

# E-Commerce Customer Churn Report Analysis

## Introduction

- This is a dataset of a leading ecommerce company and we have an analysis who are churn(leaving the company service) and have to make a predicting churn model.
- It can cost anywhere between 5 and 25 times more to attract new customers than it does to retain existing ones. Statistics show an increase in customer retention by 5% can lead to a company's profits growing by 25% to around 95% over a period of time. So I will build ML models to predict customer churn using data collected from e-commerce.

## Features

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df = pd.read_excel('content/E Commerce Dataset.xlsx', sheet_name = 'E Comm')
df
```

CustomerID	Churn	Tenure	PreferredLoginDevice	CityTier	WarehouseToHome	PreferredPaymentMode	Gender	HourSpendOnApp	NumberOfDeviceRegist
0	50001	1	4.0	Mobile Phone	3	6.0	Debit Card	Female	3.0
1	50002	1	NaN	Phone	1	8.0	UPI	Male	3.0
2	50003	1	NaN	Phone	1	30.0	Debit Card	Male	2.0
3	50004	1	0.0	Phone	3	15.0	Debit Card	Male	2.0
4	50005	1	0.0	Phone	1	12.0	CC	Male	NaN
...	...	...	...	...	...	...	...	...	...
5625	55626	0	10.0	Computer	1	30.0	Credit Card	Male	3.0
5626	55627	0	13.0	Mobile Phone	1	13.0	Credit Card	Male	3.0

- CustomerID: Unique customer ID
- Churn: Churn Flag
- Tenure: Tenure of customer in organisation.
- PreferredLoginDevice: Preferred login device of customer
- CityTier: City tier
- WarehouseToHome: Distance in between warehouse to home of customer
- PreferredPaymentMode: Preferred payment method of customer
- Gender: Gender of customer
- HourSpendOnApp: Number of hours spend on mobile application or website
- NumberOfDeviceRegistered: Total number of deceives is registered on particular customer
- PreferedOrderCat: Preferred order category of customer in last month
- SatisfactionScore: Satisfactory score of customer on service
- MaritalStatus: Marital status of customer
- NumberOfAddress: Total number of added added on particular customer
- OrderAmountHikeFromlastYear: Percentage increases in order from last year
- CouponUsed: Total number of coupon has been used in last month
- OrderCount: Total number of orders has been places in last month
- DaySinceLastOrder: Day Since last order by customer
- CashbackAmount: Average cashback in last month

## **Mission**

- Build a predictive model that can accurately identify customers who are at risk of leaving the company (churn) based on the provided variables. This can help the company take proactive steps to retain these customers and reduce the rate of churn.
- Perform a thorough exploratory analysis of the provided customer data to gain insights into the behaviour and characteristics of the customers. This includes analysing patterns and trends in variables. This analysis can help the company understand its customers better and inform future decision-making.

# PLAN

## Datasets Overview

- Review the provided customer data to familiarise yourself with the variables and their structure.
- Check the data quality, missing values, and potential errors.
- Determine if any data pre-processing is necessary

## Exploratory Data Analysis

- Analyse the distribution of the variables to identify any outliers or anomalies.
- Investigate the relationship between variables to identify any correlations or patterns.
- Visualise the data to gain insights into the behaviour and characteristics of the customers.

## Pre-Processing

- Clean the data by handling missing values, converting variables to appropriate data types, and addressing any data quality issues.
- Select the most important variables for building the predictive model

## Machine Learning

- Build a predictive model that can identify customers who are at risk of leaving the company

# 1. Setup Environment & Import Libraries

install all required libraries .

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler , LabelEncoder
from sklearn.svm import SVC

# Additional imports
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score , confusion_matrix , classification_report
from sklearn.model_selection import GridSearchCV, cross_validate

import warnings
warnings.simplefilter(action='ignore')
```

