Evolution of e-Commerce and Consumers' Behaviour to Forecast Future E-Commerce Trends

Group RL-1

WQD7001 Principles of Data Science

Prof. Dr. Rohana Binti Mahmud

University of Malaya

Name	Matrix No
NUR SHAFIQAH BINTI MOHAMAD JOHARI	22119564
LAW JIA JIN	22071390
LIM CHENG YANG	22100811
LIM SZE SING	22109557
GAN JING WEN	22065433

1.0 Project Background

The e-commerce landscape is continuously evolving, with businesses facing increasing challenges in understanding and adapting to the dynamic market conditions. To thrive in this competitive environment, the power of data is crucial to help aid businesses determine important elements to boost their sales. Our data science project aims to leverage advanced analytics and predictive techniques to optimize various aspects of the e-commerce process.

2.0 Project Objectives

Our project aims **to Investigate and analyze the customers' behaviors on the e-commerce platform.** This includes understanding purchasing patterns, identifying top contributor elements for sales, and predicting customer preferences.

To achieve this, we aim **to evaluate the most suitable analysis methodology** that best suited the target business and further aid the decision making. This will help the businesses to understand the consumer practicality and produce business strategy.

We will then **predict the forecast of yearly amount spent** based on historical data that will aid in optimizing inventory management for future sales.

3.0 Data Modelling

Code reference from Google Colab: https://colab.research.google.com/drive/1gwUrXyUPe2j6GyaFd3Ei3nNHnDvylSfZ?usp=sharing
For modeling purposes, we employ two machine learning algorithms to train our models: decision tree regressions and multiple linear regressions, respectively. The models undergo train-test split, dividing the data into 80% for training and 20% for testing. Below shows the examples of training processes of both models in form of Python codes:

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=8)
```

```
model = LinearRegression()
model.fit(X_train, Y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size= 0.2, random_state=8)
```

```
model = DecisionTreeRegressor(criterion='squared_error', random_state=8)
model.fit(x_train, y_train)
```

