A Project Report

On

Customer Retention

Submitted by -

Internship Batch 7

Under the guidance of -

Mr. Mario Thokchom

Project Mentor

Department of DADS

Skill Lync

SKILL LYNC

Table of Contents

Abstract

This report presents an Exploratory Data Analysis (EDA) conducted on customer retention in an e-commerce setting. The study aims to gain insights into customer behaviour, identify factors influencing retention rates, and recommend strategies to enhance customer loyalty. The dataset used in this analysis comprises historical transactional data, customer demographics, and activity logs collected over a specified period. Through visualizations and statistical analysis, we explore patterns and trends related to customer retention, enabling e-commerce businesses to make data-driven decisions for sustainable growth and improved customer engagement.

1. Introduction

In the highly competitive world of e-commerce, attracting new customers is undoubtedly crucial for business growth and expansion. However, equally important, if not more so, is the ability to retain existing customers. Customer retention, the process of engaging and maintaining a loyal customer base, plays a pivotal role in the long-term success of an e-commerce venture. In this report, we will delve into the significance of customer retention for the e-commerce category and explore various strategies that businesses can employ to foster customer loyalty.

In recent years, the e-commerce landscape has witnessed unprecedented growth, with consumers increasingly shifting towards online shopping due to its convenience, wide product selection, and competitive pricing. However, this surge in popularity has led to cutthroat competition among e-commerce businesses, making customer retention more challenging than ever before. In response to these challenges, e-commerce companies must recognize that retaining existing customers is a cost-effective and sustainable way to drive revenue and profitability.

Studies have shown that acquiring a new customer can cost up to five times more than retaining an existing one. Furthermore, loyal customers tend to spend more and are more likely to refer others to the e-commerce platform, acting as brand ambassadors and amplifying the customer base through positive word-of-mouth. Consequently, businesses that

focus on customer retention not only boost their bottom line but also enhance brand reputation and credibility.

In this report, we will explore various customer retention strategies that have proven to be successful in the e-commerce domain. These strategies include personalized marketing, exceptional customer service, loyalty programs, post-purchase engagement, and the effective use of data analytics to understand customer behaviour and preferences better. By implementing these tactics, e-commerce companies can create meaningful connections with their customers, fostering long-term relationships and customer loyalty.

The report will also examine the role of technology in customer retention for e-commerce businesses. From sophisticated customer relationship management (CRM) systems to AI-powered recommendation engines, technology can significantly aid in understanding customer needs, predicting behaviour, and tailoring personalized experiences. Additionally, we will discuss the potential challenges and obstacles faced by e-commerce companies when it comes to customer retention and propose solutions to overcome them.

Ultimately, the goal of this report is to equip e-commerce businesses with a comprehensive understanding of the importance of customer retention and provide actionable insights into developing effective strategies for fostering customer loyalty. By embracing customer-centric approaches and leveraging cutting-edge technology, e-commerce companies can not only survive in a highly competitive landscape but also thrive, building a loyal customer base that sustains long-term success.

1.1 Problem Statement:

In the rapidly expanding and fiercely competitive realm of e-commerce, customer retention poses a significant challenge for businesses. While attracting new customers is essential for growth, retaining existing ones is equally critical for sustained success and profitability. However, the high churn rates and diminishing customer loyalty within the e-commerce category underscore the need for a comprehensive understanding of the factors influencing customer retention and the development of effective strategies to address this issue.

The problem at hand is the declining customer retention rates experienced by e-commerce businesses, which hinders their ability to establish long-term relationships with customers and capitalize on their potential lifetime value. As the cost of customer acquisition continues to

rise, it becomes imperative for e-commerce companies to focus on retaining their existing customer base to drive revenue growth, reduce marketing expenses, and foster brand loyalty.

This report aims to identify the underlying causes of customer churn in the e-commerce domain and explore the most promising customer retention strategies that have proven to be successful in this industry. By investigating the challenges faced by e-commerce companies in retaining customers and examining the role of technology in facilitating customer retention efforts, this study seeks to equip businesses with actionable insights and recommendations to strengthen their customer retention initiatives.

In conclusion, the problem statement revolves around the pressing need for e-commerce businesses to address customer retention challenges and develop effective strategies to cultivate loyal customer relationships, ultimately ensuring sustainable growth and competitiveness in the dynamic e-commerce landscape.

1.2 Purpose:

The purpose of this report is to provide a comprehensive analysis of customer retention in the e-commerce category and its significance for businesses operating in this competitive industry. The report aims to shed light on the importance of retaining existing customers as a cost-effective and sustainable approach to driving revenue growth and enhancing brand reputation.

Through this study, we seek to explore various customer retention strategies that have proven to be successful in the e-commerce domain. By examining the role of personalized marketing, exceptional customer service, loyalty programs, post-purchase engagement, and data analytics, we aim to equip e-commerce businesses with actionable insights to develop effective customer-centric approaches.

Furthermore, the purpose of this report is to highlight the crucial role of technology in enhancing customer retention efforts for e-commerce companies. We will explore how technology, including customer relationship management (CRM) systems and AI-powered recommendation engines, can enable businesses to understand customer behaviour and preferences better, leading to more personalized and engaging experiences.

The report also serves to identify potential challenges and obstacles that e-commerce companies may face when implementing customer retention strategies. By understanding these challenges, businesses can proactively develop solutions to overcome them and optimize their customer retention initiatives.