## 2. Load the Data

Load your customer data from a CSV file, database, or any other source

```
df = pd.read_excel('/content/E Commerce Dataset.xlsx', sheet_name = 'E Comm')
df
```

Python

## 3. Initial Data Exploration

Explore the basic structure of the data, including the first few rows, data types, and summary statistics

The screenshot shows a Jupyter Notebook cell with the code `df.info()` executed. The output displays the structure of the DataFrame, including the number of entries (5630) and the data types of the columns.

```
df.info()
Out[15]:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5630 entries, 0 to 5629
Data columns (total 20 columns):
 #   Column                        Non-Null Count  Dtype
---  ---
 0   CustomerID                   5630 non-null  int64
 1   Churn                        5630 non-null  int64
 2   Tenure                       5630 non-null  float64
 3   PreferredLoginDevice         5630 non-null  object
 4   CityTier                     5630 non-null  int64
 5   WarehouseToHome             5379 non-null  float64
 6   PreferredPaymentMode         5630 non-null  object
 7   Gender                       5630 non-null  object
 8   HoursSpendOnApp              5375 non-null  float64
 9   NumberofDeviceRegistered     5630 non-null  int64
10   PreferredOrdercat            5630 non-null  object
11   SatisfactionScore            5630 non-null  int64
12   MaritalStatus                5630 non-null  object
13   NumberofAddress              5630 non-null  int64
14   Complain                     5630 non-null  int64
15   OrderAmountHikeFromLastYear  5365 non-null  float64
16   CouponUsed                   5374 non-null  float64
17   OrderCount                   5372 non-null  float64
18   DaySinceLastOrder            5323 non-null  float64
19   CashbackAmount               5630 non-null  float64
dtypes: float64(8), int64(7), object(5)
```

## 4. Data Quality Assessment

Identify any anomalies, missing values, or inconsistencies in the dataset.

- **Missing Values:** Identify the columns with missing values and decide on a strategy for handling them (e.g., imputation, deletion).
- **Data Types:** Ensure each column is of the correct data type (numerical, categorical).
- **Outliers:** Detect outliers and decide how to handle them.

File Edit Selection View Go Run Terminal Help demo\_sam\_app

Source Control 10000

- gitconfig C:\Users\91741\...
- JessHst C:\Users\91741\...
- viminfo C:\Users\91741\...
- 1f5bac0ca01cf648c02ea76a68a... C:\Users\91741\...
- CACHEDIR.TAG C:\Users\91741\...
- d031bbba323fd9e5b47e0ee5a... C:\Users\91741\...
- versions.json C:\Users\91741\...
- aau\_token C:\Users\91741\...
- environments.txt C:\Users\91741\...
- hello.json C:\Users\91741\...
- token\_seed C:\Users\91741\...
- token\_seed.lock C:\Users\91741\...
- config.json C:\Users\91741\...
- daemon.json C:\Users\91741\...
- buildNodeID C:\Users\91741\...
- lock C:\Users\91741\...
- current C:\Users\91741\...
- vghu34xkx445bnpufnidm... C:\Users\91741\...
- meta.json C:\Users\91741\...
- lock C:\Users\91741\...
- sbom.json C:\Users\91741\...
- sbom.json C:\Users\91741\...
- recent-files.lst C:\Users\91741\...
- my-1st-notebook-checkpoint... C:\Users\91741\...
- history.sqlite C:\Users\91741\...
- README C:\Users\91741\...
- migrated C:\Users\91741\...
- keras.json C:\Users\91741\...
- cfar-10-batches-py.tar.gz C:\Users\91741\...
- shakespeare.txt C:\Users\91741\...
- shakespeare.txt C:\Users\91741\...
- batches.meta C:\Users\91741\...