```
DecisionTreeRegressor
DecisionTreeRegressor(random_state=8)
```

Regression model was selected as the machine learning approach due to several factors related to the characteristics of our dataset. All variables in our dataset—namely, the yearly amount spent, the length of membership, and the amount of time spent—are quantitative in nature. This quality fits in nicely with regression analysis, which excels at working with numerical variables and determining correlations between them.

Additionally, our dataset shows relationships among the amount of time users spend on the e-commerce platform, the duration of their membership, and the annual amount they spend. The way these variables interact indicates that there may be a relationship worth investigating, and regression modelling offers an appropriate framework for doing that. Our goal is to determine and measure the impact that membership duration and time spent have on users' annual spending on the e-commerce platform using regression model.

Simply speaking, the regression model is a useful instrument for understanding the complex dynamics present in our dataset, which enables us to forecast the effects of membership duration and time spent on users' spending with the e-commerce platform. This method offers a systematic and analytical framework to provide insights into the primary drivers of user spending behaviors in this specific environment, which is in line with the quantitative nature of our variables.

In the domain of multiple linear regression, the integration of specific metrics aims to thoroughly assess the model's performance and interpretability. A clear indicator of the accuracy of the model is the Root Mean Squared Error (RMSE), which is essential for estimating the average magnitude of errors between anticipated and actual values. Concurrently, the application of R-squared, which is equivalent to the model's accuracy score, becomes essential in assessing how well the model explains the variance in the target variable. An increased R-squared value indicates that the model fits the data better. Furthermore, examining the coefficient values in multiple linear regression is still essential since it clarifies the direction and strength of correlations between independent variables and the target variable, which helps assess the prediction accuracy of the model.

Likewise, in the case of decision tree regression, a unique collection of measures is employed to evaluate the predictive power and overall performance of the model. Considering its sensitivity to greater errors—a crucial factor in regression problems—RMSE continues to have significance in

assessing prediction accuracy. R-squared gains significance in the setting of decision tree regression by offering information on how successfully the model translates response variable variability into model correctness. Furthermore, decision trees provide feature importance scores by default, which makes it easier to identify important factors. This feature importance analysis is instrumental in discerning which predictors significantly contribute to the model's accuracy, offering valuable insights for feature selection or engineering.

In summary, a comprehensive understanding of the accuracy, interpretability, and predictive powers of multiple linear regression and decision tree regression models can be obtained by combining RMSE, R-squared (known as accuracy), feature importance, and coefficient analysis in the evaluation process. Each indicator has a unique function that works together to provide a thorough evaluation of the models' performance and to support informed decision-making when it comes to data analysis and prediction.

4.0 Data Interpretation

In any modelling processes , the outcome would not be fully developed with the creation of the model itself. The subsequent phase of data interpretation plays an important role in deriving insights, refining strategies, and guiding decision-making processes. Once the model has been developed, it is crucial to comprehend its outcomes, understand the implications, and harness the power of the generated insights.

In this section we will go deep dive into data interpretation and how we can interpret the outcomes of our models.

4.1 Regression Modelling

Regression is a statistical method used to present fundamentals of linear modelling. It yields a mathematical equation that estimates a dependent variable, often known as Y, from a set of independent variable(s), often known as X.

The regression model works by estimating coefficients for each independent variable. These coefficients represent the slope of the relationship between the independent variable(s) and the dependent variable. For instance, in a simple linear regression equation is as Equation 1 below:

$$Y = b_0 + b_1X_1 + b_2X_2 + \cdots + b_kX_k + e \quad (1)$$

- Y is the dependent variable.
- X_1, X_2,…, X_k are the independent variables or predictors that influence the dependent variable.
- b_0, b_1, b_2….b_k are the regression coefficients associated with each independent variable. They represent the weights or slopes that quantify the impact of the corresponding predictors on the dependent variable.
- e represents the error term (the difference between the predicted and actual *Y* values not explained by the model).

There are various types of regression models, but in our project we will narrow down to only two models below that we think are fit to our dataset:

1. **Decision Tree Regression:**
    a. A tree-like structure to make predictions where the dataset is split based on features to create homogenous subsets
    b. A model that is able to handle nonlinear relationships and suitable for various prediction tasks from different domains
2. **Multiple Linear Regression:**
    a. Involves more than one independent variable to predict the dependent variable
    b. It assumes a linear relationship between the dependent and multiple independent variables.

4.1.1 Interpreting Decision Tree Regression Model Outcomes

Throughout the process of our modelling structure, we structured the models as such as there are two independent variables or X values. We have used **Feature Importance** as our metrics for the goodness of fit

Feature Importance refers to a technique that assigns a score to features based on how much each feature contributes to reducing impurity within decision tree nodes. This can be calculated by using Equation 2 **:**

$$\text{Feature Importance Score for } X_i = \frac{\text{Total decrease in impurity due to } X_i}{\text{Sum of total decreases in impurity for all features}} \quad (2)$$

The node probability can be calculated as such, and the higher the value the more important the feature will be.

Below is the outcome of Feature Importance for X values in our model:

Table 1 : *Decision Tree Regression Feature Importance*

X Value	Y Value	Feature Importance
Length of Membership	Yearly Amount Spent	0.79
Time Spent	Yearly Amount Spent	0.21

Key takeaways that we can get from each variables' Feature Importance :

- **Length of Membership as X Value:**
  - The feature importance of 0.79 can be considered relatively high and implies that this factor has a big influence to the yearly amount spent of the customers
  - A higher value associated with this variable indicates that as the length of membership increases, customers tend to spend more annually
- **Time Spent as X Value:**
  - Although comparatively lower than the length of membership, 0.21 feature importance is still a notable figure to give an impact to the yearly amount spent
  - This indicates that the more the customer spends time or be more engaged in the e-commerce platform, the more tendency they have to spend on products

Overall, the modelling outcome underscores the "Length of Membership" as the most influential factor in predicting yearly spendings of the customers, and "Time Spent" follows as a significant contributor.

4.1.2 Interpreting Multiple Linear Regression Model Outcomes

Similarly to our Multiple Linear Regression Model, there are two independent variables or X values. However for the measurement of the model's goodness of fit, we evaluated it as **Coefficients.**

Coefficients are prominent aspect that can change the outcome of a regression model. It is part of the formula in Equation 1 as mentioned previously, and Coefficient is defined as :

> "b_0 , b_1 , b_2 …. b_k *are the regression* ***coefficients*** *associated with each independent variable. They represent the weights or slopes that quantify the impact of the corresponding predictors on the dependent variable. "*

This indicates that the magnitude of the coefficients reflects the strength of the relationship between predictor and the target. Hence, larger value of coefficients indicates stronger effects to the equation.

Below is the outcome of Coefficients for X values in our model::

Table 2 : *Multiple Linear Regression Coefficients*

X Value	Y Value	Coefficients
Length of Membership	Yearly Amount Spent	64.98
Time Spent	Yearly Amount Spent	17.83

Key takeaways that we can get from each variables' Coefficients :

- **Length of Membership as X Value:**
  - Coefficient of 64.98 implies that each additional unit of "Length of Membership" corresponds to an increase of approximately $64.98 in the "Yearly Amount Spent"
  - This indicates that customers with longer membership duration produces more expenditure annually
- **Time Spent as X Value:**
  - Coefficient of 17.83 implies that each additional unit of "Time Spent" corresponds to an increase of approximately $17.83 in the "Yearly Amount Spent"
  - This indicates that customers that spends more time in the platform contributes to more expenditure annually

4.2 Regression Performance Metrics

In any Regression Modelling tasks, it is crucial to score the models to evaluate their performances and ensure that deployed models meet the required standards of accuracy, fairness, and reliability in any scenarios.

There are various metrics that can be applied in model development lifecycle, however in our project, we will narrow down the metrics into **Accuracy** and **Root Mean Squared Error (RMSE)** as it is deemed fit to our project.

Accuracy is a fundamental metric where It calculates the percentage between the number of correct predictions and the total number of predictions. Refer to Equation 2 :

$$\text{accuracy} = \frac{\#\text{ correct predictions}}{\#\text{ total data points}} \tag{3}$$

Which can be further granulated as Equation 3 :

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4}$$

- **True Positive (TP):** Instances that belong to the positive class and are correctly classified as positive.
- **True Negative (TN):** Instances that belong to the negative class and are correctly classified as negative.
- **False Positive (FP):** Instances that belong to the negative class but are incorrectly classified as positive (Type I error).
- **False Negative (FN):** Instances that belong to the positive class but are incorrectly classified as negative (Type II error).

An accuracy score of 1.0 or 100% indicates perfect predictions, while 0.0 or 0% indicates no correct predictions are available in the dataset. Hence, we can conclude that the higher the percentage of the accuracy score, the higher the precision of the model.

On the other hand**, Root Mean Squared Error (RMSE)** is a commonly used metric for regression tasks, it is defined as :

$$\text{RMSE} = \sqrt{\frac{\sum_i (y_i - \hat{y}_i)^2}{n}} \tag{5}$$

- y_i is the true score for the ith data point
- $\hat{y}_i$ is the predicted value
- n is the number of data points

RMSE equation involves calculating the average squared differences between predicted and actual values, then deviates between the two. Lower values indicate better predictive performance and lesser errors.

4.2.1 Interpreting Models' Performances

The performance of our Machine Learning Models have been evaluated using the approach of two key metrics as mentioned previously. Below table is the details of the performances outcome :

Table 3 : *Models' Performance Metrics*

Performance Metrics	Decision Tree Regressor	Multiple Linear Regressor
Accuracy (%)	62.84%	72.88%
Root Mean Squared Error (RMSE)	48.399	41.343

- **Decision Tree Regression:**
    - The accuracy of this model stands at 62.84% which is slightly lower than Multiple Linear Regression, indicating that the model is still relatively accurate but less as compared to the latter model
    - The RMSE is calculated as 48.399 which is higher than Multiple Linear Regression, indicating that the average deviation between predicted values and actual values is slightly higher than the other model
- **Multiple Linear Regression :**
    - The accuracy of this model stands at 72.88% which is higher than Decision Tree Regression, indicating that the model is more accurate than the other
    - The RMSE is calculated as 41.343 which is lower than Multiple Linear Regression, indicating that the average deviation between predicted values and actual values is better than the other model

Upon examining both models' performances, it's evident that Multiple Linear Regression model performs better than the Decision Tree Regressor in this scenario.

4.3 Models' Outcomes Conclusion

Based on the testing and outcomes of both models, we can conclude it as below :

- **Length of Membership as X Value:**
    - Both models show high value for this variable during testing
    - This highlights that e-commerce businesses should imply memberships and gauge customer's interest for a more extended membership duration which will further increase the yearly revenue
- **Time Spent as X Value:**
    - Both models shows that although not as significant from the "Length of Membership" variable, this variable still have an impact to the "Yearly Amount Spent"

    - This highlights the importance of retaining the users to be engaged to the e-commerce platform for a longer time by maintaining a user-friendly app/website interface or engaging content
- **Models' Performances**
    - Upon examining both models' performances, it's evident that Multiple Linear Regression model performs better than the Decision Tree Regressor in this scenario
    - Although Decision Tree Regressor can still be an acceptable model, however Multiple Linear Regression demonstrated superior performance that highlights a more precise predictions with smaller deviations from the actual values

In conclusion, both models seem to agree on the relationship directionality; both indicate that an increase in 'Length of Membership' and 'Time Spent' is associated with higher 'Yearly Amount Spent'. However, the Multiple Linear Regression model is the preferred choice for predicting the target variables.

4.4 Strategic Recommendations Based on Data Interpretation

Based on the outcomes of both models, an e-commerce platform can provide strategic insights for improving business performance. Here are some strategic recommendations based on the outcomes of our model :

a) Emphasize Customer Retention and Loyalty Programs by giving offer exclusive benefits, rewards, or incentives for long-term members to enhance their loyalty in memberships and increase their spending over time
b) Enhance user engagement and experience in e-commerce app/website such as enhancing UI/UX of said platforms, personalized contents, product recommendations etc. to increase user's engagement and time spent on platform
c) Segment customers based on their length of membership, engagement patterns, and spending behaviours and offer tailored services, discounts, or vouchers to meet the needs of segmented customers
d) Implement Multiple Linear Regression Machine Learning and deploy the model to further monitor customer's behaviour, track the effectiveness of implemented strategies, and iteratively refine marketing, user experience, and loyalty programs based on ongoing analysis and feedback

5.0 Deployment of Data Product

There are 5 broad groups of data products functions, which are raw data, derived data, algorithms, decision support, and automated decision-making. The internal complexity of product types are increasing based on the list mentioned, while the user's experience will be less complexity which is vice versa. In this project, we have higher preference to choose decision support as the suggested data product in deployment plan. The suggestion have been made by considering the project's objective and the balanced mix of technical and non-technical users which are tend to be both the internal products in an organization and technical product at the same time.

For user interface (UI), there are several methods to be chosen such as APIs, Dashboards, and web elements. However, by considering the users will be mix of both technical and non-technical users, the dashboards data visualization tool will be selected to implement in this project. Dashboards are a data visualization tool that allow all users to understand the analytics that matter to their business. The users can simply input the data to achieve the predictions results.

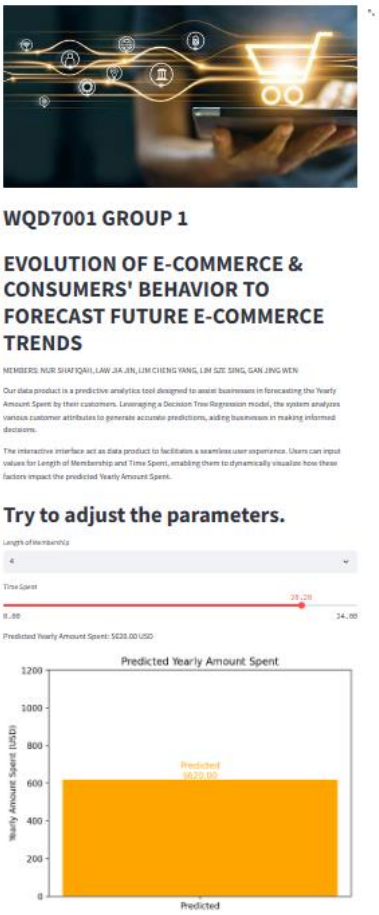


Figure 1 : *Data Product Interface*

For this project, Streamlit will be the Python library for creating web applications to show the prototype for visualization and enable deployment through Streamlit sharing. The data product can be access through the link, [Streamlit (appuctsimplegit-497ccl7svcarbzxeagzv8s.streamlit.app).](appuctsimplegit-497ccl7svcarbzxeagzv8s.streamlit.app) Interactive functionality of Streamlit interface can provide visualized insights and enable user's interaction with data.

By considering the security of the data product, the secure data transmission, user authentication, and secure storage of sensitive information such as model, transaction history and user's email will be protected. For secure data transmission, the secure communication protocols and encrypt data need to be implemented in transit to prevent data interception. In user authentication, the organization need to ensure that only authorized users can access the data product. Lastly, administration control must practice as basic employee training and regulation to ensure they understand and doing the best practices.

Also, the scalability is one of the concern during real application of data product. One of the method to ensure the data product is scalable to handle increased usage is to use the cloud service such as Amazon Web Services (AWS), and Google Cloud Platform(GCP). By keeping the data at cloud storage and integrate the trained model to the clod & compute resources, the scalability, reliability, and accessibility will be improved significantly.

6.0 Plan for Reproducible Research

For this project, comprehensive steps have been taken to ensure transparency and replicability.
Firstly, the source of our project has been documented, recording the URL and the time accessed from GitHub repository which is publicly available. This not only establishes a clear reference point for our work but also allows others to access the exact version of the project at any given time.

The entire coding process has been conducted in Google Colab, a collaborative and cloud-based environment, facilitating ease of access and reproducibility. The Colab notebook was structured with clear sections, each addressing specific tasks or analyses. Moreover, Google Drive was utilized to integrate with Google Colab for seamless data storage and retrieval.

In the report for the project, each step of the project has been documented, starting from data cleaning, exploratory data analysis (EDA), modelling, and extending to model evaluation.
The evaluation process incorporates measures of uncertainty, providing a comprehensive understanding of the model's performance. The documentation in the report includes code, comments, and explanations to ensure that the methodology is transparent and easily reproducible by other researchers.

Result interpretation is a crucial aspect, and the report has included detailed descriptions, correlation analyses, and the interpretation of coefficients. The findings have been discussed for the implications and limitations, promoting a thorough understanding of the research outcomes.

By adopting a collaborative coding environment (Google Colab) and providing a detailed and well-documented report, the foundation for reproducibility and transparency in the research project has been established. This approach enables others to reproduce the analyses, validate the results, and build upon the work in the project for further research.

7.0 Insights and Conclusion

We have undertaken a comprehensive analysis of customer spending patterns, utilizing two key independent variables: total time spent and length of membership. Our primary goal is to develop a predictive model that can assist organizations in estimating a customer's total spend, which can have substantial implications for decision-making and marketing strategies. Below, we summarize the key insights and conclusions drawn from our analysis:

7.1 Insights:

a) Length of membership and Spending: Our analysis revealed a strong positive correlation between the length of membership and the total amount spent by customers. This suggests that as customers stay loyal to the organization for a longer duration, they tend to spend more, indicating the importance of customer retention efforts.

b) Time Spent and Spending: Even though the total time spent by customers doesn't show a strong positive relationship with total spending, but the time spent on App had showed a strong positive relationship with total spending. This implies that customers who engage more with the organization, especially through phone application, tend to spend more, indicating the importance of investing in phone application.

c) Model Performance: We developed and evaluated Decision Tree and Multiple Linear Regression predictive models. Among this two models, Multiple Linear Regression showcased a high accuracy and low RMSE, indicating its ability to explain the variance in total spending.

7.2 Conclusion:

By leveraging the predictive model developed in this study, the organization can benefit in the following ways:

a) Revenue Forecasting: The model can be used to forecast future revenues based on the length of membership and total time spent by customers. This enables the organization to plan budgets and resources effectively.

b) Personalized Marketing: Understanding the impact of membership duration and engagement on spending allows for more personalized marketing strategies. Tailored promotions and incentives can be offered to customers to encourage higher spending.

c) Retention Strategies: The positive correlation between length of membership and spending highlights the importance of customer retention. The organization can focus on improving customer satisfaction and loyalty to maximize revenue from existing customers.

d) Product Recommendations: The model can be integrated into the organization's recommendation system to suggest products or services to customers based on their historical data, increasing the likelihood of purchases.

In summary, this data science project equips the organization with valuable tools and insights to optimize its marketing efforts, improve customer retention, and enhance overall business performance. The predictive model, combined with a focus on understanding customer behaviour, can lead to a more profitable and sustainable future for the organization.

8.0 References

1. Darlington, R. B. (2016, September 27). *Regression analysis and linear models: Concepts, applications, and implementation*. Google Books. https://books.google.com/books/about/Regression_Analysis_and_Linear_Models.html?id=YDgoDAAAQBAJ
2. Zheng, A. (2015). *Evaluating Machine Learning Models: A Beginner's Guide to Key Concepts and Pitfalls.* Google Books. https://www.google.com.my/books/edition/Evaluating_Machine_Learning_Models/OFhauwEACAAJ?hl=en
3. Ronaghan, S. (2022, February 15). The mathematics of decision trees, random forest and feature importance in Scikit-Learn and Spark. *Medium*. https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3

9.0 Appendix

WQD7001 Principles of Data Science Assignment GA2 Google Colab Codes

Project title: Evolution of e-Commerce and Consumers' Behavior to Forecast Future E-Commerce Trends

#Project Objectives

- To Investigate and analyze the customers' behaviors on the e-commerce platform. This includes understanding purchasing patterns, identifying top contributor elements for sales and predicting customer preferences.
- To evaluate the most suitable analysis methodology that best suited the target business and further aid the decision making. This will help the businesses to understand the consumer practicality and produce business strategy.
- To develop predictive models to forecast product demand based on historical data that will aid in optimizing inventory management for future sales.

Dataset:

https://github.com/Aditya-Karanth/Machine-Learning/blob/c75ec04c3e2d755cb79dd27b908bf51fa0660eac/Sample%20Projects/Ecommerce%20Customers%20(Multiple_Linear_Regression)/Ecommerce%20Customers

#Extracting and Importing Data