Ultimately, the purpose of this report is to serve as a valuable resource for e-commerce businesses, offering insights, data, and best practices that will enable them to create meaningful connections with customers, foster brand loyalty, and secure long-term success in the ever-evolving e-commerce landscape**.**

2. Methodology:

To achieve the objectives of this report and gain a comprehensive understanding of customer retention in the e-commerce category, a multi-faceted methodology will be employed. The methodology will involve the following key

a) **Data Collection**: Collect both qualitative and quantitative data from reputable sources and case studies. This data may include purchase behaviour, churn, preferred order category, and payment mode that have excelled in customer retention.

b) **Data Analysis:** Perform an initial analysis of the gathered data to gain insights into its structure and characteristics. This step involves understanding the variables, their distributions, and potential patterns or trends.

c) **Data Cleansing:** Cleanse the data by handling missing values, removing duplicates, and addressing any inconsistencies or errors. This ensures the data is reliable and suitable for further analysis.

d) **Data Formatting:** Format the data in a consistent and standardized manner to facilitate efficient processing and analysis. This may involve converting data types, normalizing values, and structuring the dataset for modelling.

e) **Exploratory Data Analysis (EDA):** Conduct EDA to explore relationships between variables, identify patterns, and gain deeper insights into the data. This step involves visualizations, statistical analyses, and data summarization techniques.

3. Exploratory Data Analysis (EDA)

3.1 Data Cleaning:

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct.

For all data cleaning operations, we used Python language in jupyter notebook.

Following steps, we performed to clean the dataset:

We use **panda's library** to for analysis.

3.2 Handling Missing Values:

To prevent biased or incorrect analyses, maintain data integrity, avoid errors, ensure compatibility with machine learning algorithms and capture missing data patterns we need to handle null values from dataset.

Step 1) Checked the null values in each column using .info () method which gives overview of dataframe df. From that we got total number of null values on each column.

import pandas as pd

df.info ()

Step 2) Dropped null values for some fields and updated with zero using pandas.

```
df.dropna(subset=["WarehouseToHome"],axis=0,inplace=True)
df.dropna(subset=["OrderAmountHikeFromlastYear"],axis=0,inplace=True)
df.dropna(subset=["OrderCount"],axis=0,inplace=True)
df.fillna(0,inplace=True)
```

Step 3) Checked the null values again using isnull () function.

df.isnull().sum()

3.3 Removing Duplicates:

Duplicate entries are problematic for multiple reasons. An entry appearing more than once receives disproportionate weight during training. Models that succeed on frequent entries only look like they perform well. Duplicate entries can ruin the split between train, validation and test sets where identical entries are not all in the same set. This can lead to biased performance estimates that result in disappointing the model in production.

We used drop_duplicates method to remove the duplicates in our dataset.

- df.drop_duplicates(ignore_index=True,inplace=True)

3.4 Change column format, order and datatype:

To change the column format, order, and data types in a Dataframe using pandas, we use various methods and functions provided by the library.

- To change column datatype, we used the **astype()** method. This allows to convert the values of a column to a specified data type or format.

  df["Product Price"]=df["Product Price"].astype("float64")

- To rename and replace columns we used **.rename** function and **.replace** function.

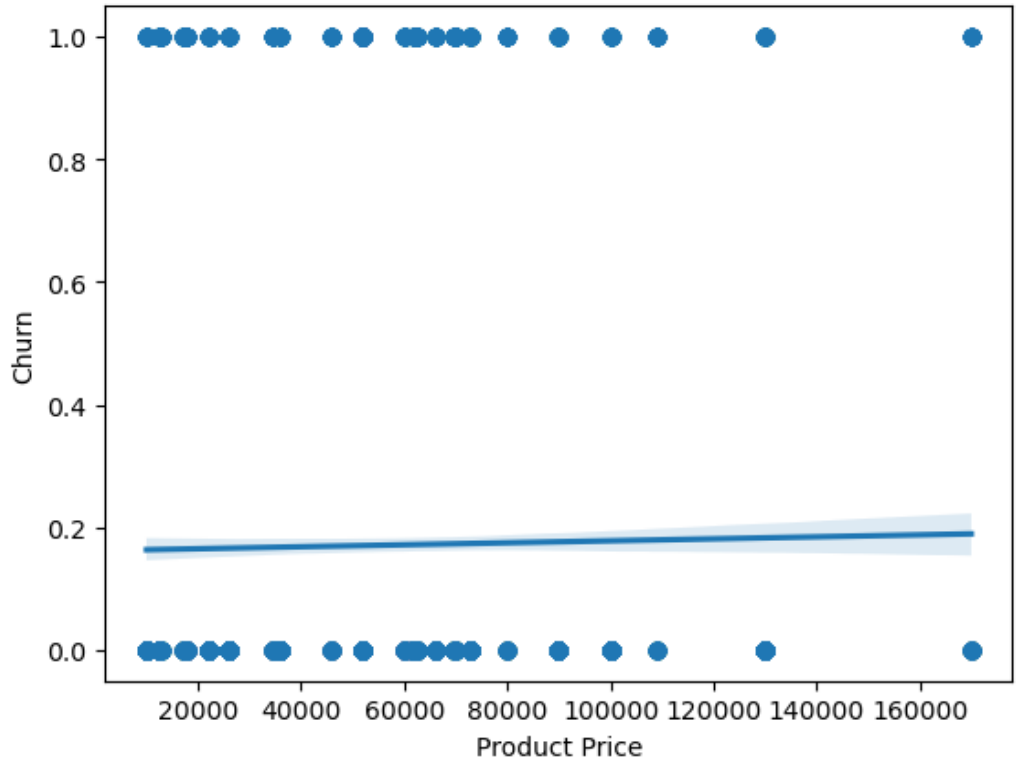
```
df["PreferredPaymentMode"]=df["PreferredPaymentMode"].replace("COD","Cash on Delivery")
df["PreferredPaymentMode"]=df["PreferredPaymentMode"].replace("CC","Credit Card")
df["PreferredLoginDevice"]=df["PreferredLoginDevice"].replace("Phone","Mobile Phone")
```

