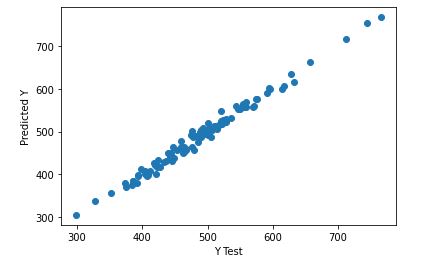
plt.xlabel('Y Test')

plt.ylabel('Predicted Y')

plt.scatter(y\_test, predictions)

Out[ ]:

<matplotlib.collections.PathCollection at 0x7f7a222f47d0>



### ****======================Evaluating the Model=========================****

**Let's evaluate our model performance by calculating the residual sum of squares and the explained variance score (R^2).**

In [ ]:

**from** **sklearn** **import** metrics

In [ ]:

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: 7.645674798915295

MSE: 92.89010304498562

RMSE: 9.637951185028156

In [ ]:

metrics.explained\_variance\_score(y\_test, predictions)

Out[ ]:

0.9862059058088554

In [ ]:

cdf = pd.DataFrame(lm.coef\_, X\_train.columns, columns=['Coefficient'])

cdf

Out[ ]:

# =================CONCLUSION=====================

**Given that the coefficients are all positive, for every unit change in the features, the average yearly spend increases by the coefficient holding all other features fixed. In this case, the most important factor seems to be the length of membership of a customer.**

**Should the company focus more on their mobile app or on their website?**

**If the company really needs to choose now between the two, they should focus more on their mobile app as it has a bigger influence on yearly spend based on the length of time the customers spend on it. It would be also good to explore the relationship between how long a customer has been a member (length of membership) and the time they spend on the app and website. That might yield some better conclusions and action plans for the company.**