```
import pandas as pd     #read from text file with comma as separator
# Data
import numpy as np
# Visualization
import matplotlib  #data visualization library
import matplotlib.pyplot as plt  #'pyplot' module to customize plots
import seaborn as sns  #library of data visualization on top of Matplotlib
for statistical graphics
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor, plot_tree
from sklearn import metrics
from sklearn.metrics import mean_squared_error, accuracy_score,
precision_score, recall_score, f1_score, confusion_matrix, r2_score
from sklearn.linear_model import LinearRegression

matplotlib.rcParams['figure.figsize'] = (10,6)  #To set the default figure
size of Matplolib
df = pd.read_csv('/content/Ecommerce Customers.csv', sep=",")
#To configure to display all columns in just one line.
#By default is true, it will display the dataset based on your screen size,
if have many columns and it cannot fit into the screen, the columns will be
display at the bottom.
pd.set_option('expand_frame_repr', False)
```

```
#To configure to display all the rows. By default, says if have 500 rows of
data, it will show only the head & tail of the data
pd.set_option('display.max_rows', None)
#print only the top 5 rows
print(df.head(5))
#print the whole dataset
#print(df)
#Display only row 1 data
#print(df.loc[0])
#Display row 1 and 2
#print(df.loc[[0, 1]])

#Check if there's any duplicate values in Avatar column
boolean = df['Avatar'].duplicated().any()
print(boolean)
```

```
                          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
True
```

```
df.tail()
```

```
                          Email
Address         Avatar  Avg. Session Length  Time on App  Time on Website
Length of Membership  Yearly Amount Spent
495  lewisjessica@craig-evans.com  4483 Jones Motorway Suite 872\nLake
Jamiefurt,...            Tan            33.237660    13.566160
36.417985              3.746573           573.847438
496           katrina56@gmail.com  172 Owen Divide Suite 497\nWest Richard,
CA 19320  PaleVioletRed            34.702529    11.695736        37.190268
3.576526           529.049004
497            dale88@hotmail.com  0787 Andrews Ranch Apt. 633\nSouth
Chadburgh, ...       Cornsilk            32.646777    11.499409
38.332576              4.958264           551.620145
```

```
498           cwilson@hotmail.com  680 Jennifer Lodge Apt.
808\nBrendachester, TX...           Teal            33.322501    12.391423
36.840086              2.336485          456.469510
499     hannahwilson@davidson.com  49791 Rachel Heights Apt. 898\nEast
Drewboroug...    DarkMagenta           33.715981    12.418808
35.771016              2.735160          497.778642
```

```
df.shape
```

```
(500, 8)
```

#Data Pre-Processing

```
df.isnull().sum()
```

```
Email                   0
Address                 0
Avatar                  0
Avg. Session Length     0
Time on App             0
Time on Website         0
Length of Membership    0
Yearly Amount Spent     0
dtype: int64
```

```
df['Time Spent'] = df['Time on App'] + df['Time on Website']

df.drop('Avg. Session Length', axis=1, inplace=True)
df.drop('Email', axis=1, inplace=True)
df.drop('Address', axis=1, inplace=True)
df.drop('Avatar', axis=1, inplace=True)
#By default is true, it will display the dataset based on your screen size,
if have many columns and it cannot fit into the screen, the columns will be
display at the bottom.
pd.set_option('expand_frame_repr', False)
#To configure to display all the rows. By default, says if have 500 rows of
data, it will show only the head & tail of the data
pd.set_option('display.max_rows', None)
#print only the top 5 rows
print(df.head(5))
```

```
   Time on App  Time on Website  Length of Membership  Yearly Amount Spent
Time Spent
0    12.655651        39.577668              4.082621           587.951054
52.233319
1    11.109461        37.268959              2.664034           392.204933
48.378420
2    11.330278        37.110597              4.104543           487.547505
48.440875
3    13.717514        36.721283              3.120179           581.852344
50.438796
```

```
4    12.795189        37.536653              4.446308           599.406092
50.331842
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Time on App           500 non-null    float64
 1   Time on Website       500 non-null    float64
 2   Length of Membership  500 non-null    float64
 3   Yearly Amount Spent   500 non-null    float64
 4   Time Spent            500 non-null    float64
dtypes: float64(5)
memory usage: 19.7 KB
```

Ther are 9 columns in total.

#Exploratory Data Analysis

```
df.describe()  #To get count/mean/std/min/max
```

```
       Time on App  Time on Website  Length of Membership  Yearly Amount
Spent  Time Spent
count   500.000000       500.000000            500.000000
500.000000  500.000000
mean     12.052488        37.060445              3.533462
499.314038   49.112933
std       0.994216         1.010489              0.999278
79.314782    1.474820
min       8.508152        33.913847              0.269901
256.670582   43.970552
25%      11.388153        36.349257              2.930450
445.038277   48.192759
50%      11.983231        37.069367              3.533975
498.887875   49.056487
75%      12.753850        37.716432              4.126502
549.313828   50.098144
max      15.126994        40.005182              6.922689
765.518462   53.112026
```

In this project, categorical and continuous variable are used to perform EDA to get overall pictures of data set for the project.

Type of Variables

a. Categorical Variable:

- nominal or discrete variable that represents integer data that can take on distinct categories or labels.

- Do not have natural order or numerical value.
- Suitable tools: frequency table, bar charts, and statistics graph

  Example: Gender, Colour, City

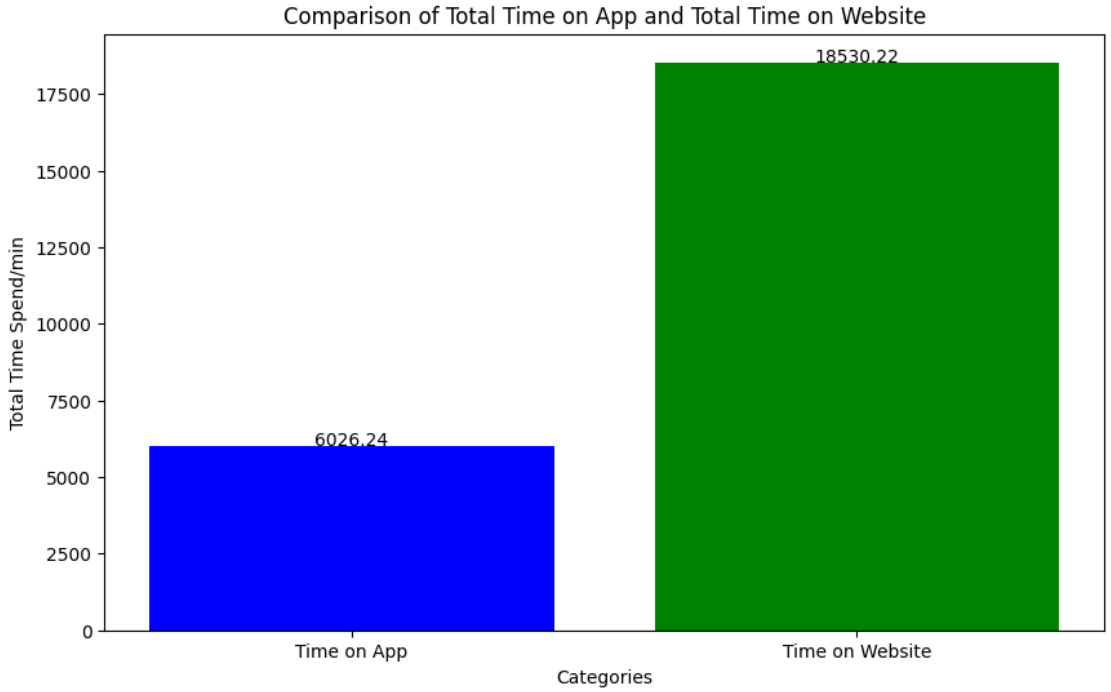
1. Multivariate Analysis

- Involve two or more variables.
- Use to explore the relationships between the variables.