Modeling.ipynb

```
daySinceLastOrder float64
CashbackAmount float64
dtype: object
```

```
df.duplicated().sum()
[28] 0
```

```
# the sum of null values
grouped_data = []
for col in columns:
    n_missing = df[col].isnull().sum()
    percentage = n_missing / df.shape[0] * 100
    grouped_data.append([col, n_missing, percentage])

# Create a new DataFrame from the grouped data
grouped_df = pd.DataFrame(grouped_data, columns=['column', 'n_missing', 'percentage'])

# Group by 'col', 'n_missing', and 'percentage'
result = grouped_df.groupby(['column', 'n_missing', 'percentage']).size()
result
[29]
```

column	n_missing	percentage
CashbackAmount	0	0.000000
Churn	0	0.000000

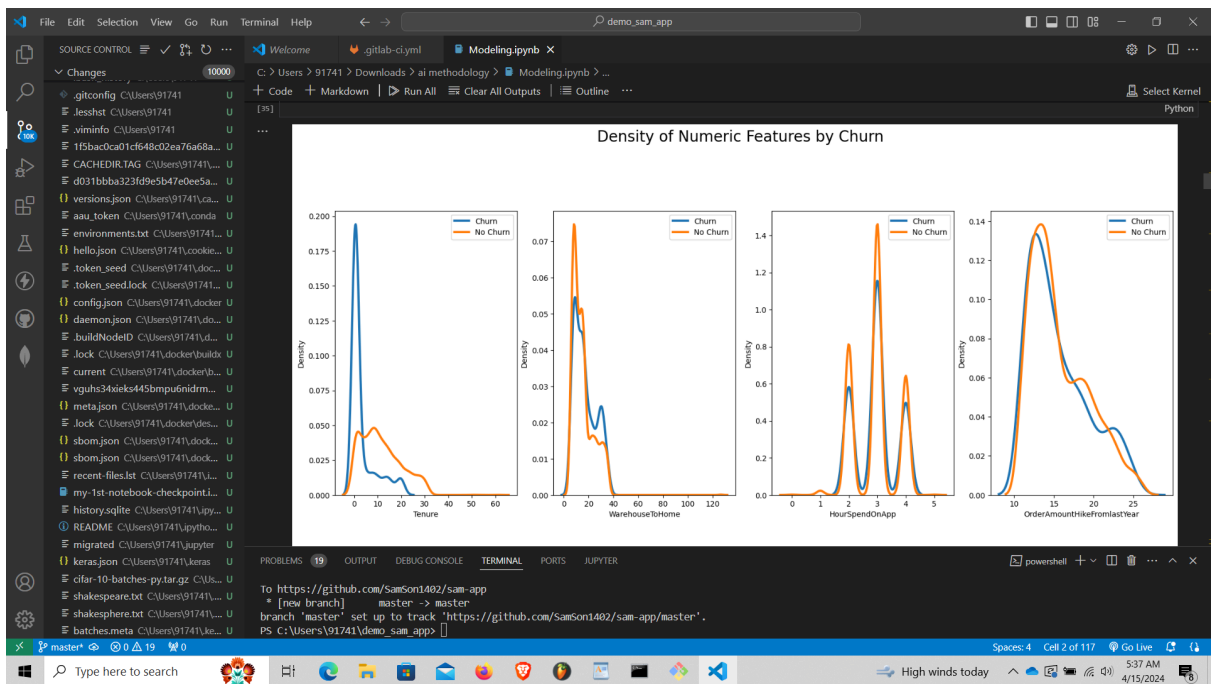
PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER

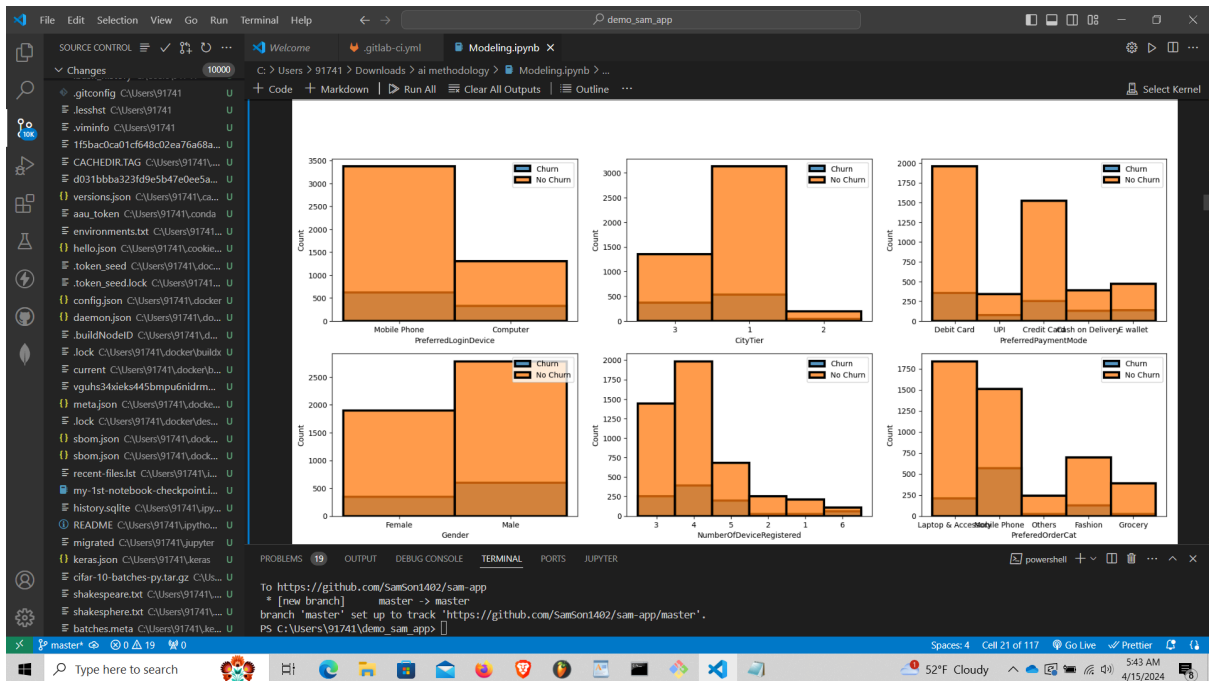
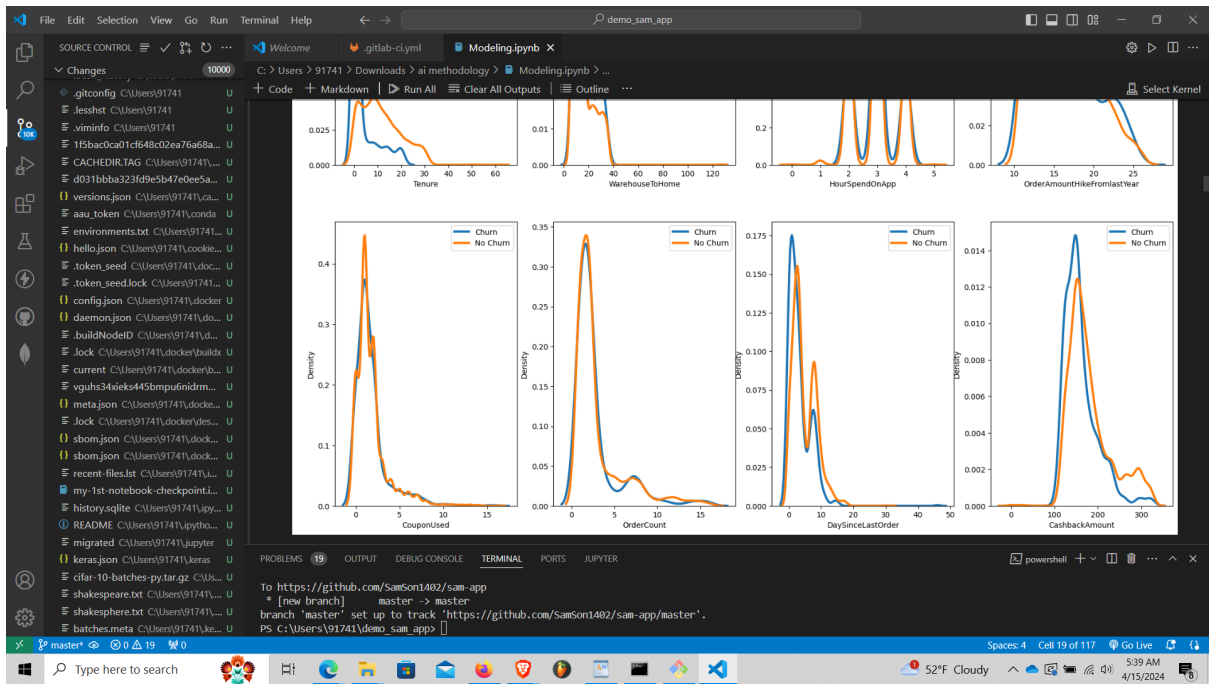
```
To https://github.com/SamSoni1402/sam-app
* [new branch] master -> master
branch 'master' set up to track 'https://github.com/SamSoni1402/sam-app/master'.
PS C:\Users\91741\demo_sam_app>
```