Exploratory Data Analysis

```
def reg(Predictor,Target="Churn",dataframe=df):
    sns.regplot(x=Predictor,y=Target,data=dataframe)
```

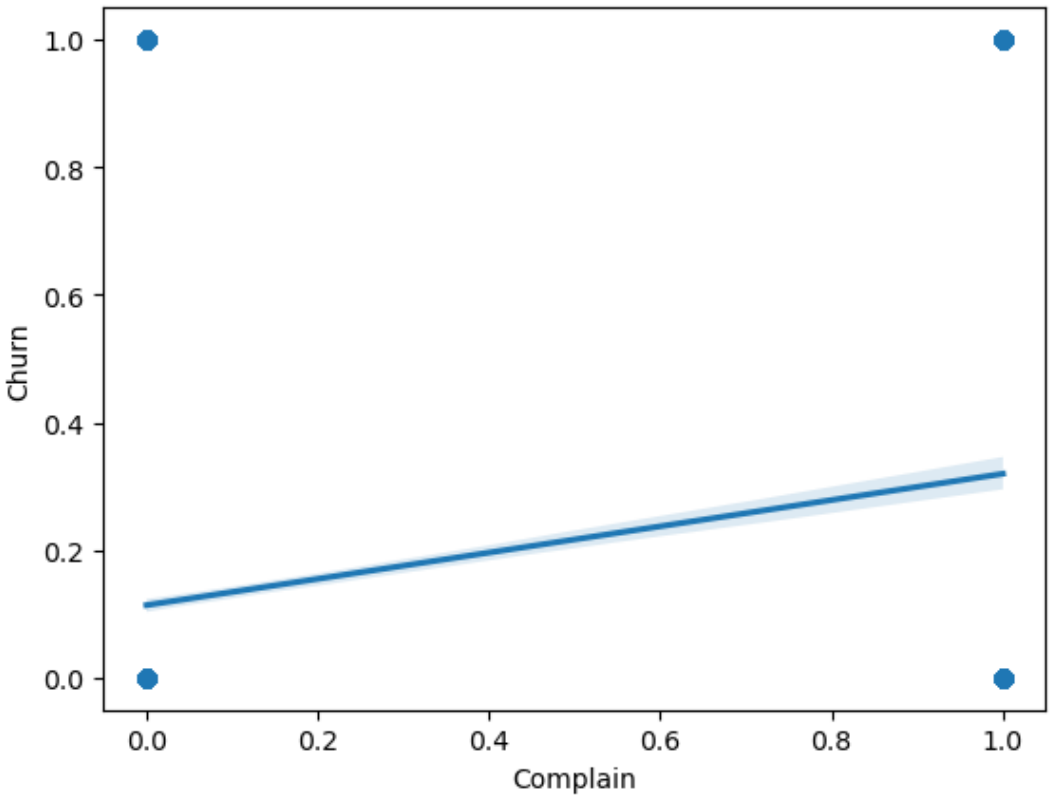
```
reg("Product Price")
```

Weak correlation



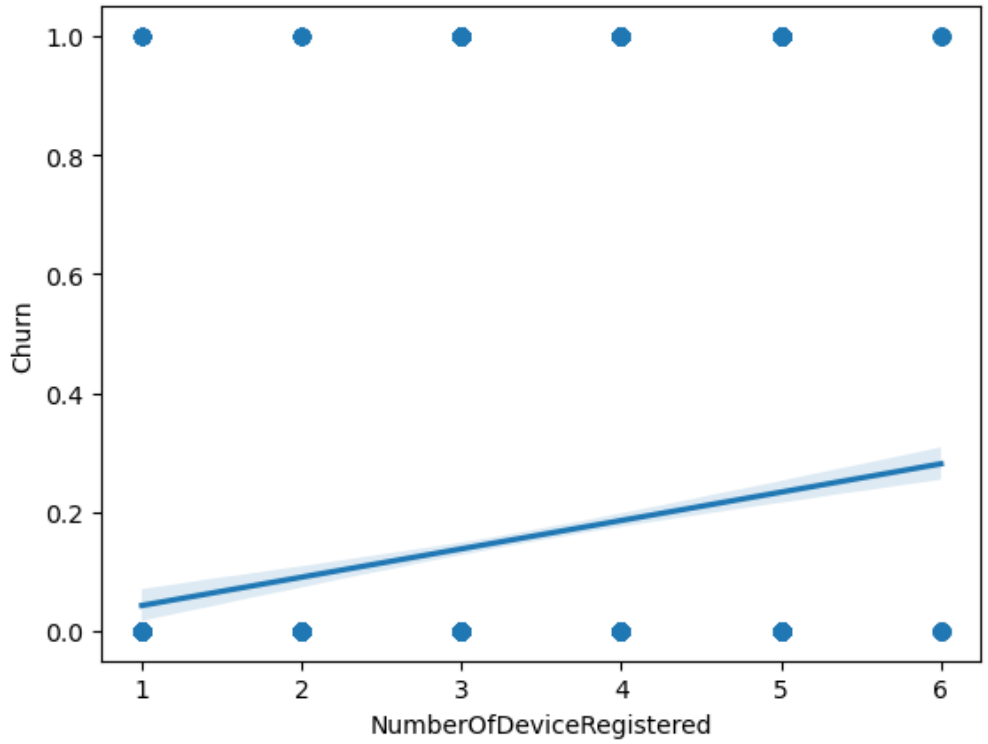

```
reg("Complain")
```

Positive Linear Relationship



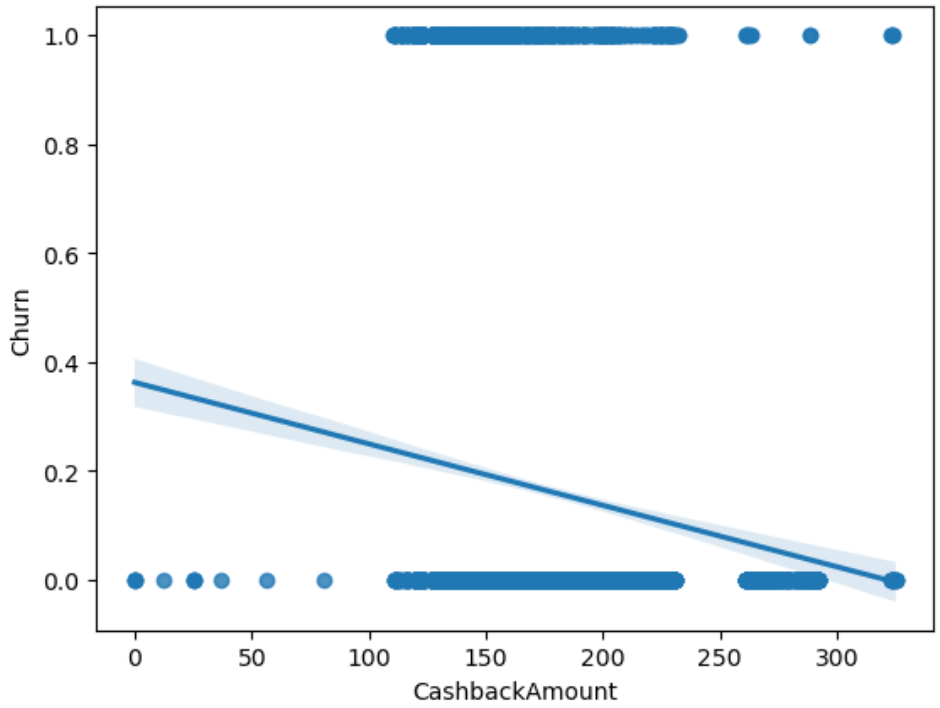

```
reg("NumberOfDeviceRegistered")
```

Positive Linear Relationship




```
reg("CashbackAmount")
```

Negative Linear Relationship



4. Processing Steps:

To accomplish the task, the following steps can be followed:

1. Load the dataset: Read the E commerce dataset into a data structure that allows for easy manipulation and analysis. This can be done using libraries such as Pandas in Python.
2. Iterate through the dataset: Iterate over each row in the dataset and compare the churn the categorization criteria
3. Save the modified dataset: Save the updated dataset with the new column included, preserving the original data and the newly added categorization.

By following the outlined steps, the dataset can be processed for further analysis. This categorization will provide an additional insight into the performance and allow for further analysis and exploration of factors.

After doing the process of data cleaning we have **5630 rows × 25 columns in datasets**.

Customer retention

It refers to the ability of a business to retain its existing customers over a period of time. It involves building long-term relationships with customers by providing high-quality products or services, excellent customer service, and personalized experiences that meet their needs and expectations. Customer retention is important for businesses because it can lead to increased customer loyalty, repeat purchases, positive word-of-mouth referrals, and ultimately, sustained revenue growth.

Customer churn

On the other hand, Customer churn refers to the rate at which customers discontinue using a product or service. It reflects the number of customers who cancel their subscriptions, switch to a competitor, or simply stop using a product or service over a given time period. High churn rates can be a cause for concern for businesses because they indicate a lack of customer loyalty or satisfaction, and can lead to a decline in revenue and market share. Therefore, reducing churn and retaining more customers is a key objective for businesses that want to achieve sustainable growth.