```
#Preprocess the data: Calculate total sum for Time on App and Website
Totaltime_app = df["Time on App"].sum()
Totaltime_website = df["Time on Website"].sum()
print(Totaltime_app)
print(Totaltime_website)
```

```
6026.243968580483
18530.22271049034
```

```
#Categories & values for the bars
values = [Totaltime_app,Totaltime_website]
categories = ["Time on App","Time on Website"]

#set up barplot environment
bar = plt.bar(categories, values, color=['blue','green'])
for bar, value in zip(bar, values):
  plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height() + 5,
f'{value:.2f}', ha='center')

#create bar plot
plt.xlabel("Categories")
plt.ylabel("Total Time Spend/min")
plt.title("Comparison of Total Time on App and Total Time on Website")
plt.show()
```



b.Continuous Variable:

- numerical variable that represents infinite number of values within a specified range.
- With precision data which may vary continuously.
- Suitable tools: Histogram, Box plots, central tendency

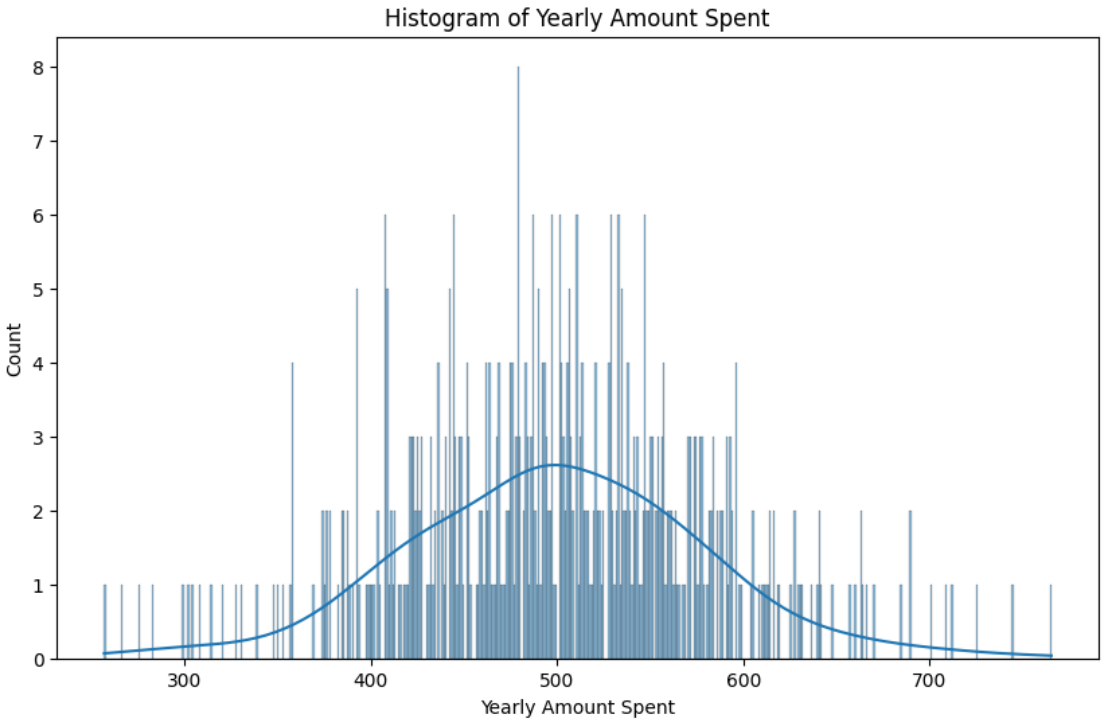
  Example: Age, Height, Income

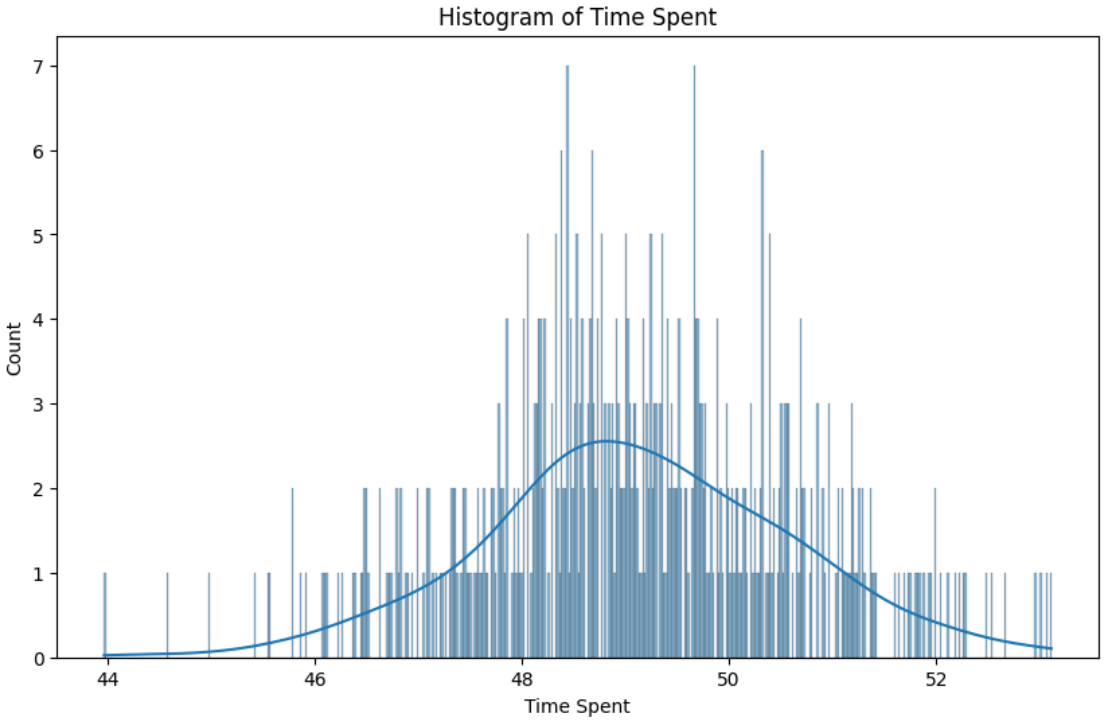
1. Univariate Analysis

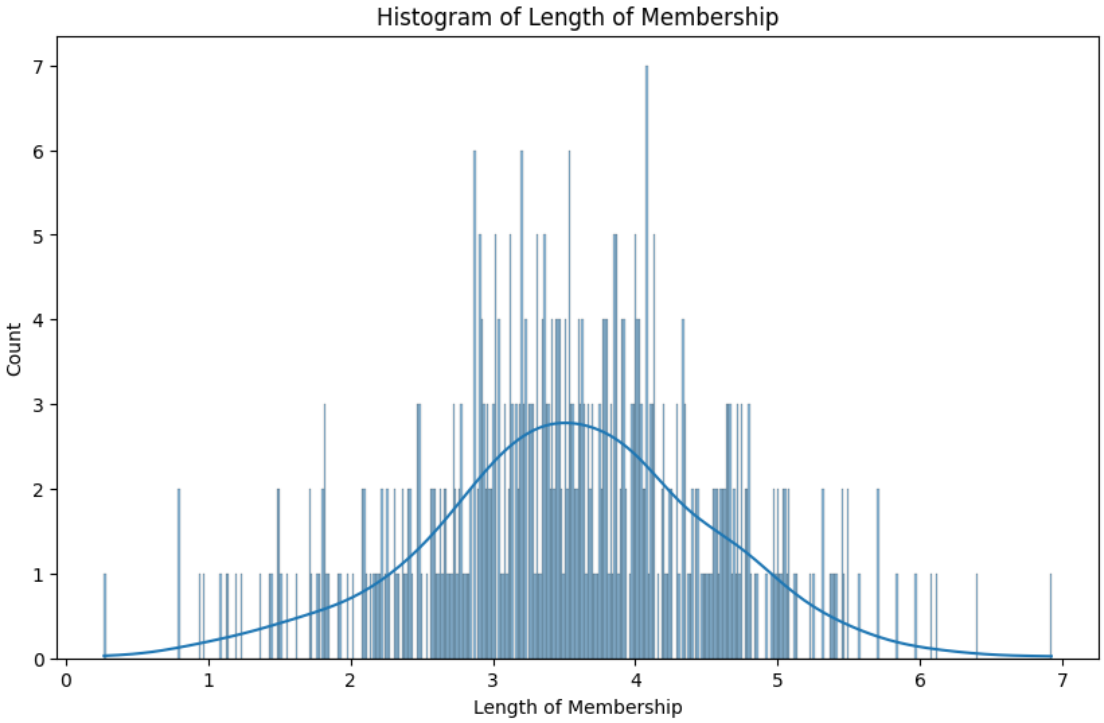
- Focus on single variable at one time.
- Use to check the central tendency and spread of variable.
- Useful to detect any outliers or abnormality in single variable.

```
#Histogram is used to shows frequency of values.