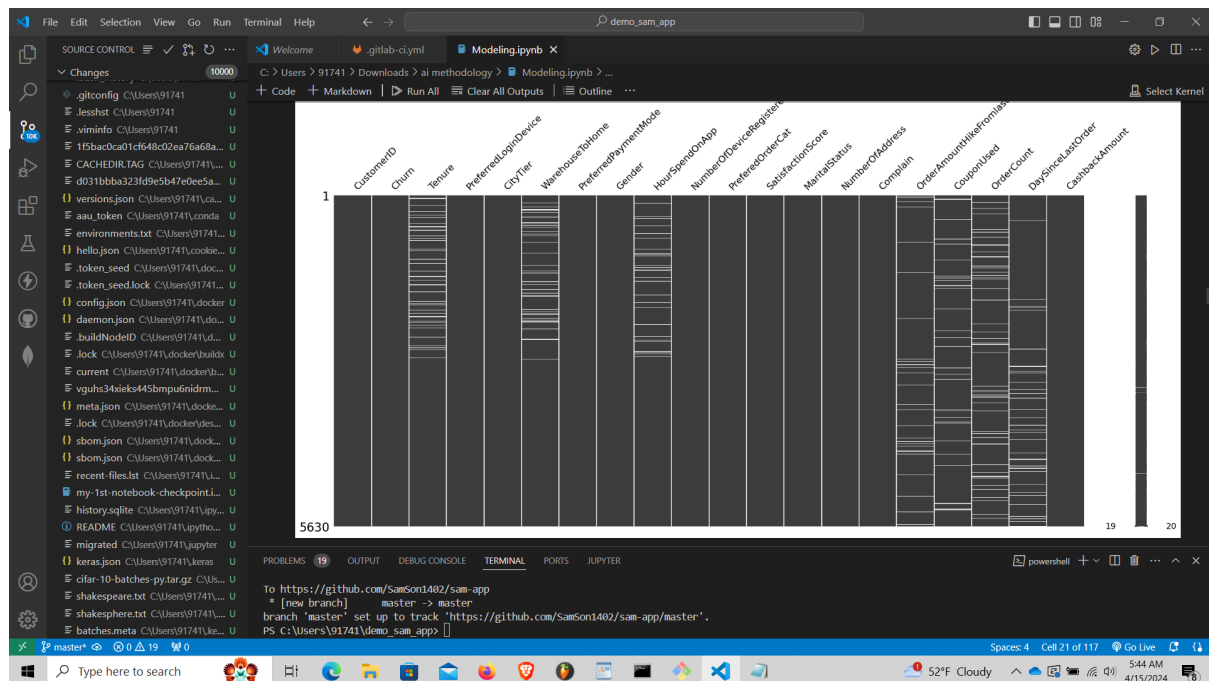
Spaces: 4 Cell 2 of 117 Go Live

Type here to search

52°F Cloudy 5:36 AM 4/15/2024







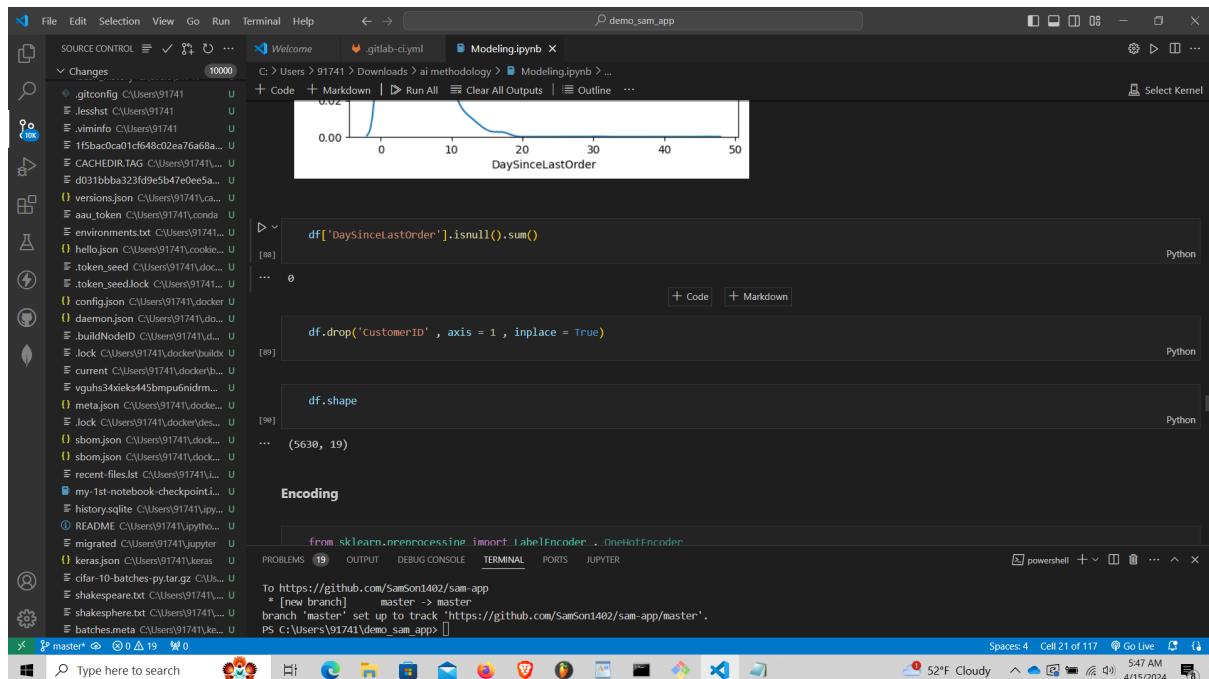
- \* **Tenure:** Customers with longer tenure seem less likely to churn. Makes sense as longer tenure indicates satisfaction
- \* **CityTier:** Churn rate looks similar across tiers. City tier does not seem predictive of churn
- \* **WarehouseToHome:** Shorter warehouse to home distances have a lower churn rate. Faster deliveries may improve satisfaction
- \* **HourSpendOnApp:** More time spent on app correlates with lower churn. App engagement is a good sign
- \* **NumberOfDeviceRegistered:** More registered devices associates with lower churn. Access across devices improves convenience
- \* **SatisfactionScore:** Higher satisfaction scores strongly associate with lower churn, as expected. Critical driver
- \* **NumberOfAddress:** Slight downward trend in churn as number of addresses increases. More addresses indicates loyalty
- \* **Complain:** More complaints associate with higher churn, though relationship isn't very strong. Complaints hurt satisfaction