Calculating churn depends on the nature of the business and the available data. There are generally two types of churns: customer churn and revenue churn.

1. Customer Churn:
Customer churn measures the percentage of customers who have stopped using a product or service over a specific period. To calculate customer churn, you'll need the following data:

- Number of customers at the beginning of the period (C_start)
- Number of customers at the end of the period (C_end)
- Number of new customers acquired during the period (C_new)

The formula for calculating customer churn is as follows:

Customer Churn Rate = ((C_start - C_end + C_new) / C_start) * 100

For example, if you had 1000 customers at the beginning of the month, 900 customers at the end of the month, and acquired 200 new customers during the month, the customer churn rate would be:

Customer Churn Rate = ((1000 - 900 + 200) / 1000) * 100 = (300 / 1000) * 100 = 30%

2. Revenue Churn:
Revenue churn measures the percentage of revenue lost due to customer churn during a specific period. To calculate revenue churn, you'll need the following data:

- Total revenue from all customers at the beginning of the period (R_start)
- Total revenue from all customers at the end of the period (R_end)
- Revenue from new customers acquired during the period (R_new)

The formula for calculating revenue churn is as follows:

Revenue Churn Rate = ((R_start - R_end + R_new) / R_start) * 100

For example, if you had $50,000 in revenue from all customers at the beginning of the month, $45,000 in revenue at the end of the month, and generated $10,000 in revenue from new customers during the month, the revenue churn rate would be:

Revenue Churn Rate = (($50,000 - $45,000 + $10,000) / $50,000) * 100 = ($15,000 / $50,000) * 100 = 30%

Keep in mind that churn calculation may vary depending on the specific business and its data. Additionally, other factors like customer acquisition rate, customer lifetime value, and overall growth should be considered in conjunction with churn rates to gain a comprehensive understanding of the business's health.