#Yearly amount spent
sns.histplot(data=df, x='Yearly Amount Spent', bins=500, kde=True)
plt.title("Histogram of Yearly Amount Spent")
plt.show()

#Time Spent
sns.histplot(data=df, x='Time Spent', bins=500, kde=True)
plt.title("Histogram of Time Spent")
plt.show()

#Length of membership
sns.histplot(data=df, x='Length of Membership', bins=500, kde=True)
```

```
plt.title("Histogram of Length of Membership")
plt.show()
```

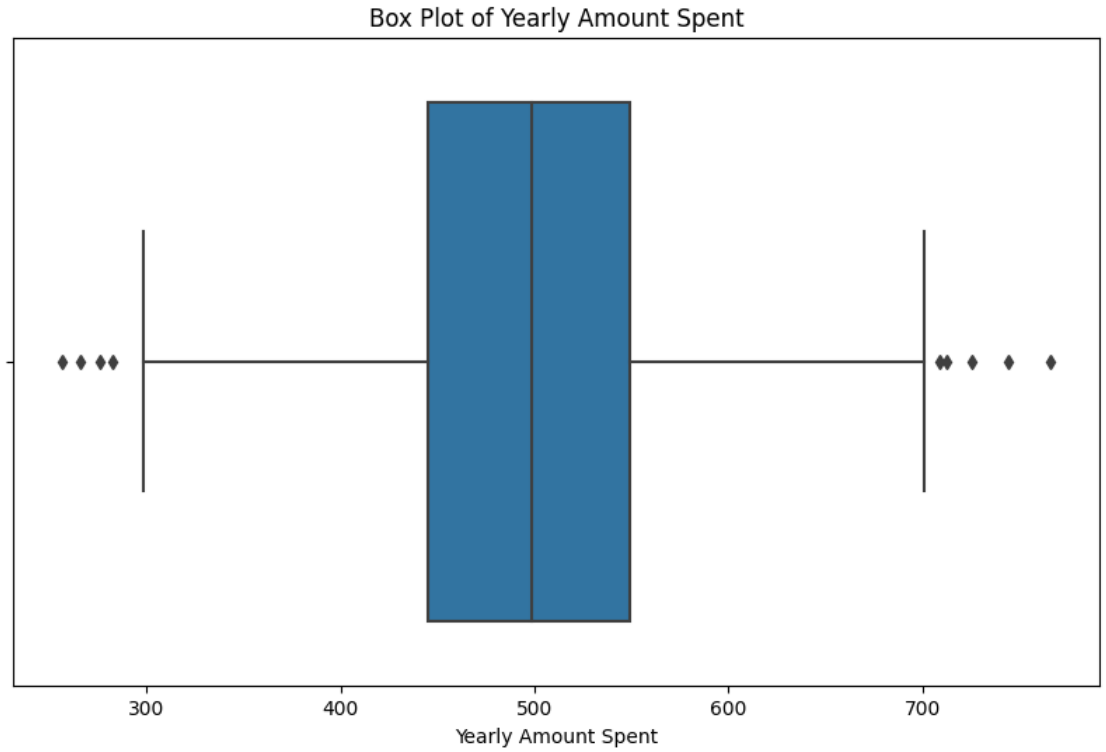


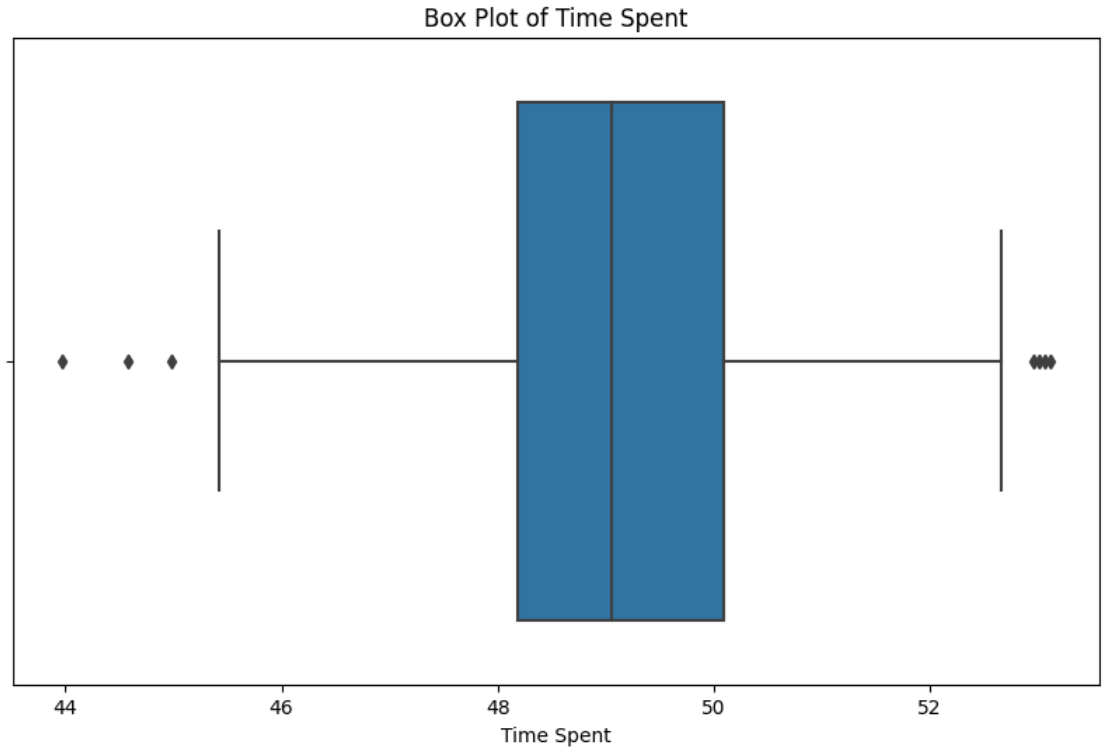


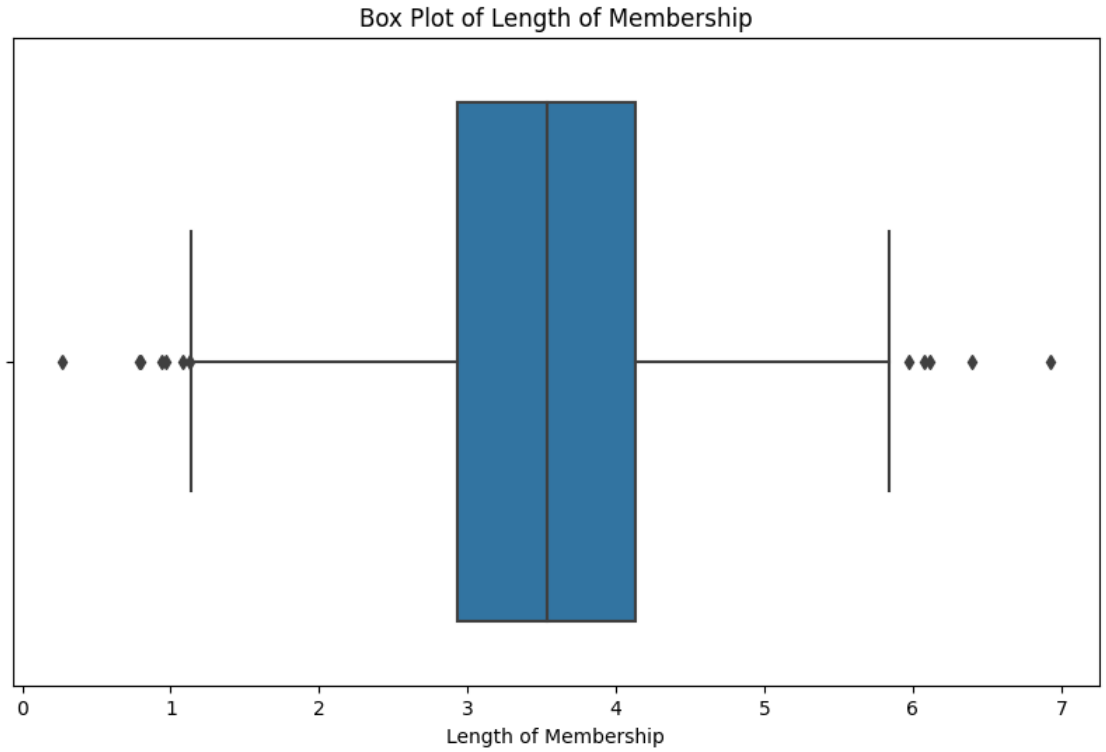


From the histogram above, the yearly amount spent are mostly contributed at 500USD/year which fall at the median of the data with normal distribution.

```
#Box-and-whisker plot is used to provides information about median,
quartiles, and potential outliers in the data distribution.