- \* OrderAmountHikeFromLastYear: Big spenders from last year are less likely to churn. Good to retain big customers
- \* CouponUsed: Coupon usage correlates with lower churn. Coupons enhance loyalty
- \* OrderCount: Higher order counts associate with lower churn. Frequent usage builds habits
- \* DaySinceLastOrder: Longer since last order correlates with higher churn. Recency is a good predictor

## 5. Data Preprocessing

### Handling Missing Values:



# Convert Data Types:

```
le = LabelEncoder()

for i in df.columns:
    if df[i].dtype == 'object':
        df[i] = le.fit_transform(df[i])

df.head(4)
```

	Churn	Tenure	PreferredLoginDevice	CityTier	WarehouseToHome	PreferredPaymentMode	Gender	HourSpendOnApp	NumberOfDeviceRegistered	PreferredOn
0	1	4.0	1	3	6.0	2	0	3.0		3
1	1	0.0	1	1	8.0	4	1	3.0		4
2	1	0.0	1	1	30.0	2	1	2.0		4
3	1	0.0	1	3	15.0	2	1	2.0		4

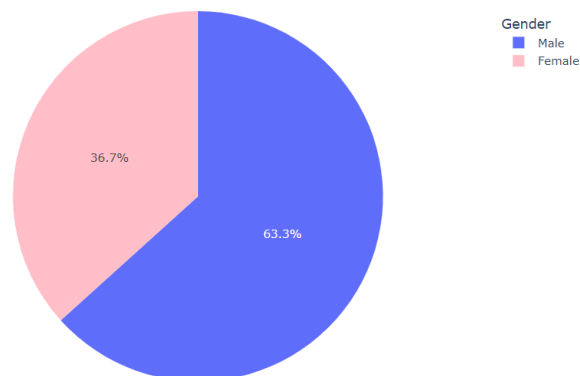
```
for i in data.columns:
    data[i] = le.fit_transform(data[i])

data.head(4)
```

## EDA (Exploratory Data Analysis)

1. Is there a relationship between Gender and Churn? & Which Gender has more Orders?

Churn Rate by Gender



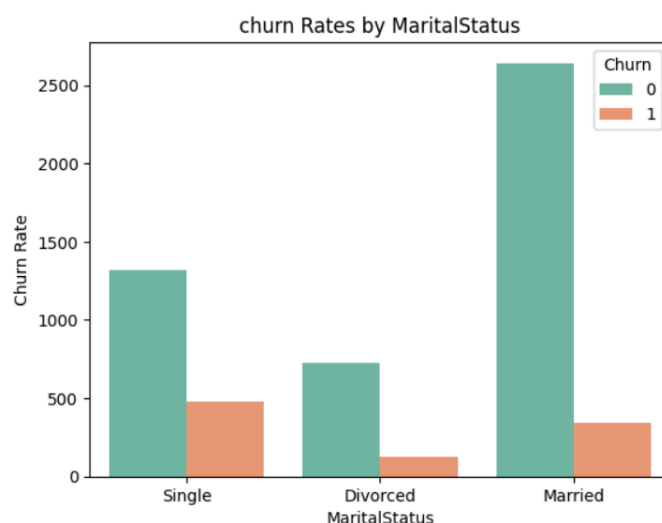
The data reveals an intriguing trend: a significantly higher churn rate among male users, with 63.3% of churned customers being males. This observation prompts a deeper investigation into potential factors driving this disparity. While one plausible explanation suggests an opportunity to expand product offerings tailored to male interests, a more comprehensive analysis is warranted to uncover additional insights.

By delving into the data, we aim to identify underlying reasons contributing to the elevated churn rate among male users. This analysis extends beyond gender segmentation to explore nuanced behavioural patterns, engagement metrics, and external influences. Such an approach not only sheds light on the current landscape but also informs strategic decision-making for mitigating churn and enhancing customer retention.