5. Python Language:

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems.

Python is an interpreter, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high-level dynamic data types, and classes. It supports multiple programming paradigms beyond object-oriented programming, such as procedural and functional programming.

Python has a simple syntax similar to the English language. Python has syntax that allows developers to write programs with fewer lines than some other programming languages. Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

***The following are some of the features in Python that are discussed below:**

- Easy to Code. Python is a very high-level programming language, yet it is effortless to learn.
- Easy to Read.
- Free and Open-Source.
- Robust Standard Library.
- Interpreted.
- Portable.
- Object-Oriented and Procedure-Oriented.
- Extensible.

5.1 Advantages of Using Python:

1) Independence across platforms

Due to its ability to run on multiple platforms without the need to change, developers prefer Python, unlike in other programming languages. Python runs across different platforms, such as Windows, Linux, and macOS, thus requiring little or no changes.

2) Consistency and simplicity

The Python programming language is a haven for most software developers looking for simplicity and consistency in their work. The Python code is concise and readable, which simplifies the presentation process.

3) Frameworks and libraries variety

Libraries and frameworks are vital in the preparation of a suitable programming environment. Python frameworks and libraries offer a reliable environment that reduces software development time significantly.

5.2 Why Python is Most Suitable for Data Analysis:

As of my last update in September 2021, Python remains one of the most popular and widely used programming languages for data analysis, and several factors contribute to its suitability for this purpose:

1. **Ease of Learning and Use**: Python is known for its simplicity and readability. Its syntax is clear and resembles the English language, making it accessible even to those new to programming. This ease of learning allows data analysts to quickly grasp the language and start using it for data analysis tasks.

2. **Rich Ecosystem of Libraries**: Python has a vast ecosystem of powerful libraries and frameworks that are specifically designed for data analysis. Some of the most popular ones are:

- NumPy: A fundamental package for scientific computing with support for large, multi-dimensional arrays and matrices.

- Pandas: Offers high-performance data manipulation and analysis tools, particularly with structured data.
- Matplotlib and Seaborn: Used for data visualization to create insightful plots and charts.
- SciPy: Provides additional scientific computing functions built on top of NumPy.
- Scikit-learn: A machine learning library that is widely used for various predictive data analysis tasks.

3. **Flexibility and Versatility**: Python is a general-purpose programming language, which means it can be used for various purposes beyond data analysis. This versatility is valuable as data analysis often involves integrating different tasks, such as data cleaning, visualization, and machine learning, which can all be performed within the same language.

4. **Active Community and Support**: Python has a large and active community of developers and data analysts. This means there are extensive online resources, tutorials, and forums where beginners can seek help and experts can share their knowledge. The Python community is also committed to continually improving the language and its data-related libraries.

5. **Interoperability**: Python plays well with other languages and tools. Data analysts can easily integrate Python with databases, web frameworks, big data tools like Apache Hadoop, and cloud services, allowing seamless data flow across various components of a data analysis pipeline.

6. **Employment Market**: Python's popularity in data analysis and other fields has led to a robust job market for Python developers with data analysis skills. Many organizations use Python for data analysis, making it a valuable skill for those pursuing a career in the data science domain.

7. **Jupyter Notebooks**: Jupyter Notebooks provide an interactive computing environment that allows data analysts to create and share documents containing live code, equations, visualizations, and narrative text. They have become an integral part of data analysis workflows, and Python is one of the main programming languages used in Jupyter Notebooks.

While Python is highly suitable for data analysis, it's essential to note that other programming languages, such as R, are also popular choices, depending on the specific needs and preferences of the data analysis tasks at hand. The choice of language may vary based on factors like data size, complexity of analysis, team expertise, and the organization's existing technology stack.

We use **Anaconda Navigator as an Integrated Development Environment (IDE)** which is a software application that helps programmers develop software code efficiently.

***Python Version**: 3.9.13 [MSC v.1916 64 bit (AMD64)]

***Python system requirements:**

- **Processors:** Minimum Intel Atom processor or Intel Core i3 processor
- **Disk Space:** Min 1 GB
- **Operating System:** Windows 7 or later, macOS and Linux
- **Python versions:** Min Python 2.7.X, 3.6.X

6. Anaconda Navigator:

It is a desktop graphical user interface (GUI) includes in Anaconda Distribution that allows you to launch applications and manage conda packages, environments and channels without using command line interface commands. It is available for Windows, macOS and Linux.

Anaconda Navigator has many applications such as Jupyter Notebook, JupyterLab, DataSpell, Datalore, Spyder, VS code, Orange 3, Qt Console, RStudio, and IBM Watson Studio Cloud.

We are using **Anaconda Navigator Version 2.3.1**

***Anaconda Navigator System requirements:**

Anaconda Navigator supports the same operating systems that the Anaconda Distribution supports.