#Yearly Amount Spent
sns.boxplot(data=df, x='Yearly Amount Spent')
plt.title("Box Plot of Yearly Amount Spent")
plt.show()

#Time Spent
sns.boxplot(data=df, x='Time Spent')
plt.title("Box Plot of Time Spent")
plt.show()

#Length of Membership
sns.boxplot(data=df, x='Length of Membership')
plt.title("Box Plot of Length of Membership")
plt.show()
```


Box Plot of Yearly Amount Spent
300
400
500
600
700
Yearly Amount Spent

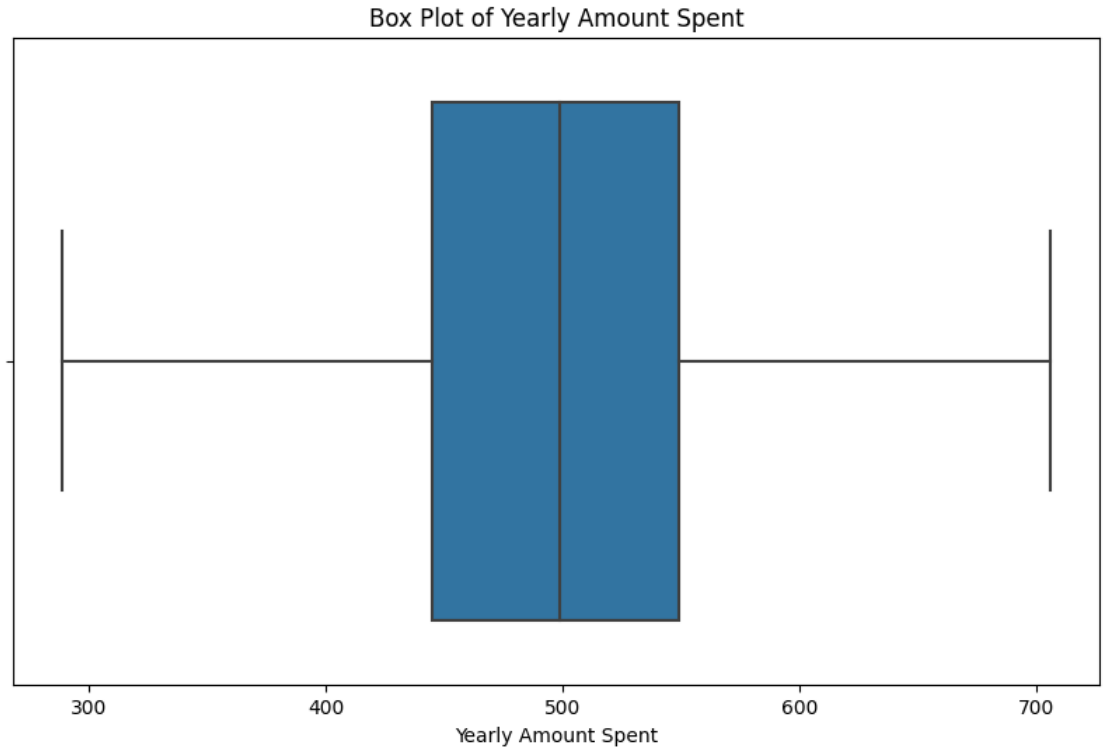

Box Plot of Time Spent
44
46
48
50
52
Time Spent

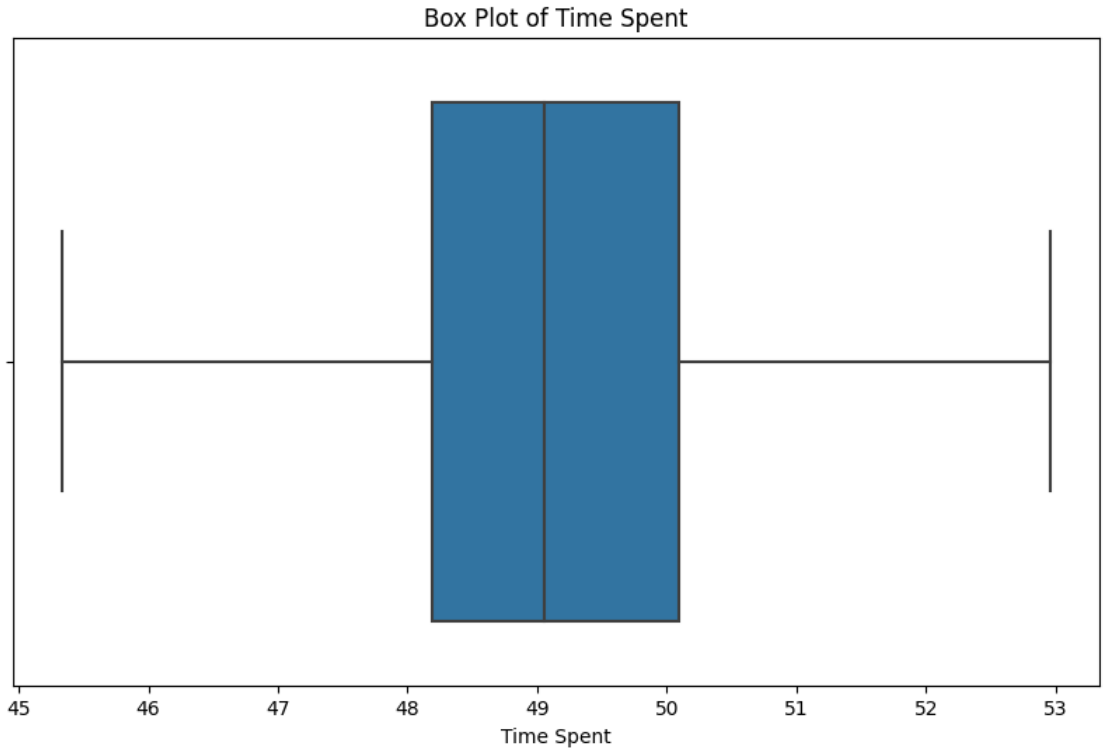


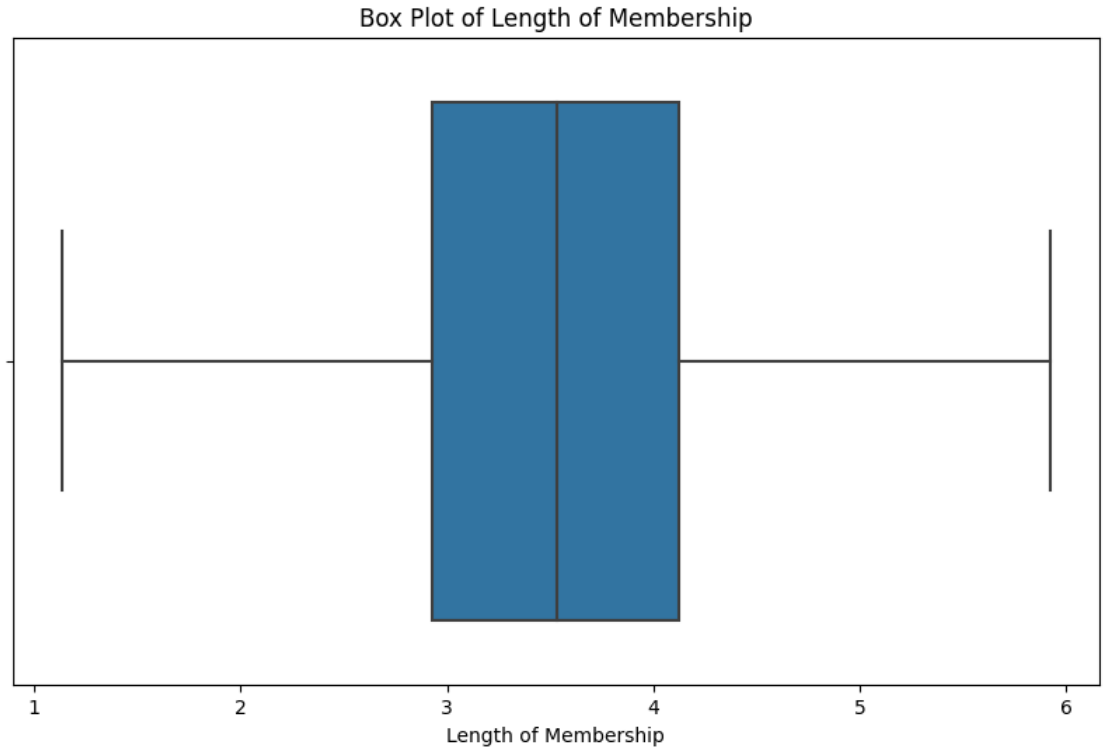
By Box plots above, there are 9 potential outliers in dataset which fall outside the whiskers, represented the values that are abnormally higher or lower as compared to average of the data.

```
def remove_outlier(col):
  sorted(col)
  Q1, Q3 = col.quantile([0.25, 0.75])
  IQR = Q3 - Q1
  lower_range = Q1 - (1.5 * IQR)
  upper_range = Q3 + (1.5 * IQR)
  return lower_range, upper_range

lower_range, upper_range = remove_outlier(df["Yearly Amount Spent"])
df["Yearly Amount Spent"] = np.where(df["Yearly Amount Spent"] > upper_range,
upper_range, df["Yearly Amount Spent"])
df["Yearly Amount Spent"] = np.where(df["Yearly Amount Spent"] < lower_range,
lower_range, df["Yearly Amount Spent"])

lower_range, upper_range = remove_outlier(df["Time Spent"])
df["Time Spent"] = np.where(df["Time Spent"] > upper_range, upper_range,
df["Time Spent"])
df["Time Spent"] = np.where(df["Time Spent"] < lower_range, lower_range,
df["Time Spent"])

lower_range, upper_range = remove_outlier(df["Length of Membership"])
df["Length of Membership"] = np.where(df["Length of Membership"] >
upper_range, upper_range, df["Length of Membership"])
```

```
df["Length of Membership"] = np.where(df["Length of Membership"] <
lower_range, lower_range, df["Length of Membership"])

#Box-and-whisker plot is used to provides information about median,
quartiles, and potential outliers in the data distribution.