- Windows 10 X 86_64 or newer
- macOS 10.14+, 64-bit

- Ubuntu 14+/Centos+, 64-bit

7. Jupyter Notebook:

We are using Jupyter Notebook for writing all python programs. The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience. Jupyter supports over 40 programming languages, including Python, R, Julia, and Scala.

Notebooks can be shared with others using email, Dropbox, GitHub and the Jupyter notebook viewer. Written code in jupyter notebook can produce rich, interactive output such as HTML, images, videos, LaTeX, and custom Multipurpose Internet Mail Extensions types. It can leverage big data tools such as Apache Spark, from Python, R, and Scala. We can also explore that same data with pandas, scikit-learn, ggplot2, and TensorFlow.

We are using **Jupyter Notebook version 6.4.12**

Jupyter Notebook System requirements:

- **OS**: Windows 10 X 86_64 or newer, macOS 10.14+, 64-bit, Ubuntu 14+/Centos+, 64-bit
- **Python**: Compatible with Python 3.3 and above
- **Memory (RAM)**: At least 2 GB of RAM available
- **Web Browser**: Compatible with Google Chrome, Mozilla Firefox and Safari

8. Python Libraries used

8.1. Pandas' library:

Pandas is a Python library used for working with data sets. It has functions for analysing, cleaning, exploring, and manipulating data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008. Data Frame and Series are part of Pandas.

***Key Features of Pandas:**

- Quick and efficient data manipulation and analysis.
- Tools for loading data from different file formats into in-memory data objects.
- Label-based Slicing, Indexing, and Sub setting can be performed on large datasets.
- Merges and joins two datasets easily.
- Pivoting and reshaping data sets
- Best library for data cleaning operations

***Benefits of Pandas:**

- Easy handling of missing data (represented as NaN) in both floating point and non-floating-point data.
- Size mutability: columns can be inserted and deleted from Dataframe and higher-dimensional objects.

***** We are using **Pandas Library version 1.4.4**

**** pip instal pandas:*** To install pandas by running this command in Anaconda Prompt

**** Import pandas as pd:*** Syntax to import pandas' library in jupyter notebook

8.2 NumPy Library:

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open-source project and you can use it freely.

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python.

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data.

***Arrays in NumPy:**

NumPy main object is the homogeneous multidimensional array.

- It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers.
- In NumPy, dimensions are called axes. The number of axes is rank.
- NumPy's array class is called **ndarray**. It is also known by the alias **array**.

***Benefit of NumPy library in Python:**

NumPy arrays are faster and more compact than Python lists. An array consumes less memory and is convenient to use. NumPy uses much less memory to store data and it provides a mechanism of specifying the data types. This allows the code to be optimized even further.

***Why we use NumPy library in machine learning:**

NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning

* We are using **NumPy Library version 1.21.5**

*** pip install numpy :** To install NumPy by running this command in Anaconda Prompt

***Import numpy as np** : Syntax for importing NumPy library in jupyter notebook

8.3. Matplotlib Library:

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Matplotlib is a python library used to create 2D and 3D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc.

***Benefits of Matplotlib**:

- Create publication quality plots
- Make interactive figures that can zoom, pan, and update.
- Customize visual style and layout.
- Export to many files' formats
- Embed in JupyterLab and Graphical User Interface.
- Use a rich array of third-party packages built on Matplotlib.

***Advantages of Matplotlib:**

- Matplotlib provides a simple way to access large amounts of data.
- It is flexible and supports various forms of data representation.
- It is easy to navigate.
- It ensures accessibility by providing high-quality images.
- It is a powerful tool with numerous applications.

***Application of Matplotlib:**

- Matplotlib has two major application interfaces, or styles of using the library:
- An explicit "Axes" interface that uses methods on a Figure or Axes object to create other Artists, and build a visualization step by step. This has also been called an "object-oriented" interface.
- An implicit "pyplot" interface that keeps track of the last Figure and Axes created, and adds Artists to the object it thinks the user wants.

*** import matplotlib.pylot as plt *:*** To import Matplotlib this line of code is used in python where 'plt' alias is used for 'pylot', a sub module of 'Matplotlib' that provides a MATLAB-like interface for creating plots and visualizations.
* We used Matplotlib **Library version 3.5.**

9. Data Analysis Types:

1. Descriptive Data Analysis:

As described earlier, descriptive data analysis involves the exploration and summary of data to understand its main features and characteristics. It includes calculating measures of central tendency and variability, creating graphical representations, and providing an overview of the dataset. Descriptive analysis helps researchers and analysts gain initial insights into the data and identify patterns, trends, and potential outliers.

2. Inferential Data Analysis:

Inferential data analysis goes beyond the observed dataset and uses statistical techniques to make inferences and draw conclusions about a larger population. It involves using sample data to make predictions or test hypotheses about the entire population from which the sample was drawn. Inferential analysis relies on probability theory and statistical methods to determine the level of confidence in the generalizations made from the sample data.

3**. Exploratory Data Analysis (EDA):**

Exploratory data analysis is a data analysis approach that focuses on visually exploring and summarizing the data to discover patterns, relationships, and potential insights. EDA involves using various data visualization techniques to gain an initial understanding of the data's distribution, identify outliers, and uncover hidden patterns or trends. EDA is often used as a preliminary step before more formal analysis and hypothesis testing.

4. Predictive Data Analysis:

Predictive data analysis involves using historical data and statistical models to make predictions or forecasts about future outcomes or events. It leverages machine learning algorithms and statistical techniques to identify patterns and relationships in the data and develop predictive models. Predictive data analysis is commonly used in

areas such as business forecasting, stock market prediction, weather forecasting, and customer behaviour prediction.