#Yearly Amount Spent
sns.boxplot(data=df, x='Yearly Amount Spent')
plt.title("Box Plot of Yearly Amount Spent")
plt.show()

#Time Spent
sns.boxplot(data=df, x='Time Spent')
plt.title("Box Plot of Time Spent")
plt.show()

#Length of Membership
sns.boxplot(data=df, x='Length of Membership')
plt.title("Box Plot of Length of Membership")
plt.show()
```



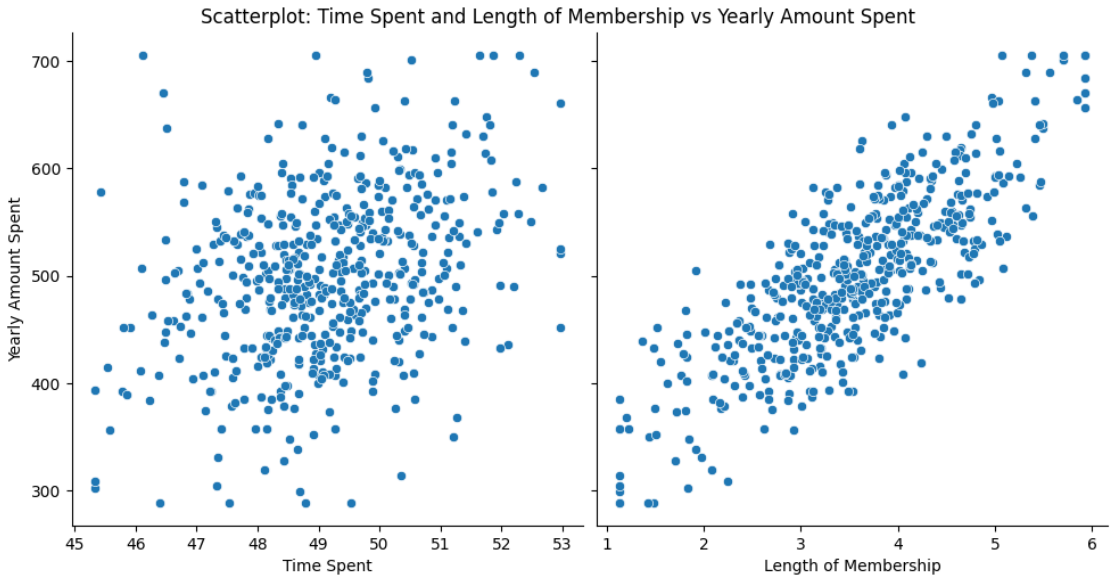




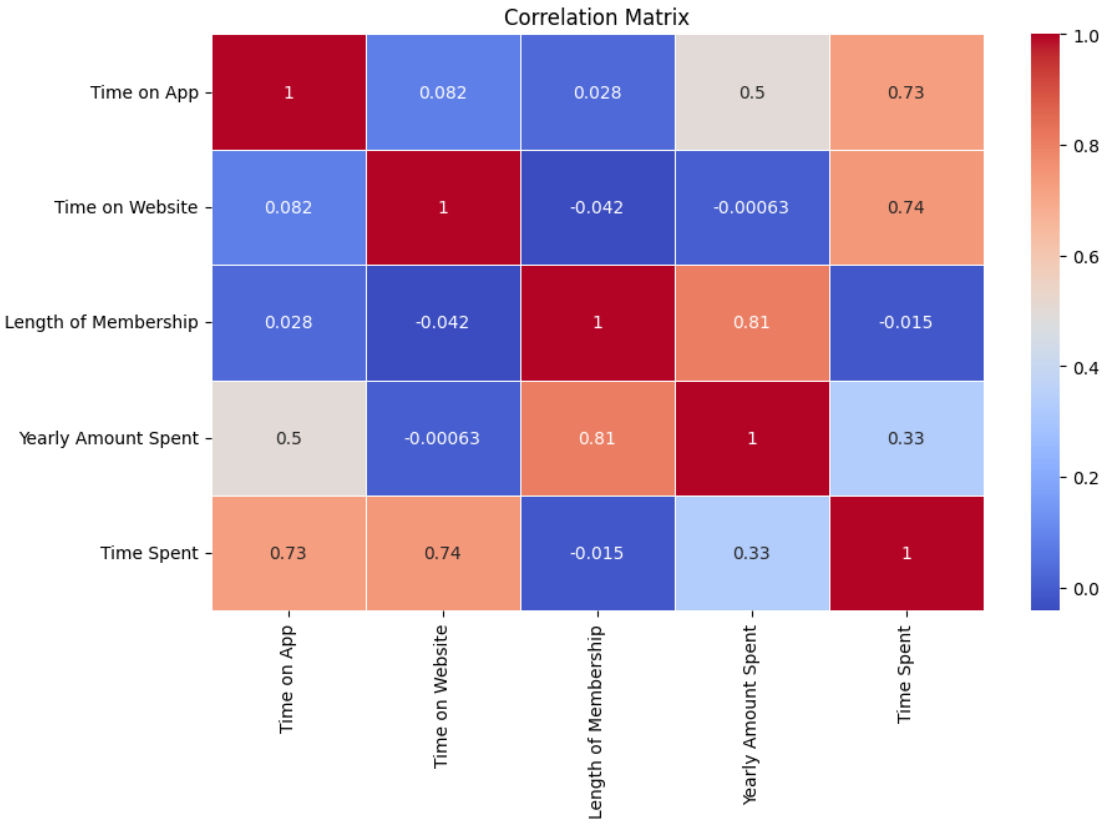
1. Multivariate Analysis

- Involve two or more variables.
- Use to explore the relationships between the variables.

```
sns.pairplot(df, x_vars=['Time Spent', 'Length of Membership'],
y_vars='Yearly Amount Spent', kind='scatter', height=5, aspect=1,
diag_kind='hist')
plt.suptitle('Scatterplot: Time Spent and Length of Membership vs Yearly
Amount Spent', y=1.02)
plt.show()
```




```
#Heatmap with correlation matrix is use to show the relationship between each
variables.
correlation_matrix = df.corr()   #to calculate correlation matrix

#Input Heatmap of correlation matrix with annotations, result will shows in
correlation coefficients.
#0 = no linear correlation, 1 = perfect positive correlation, -1 = perfect
negative correlation.
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=.5)
#annot=true means value will be display on heatmap. 'cmap' means colour
pallete
plt.title('Correlation Matrix')
plt.show()
```



By referring to heatmap, the yearly ammount spent have significant correlation with (from highest to lowest):

1. Length of membership with correlation coefficients of 0.81.
1. Time on App with correlation coefficients of 0.5.
2. Average Session Length with correlation coefficients of 0.36.
3. Time on website with correlation coefficients of -0.0026.

Since the length of membership no correlation with time spent improve web/app won't help on retain them.

Also, most of the current customer are now with 3-4 year membership, it is important also to consider improve in Time Spend beside retain customer's loyalty.

#Data Modelling

Proposal to increase yearly sales:

1. Clustering: To divide the yearly amount spend into few categories to check most of the customers prefer website or app. Since heatmap unable to see the correlation. - behaviour(human)

1. regression: Use to predict if we improve in time spend & length of membership How much will the impact towards sales in % -impact -> predit

G3: length of membership - 7yrs member may spent highest amount Y.A.S total spend of 5-7yrs member = 10,000 USD time spend on app/website= ?

G2: 7yrs members may not be the biggest group in overall member. most of the member having 3-4year membership. assumption: since spend 500USD/year total spend of 3-4yrs member = >10,000USD time spend on app/website=?

G1: less than 3yrs membership ?

Get Categorical Columns

```
cat_cols = df.select_dtypes(exclude=[np.number]).columns.values
print('Categorical cols :',cat_cols)
df.drop(cat_cols,axis=1,inplace=True)
print(df.head(5))
```

```
Categorical cols : []
```

```
   Time on App  Time on Website  Length of Membership  Yearly Amount Spent
Time Spent
0    12.655651        39.577668              4.082621           587.951054
52.233319
1    11.109461        37.268959              2.664034           392.204933
48.378420
2    11.330278        37.110597              4.104543           487.547505
48.440875
3    13.717514        36.721283              3.120179           581.852344
50.438796
4    12.795189        37.536653              4.446308           599.406092
50.331842
```

```
# Get numeric columns
num_cols = df.select_dtypes(include=[np.number]).columns.values
print('Numeric cols :',num_cols)
```

```
Numeric cols : ['Time on App' 'Time on Website' 'Length of Membership'
 'Yearly Amount Spent' 'Time Spent']
```

```
x = df.drop(['Time on App', 'Time on Website', 'Yearly Amount Spent'],
axis=1)
y = df['Yearly Amount Spent']
print(x.head(5))
```

```
   Length of Membership  Time Spent
0              4.082621   52.233319
1              2.664034   48.378420
2              4.104543   48.440875
3              3.120179   50.438796
4              4.446308   50.331842
```

```
print(y.head(5))
```

```
0    587.951054
1    392.204933
2    487.547505
3    581.852344
4    599.406092
Name: Yearly Amount Spent, dtype: float64
```

##Fitting Model

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size= 0.2,
random_state=8)
print(x_train.shape, x_test.shape, y_train.shape, y_test.shape)
```

```
(400, 2)
```

```
(100, 2)
```

```
(400,)
```

```
(100,)
```

```
model = DecisionTreeRegressor(criterion='squared_error', random_state=8)
model.fit(x_train, y_train)
```

```
# Assuming x_test is also a NumPy array or Pandas DataFrame converted to an
array
x_test = x_test.to_numpy() if isinstance(x_test, pd.DataFrame) else x_test
```

```
y_pred = model.predict(x_test)
accuracy_dt = r2_score(y_test, y_pred)
print("Accuracy (R² score) of Decision Tree Regression model:", accuracy_dt)
```

```
Accuracy (R² score) of Decision Tree Regression model: 0.6283621250237703
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X
does not have valid feature names, but DecisionTreeRegressor was fitted with
feature names
  warnings.warn(
```

```
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
print("RMSE:", rmse)
```

```
RMSE: 48.39916727470188
```

```
feature_importances = model.feature_importances_
for feature, importance in zip(x_train.columns, feature_importances):
    print(f'{feature}: {importance}')
```

```
Length of Membership: 0.7854156468048098
Time Spent: 0.21458435319519012
```

```
feature_importance_df = pd.DataFrame({'Feature': x_train.columns,
'Importance': feature_importances})
print(feature_importance_df)
```