Each type of data analysis serves a specific purpose and can be applied in various fields and industries to gain valuable insights, make informed decisions, and drive evidence-based strategies. The choice of data analysis type depends on the research questions, objectives, and the nature of the data being analysed.

Descriptive Statistical Analysis

Descriptive statistical analysis is a fundamental method used to summarize and describe the main features of a dataset. It provides a concise and meaningful representation of the data, enabling researchers and analysts to gain insights and draw initial conclusions. Descriptive statistics is typically the first step in data analysis and is often used to explore the basic characteristics, patterns, and trends within a dataset. The main techniques used in descriptive statistical analysis include measures of central tendency, measures of variability, and graphical representations.

1. Measures of Central Tendency: These statistics provide information about the center or average of a dataset. The most common measures of central tendency include:

a. Mean: The arithmetic average of all values in the dataset, calculated by summing all values and dividing by the number of data points.

b. Median: The middle value of the dataset when it is arranged in ascending or descending order. It is less affected by extreme values and is a robust measure of central tendency.

c. Mode: The value that occurs most frequently in the dataset. A dataset may have one mode (unimodal) or multiple modes (multimodal).

2. Measures of Variability: These statistics indicate the spread or dispersion of data points around the central tendency. Common measures of variability include:

a. Range: The difference between the maximum and minimum values in the dataset, providing a basic idea of the data's spread.

b. Variance: The average of the squared differences between each data point and the mean. It quantifies the degree of dispersion in the data.

c. Standard Deviation: The square root of the variance, representing the average distance between each data point and the mean. It provides a more interpretable measure of variability.

3. Graphical Representations: Descriptive statistical analysis often involves visualizing the data through various graphical representations to gain a better understanding of the dataset's distribution and patterns. Common graphical representations include:

a. Histograms: A graphical representation of the frequency distribution of data, where data is grouped into bins or intervals.

b. Boxplots: A visual representation of the five-number summary (minimum, first quartile, median, third quartile, and maximum), showing the distribution's spread and outliers.

c. Scatter Plots: Used to explore the relationship between two continuous variables, with data points plotted on a Cartesian plane.

d. Bar Charts: Used for categorical data, displaying the frequency or proportion of each category.

e. Pie Charts: A circular chart representing the proportion of each category in a dataset.

Descriptive statistical analysis provides an initial overview of the dataset, aiding in the identification of potential patterns, outliers, and data quality issues. It helps researchers and analysts to make informed decisions on the appropriate next steps in their data analysis process, such as inferential statistical analysis or hypothesis testing

10. Pearson Method-Numerical Column

The Pearson correlation coefficient, often denoted as "r," is a statistical measure that quantifies the strength and direction of the linear relationship between two numerical variables. It assesses how well the data points of the two variables fit a straight line, providing a value between -1 and 1, where:

- r = +1 indicates a perfect positive linear relationship, meaning that as one variable increases, the other also increases proportionally.

- r = -1 indicates a perfect negative linear relationship, meaning that as one variable increases, the other decreases proportionally.

- r = 0 indicates no linear relationship between the two variables.

The formula for calculating the Pearson correlation coefficient is as follows:

$$ r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} $$

Where:

- x_i and y_i are the individual data points of the two numerical variables.

- $\bar{x}$ and $\bar{y}$ are the means of the two variables, respectively.

To compute the Pearson correlation coefficient for a given set of numerical data, follow these steps:

1. Calculate the mean of both numerical columns (variables), $\bar{x}$ and $\bar{y}$.

2. For each data point in the two columns, subtract the corresponding means, i.e., $(x_i - \bar{x})$ and $(y_i - \bar{y})$.

3. Square the results obtained in step 2, i.e., $(x_i - \bar{x})^2$ and $(y_i - \bar{y})^2$.

4. Multiply the differences obtained in step 2 for each data point, i.e., $(x_i - \bar{x})(y_i - \bar{y})$.

5. Sum all the results obtained in step 4.

6. Calculate the square root of the sum of squared differences from step 3 for both columns.

7. Divide the sum obtained in step 5 by the square root obtained in step 6.

The final result is the Pearson correlation coefficient (r). A positive value indicates a positive linear relationship, a negative value indicates a negative linear relationship, and a value close to zero indicates a weak or no linear relationship between the two numerical variables.

11. SciPy

Introduction to the Scipy.Stats Library:

The Scipy.Stats library is a powerful module within the Scipy ecosystem that provides a wide range of statistical functions and distributions for data analysis in Python. This library offers a comprehensive set of tools for conducting various statistical tests, probability calculations, and data modeling. By leveraging Scipy.Stats, researchers, data scientists, and analysts can efficiently perform statistical analyses, hypothesis testing, and probability computations with ease.

Key Features of Scipy.Stats:

1. Probability Distributions: Scipy.Stats includes an extensive collection of probability distributions, such as normal, binomial, Poisson, exponential, and many others. These distributions enable users to model and analyze various real-world phenomena and random variables, making it easier to assess the likelihood of different outcomes.

2. Descriptive Statistics: The library provides essential descriptive statistics functions, such as mean, median, variance, standard deviation, and percentiles. These functions offer valuable insights into the central tendency and spread of the data, facilitating data exploration and summarization.

3. Hypothesis Testing: Scipy.Stats offers a wide range of statistical tests to assess hypotheses and make inferences about data samples. Common hypothesis tests, including t-tests, ANOVA, chi-square tests, and more, are readily available for comparing sample groups and drawing conclusions from data.