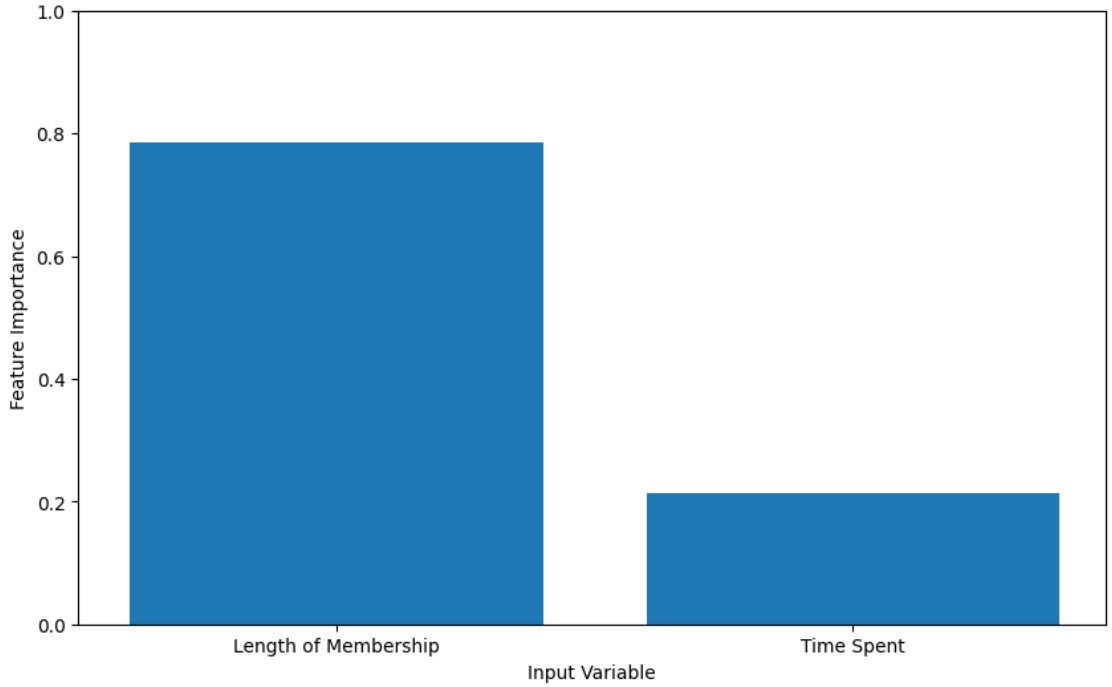
```
                Feature  Importance
0  Length of Membership    0.785416
1            Time Spent    0.214584
```

#Summary of features importances:

With a feature importance of 0.7854, 'length of membership' has the most significant impact on 'yearly amount spent in terms of model calculation.

'Time on app' has a relatively high impact on 'yearly amount spent' with a feature importance of 0.2145 .

```
input_features = x_train.columns

plt.bar(input_features, feature_importances)
plt.ylim(0.0, 1.0)
plt.xlim(-0.5)
plt.xlabel('Input Variable')
plt.ylabel('Feature Importance')
plt.xticks(rotation=0)
plt.show()
```



Modeling (Multiple Linear Regression)

```
num_cols = df.select_dtypes(include=[np.number]).columns.values
print('Numeric column: ', num_cols)
```

```
Numeric column:  ['Time on App' 'Time on Website' 'Length of Membership'
 'Yearly Amount Spent' 'Time Spent']
```

```
X = df.drop(['Time on App', 'Time on Website', 'Yearly Amount Spent'],
axis=1)
Y = df['Yearly Amount Spent']
print(X.head(5))
```

```
   Length of Membership  Time Spent
0              4.082621   52.233319
1              2.664034   48.378420
2              4.104543   48.440875
3              3.120179   50.438796
4              4.446308   50.331842
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
random_state=8)
```

```
model = LinearRegression()
model.fit(X_train, Y_train)
```

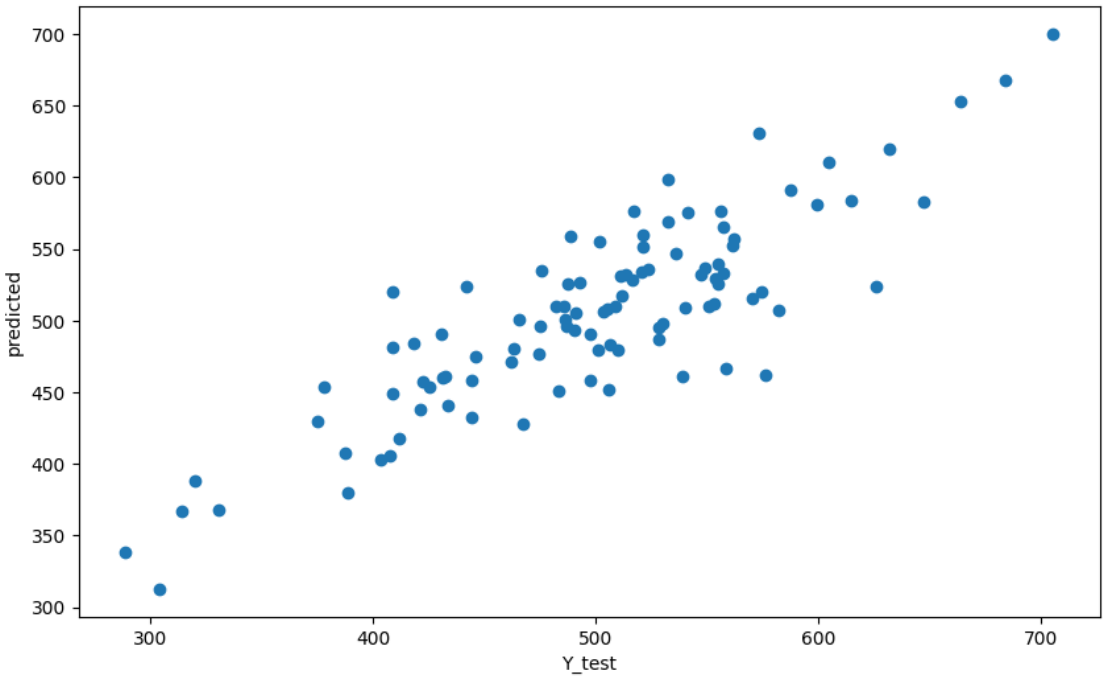
```
LinearRegression()
```

```
Y_pred = model.predict(X_test)
accuracy = r2_score(Y_test, Y_pred)
print(f"Accuracy (R² score) of the model: {accuracy}")
```

```
Accuracy (R² score) of the model: 0.7288189214032095
```

```
predicted = model.predict(X_test)
plt.scatter(Y_test, predicted)
plt.xlabel('Y_test')
plt.ylabel('predicted')
```

```
Text(0, 0.5, 'predicted')
```




```
# Evaluation of Regression model
print('RMSE:', np.sqrt(metrics.mean_squared_error(Y_test, predicted)))
```

```
RMSE: 41.343534042210045
```

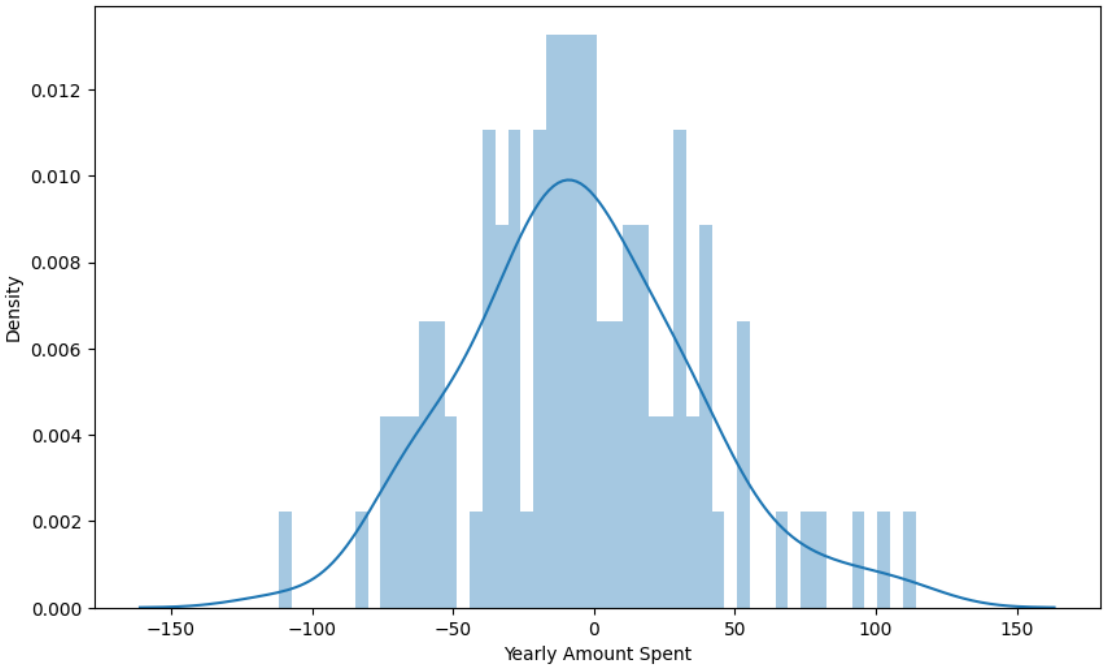
```
# The coefficients of the model
print('Coefficients: ', model.coef_)
```

```
Coefficients:  [64.9772324  17.82634125]
```

```
# Residuals
# Plotting a histogram of the residuals and it looks normally distributed
sns.distplot((Y_test-predicted),bins=50)
```

```
<ipython-input-59-1647676772b6>:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot((Y_test-predicted),bins=50)
```

```
<Axes: xlabel='Yearly Amount Spent', ylabel='Density'>
```



Conclusion

```
coefficients = pd.DataFrame(model.coef_,X.columns)
coefficients.columns = ['Coeffecient']
print(coefficients)
```

```
                     Coeffecient
Length of Membership    64.977232
Time Spent              17.826341
```

Holding all other features fixed, a 1 unit increase in Length of Membership is associated with an increase of 64.98 total dollars spent.

Holding all other features fixed, a 1 unit increase in Time Spent is associated with an increase of 17.83 total dollars spent.