4. Correlation and Regression: The library provides functions for computing correlation coefficients, such as Pearson, Spearman, and Kendall, to analyze relationships between variables. Additionally, users can perform linear and non-linear regression analyses to model and predict data trends.

5. Probability Functions: Scipy.Stats includes functions to calculate probability density functions (PDFs), cumulative distribution functions (CDFs), and quantiles for various probability distributions. These functions allow users to estimate probabilities and assess the likelihood of specific events.

6. Statistical Summary Functions: Users can generate summary statistics for datasets using functions like `describe`, which provides a comprehensive summary including count, mean, standard deviation, minimum, maximum, and quartiles.

7. Random Number Generation: Scipy.Stats offers facilities for generating random samples from various probability distributions, allowing users to simulate and model random processes.

Getting Started with Scipy.Stats:

To begin using the Scipy.Stats library, you first need to install Scipy if you haven't already. You can install Scipy using pip:
pip install scipy
from scipy import stats

Conclusion:

Scipy.Stats is an indispensable tool for statistical analysis and probability calculations in Python. Its diverse range of statistical functions, probability distributions, and hypothesis testing capabilities empower users to gain valuable insights from data and make data-driven decisions. Whether you are performing hypothesis tests, computing probability distributions, or analyzing correlation and regression, Scipy.Stats is a valuable library that can significantly enhance your data analysis workflow.

12. ANOVA-For Categorical Column

ANOVA (Analysis of Variance) is a statistical method used to compare the means of two or more groups to determine if there are any significant differences among them. While ANOVA is commonly used for numerical data, it can also be applied to analyse the variation between groups for a categorical column.

When dealing with a categorical column, the data typically consists of categories or groups, and we want to assess whether there are statistically significant differences in a numerical variable across these categories. The one-way ANOVA is the appropriate test for comparing the means of a numerical variable among multiple groups (categories) of a categorical variable.

Please note that before conducting ANOVA, it's essential to ensure that the assumptions of the test are met, such as normality of the data and homogeneity of variances. If the assumptions are not met, alternative non-parametric tests may be more appropriate, such as the Kruskal-Wallis test.

Keep in mind that this is a general outline of how to perform ANOVA for a categorical column. The specific implementation may vary depending on your dataset and the tools you are using for data manipulation and analysis.

13. Power BI

Power BI is a business analytics service provided by Microsoft that allows users to visualize and analyze their data quickly and effectively. It enables data-driven decision-making by transforming raw data into interactive and visually appealing reports and dashboards. Power BI is widely used by businesses to gain insights from their data, monitor key performance indicators (KPIs), and share valuable information with stakeholders.

Here are some key features and components of Power BI:

1. Data Sources: Power BI can connect to a wide range of data sources, including Excel spreadsheets, SQL databases, cloud-based services (such as Azure, Google Analytics, Salesforce), and many others.

2. Data Transformation and Modelling: Within Power BI, you can perform data transformation tasks like cleaning, shaping, and combining data from different sources. The Power Query Editor allows you to perform these data preparation steps.

3. Data Visualization: Power BI offers a variety of visualization options, including charts, graphs, tables, maps, and more. Users can drag and drop fields onto the report canvas to create visual representations of their data.

4. Reports: Reports in Power BI are interactive and dynamic. Users can explore data, filter information, drill down into details, and ask ad-hoc questions. Reports can also be connected to live data sources, enabling real-time insights.

5. Dashboards: Dashboards are collections of visualizations and KPIs from multiple reports. They provide a high-level overview of important metrics and allow users to monitor the health of their business at a gla

6. Power BI Service: The Power BI service is a cloud-based platform where users can publish and share their reports and dashboards with others. It provides collaboration features, data refresh options, and access control.

7. Power BI Desktop: Power BI Desktop is a Windows application used for creating complex reports and dashboards. It allows users to design reports offline and later publish them to the Power BI service.

8. Power BI Mobile: Power BI offers mobile apps for iOS and Android devices, allowing users to access their reports and dashboards on the go.

9. Power BI Embedded: Power BI Embedded allows developers to integrate Power BI reports and dashboards into their own applications, extending the capabilities of Power BI to custom solutions.

Power BI provides a user-friendly interface, making it accessible to both technical and non-technical users. It has become a popular choice for data analysis and visualization due to its ease of use, rich set of features, and integration with other Microsoft products and services.

14. **Conclusion:**

The conclusion of the e-commerce data analysis depends on the specific objectives, data, and analysis performed

"After conducting a comprehensive data analysis of the e-commerce dataset, several key insights and trends have been identified, providing valuable information for business decision-making and strategy development. The analysis encompassed various aspects, including customer behaviour, sales performance, product Category, and Churn Rate. The following are some of the key findings:

Customer Segmentation: Through clustering analysis, distinct customer segments were identified based on their purchase behaviour and preferences. This allows for targeted marketing strategies and personalized recommendations to enhance customer satisfaction and retention.

Sales Performance: The analysis revealed specific periods of peak sales and identified high-demand products. This information can guide inventory management and promotional efforts to optimize sales revenue.

Customer Retention: The analysis highlighted the importance of customer retention in driving long-term revenue. Implementing customer retention strategies, such as Churn Rate and preferred Categories, Satisfaction Score.

Ecommerce retailers can also create special programmes that focus on rewarding their best customers, to foster brand loyalty and boost the customer retention rate.

How you reward these customers will depend on your business model, but examples include offering special discounts, early bird access to sales and exclusive event invites and content for your best customers to make them feel like you value them as a customer.