Drawing upon advanced analytical techniques and strategic foresight, we endeavour to uncover actionable insights that empower the company to tailor its offerings, refine its marketing strategies, and bolster customer engagement initiatives. Through a holistic understanding of customer behaviour and preferences, we pave the way for targeted interventions that resonate with male users and foster long-term loyalty.

In essence, our objective is to transform raw data into actionable intelligence, guiding the company towards informed decisions and sustainable growth in an ever-evolving market landscape

## 2-Which MaritalStatus has the highest Churn rate?



The data highlights an intriguing trend: married customers constitute the largest segment within the company's customer base. However, a closer examination reveals that single customers exhibit a higher likelihood of churning from the app. This insight underscores the importance of tailoring product offerings and customer experiences to meet the diverse needs of both single and married individuals.

Given the elevated churn rate among single customers, there is an opportunity for the company to prioritise initiatives that resonate with this demographic. By understanding the unique preferences and pain points of single customers, the company can develop targeted strategies to enhance their satisfaction and loyalty. This may involve introducing personalised product recommendations, implementing engaging marketing campaigns, or providing exclusive benefits tailored to single individuals.

Simultaneously, it's essential for the company to continue catering to the needs of married customers, leveraging their significant presence within the customer base. By maintaining a customer-centric approach and delivering value-added experiences to married individuals, the company can strengthen relationships and foster long-term loyalty.

In essence, our analysis underscores the importance of a nuanced approach to customer segmentation and engagement. By recognizing the distinct characteristics and behaviours of single and married customers, the company can adapt its strategies to better meet their needs, reduce churn, and drive sustainable growth in the competitive market landscape

### 3-Which CityTier has higher Tenure and OrderCount?

	mean	max
CityTier		
1	10.528818	51.0
2	11.169725	31.0
3	9.361740	61.0

```
df_grouped_OrderCount = df.groupby('CityTier')['OrderCount'].agg(['mean', 'max'])
df_grouped_OrderCount
```

	mean	max
CityTier		
1	2.953255	16.0
2	2.584034	13.0
3	3.185185	16.0

While CityTier 2 exhibits the highest tenure rate among customers, it appears that tenure alone may not be a significant factor influencing churn. This observation prompts a deeper exploration into the interplay between CityTier and other variables to understand their collective impact on customer behavior.

CityTier 2's higher tenure rate suggests a level of stability or satisfaction among customers in this demographic segment. However, it's essential to consider additional factors that may contribute to churn, such as preferred payment mode, satisfaction score, or order frequency.

A comprehensive analysis could involve examining churn rates across different CityTiers while controlling for other variables. By conducting regression analysis or employing machine learning models, we can identify the relative importance of CityTier in predicting churn while considering its interaction with other features.

Furthermore, qualitative research methods like customer surveys or interviews can provide valuable insights into the underlying reasons for churn among customers in different CityTiers. Understanding the unique challenges and preferences of customers in CityTier 2 can inform targeted retention strategies tailored to address their specific needs and concerns.

In summary, while CityTier 2 may have a higher tenure rate, its influence on churn requires a nuanced examination in conjunction with other variables. By conducting a comprehensive analysis and leveraging qualitative insights, the company can develop more effective churn mitigation strategies to enhance customer retention across all CityTiers.

```
df.groupby("CityTier")["OrderCount"].mean()
```

```
CityTier
1      2.953255
2      2.584034
3      3.185185
Name: OrderCount, dtype: float64
```

Although CityTier 3 boasts the highest average order amount, this metric alone may not significantly impact customer churn. To gain deeper insights into the relationship between order amount and churn, it's crucial to explore other variables that could influence customer behavior.

While a high average order amount in CityTier 3 indicates robust purchasing behavior, it's essential to consider additional factors such as satisfaction score, preferred order category, or frequency of complaints. These variables may provide context for understanding why customers in CityTier 3 churn despite their substantial spending.

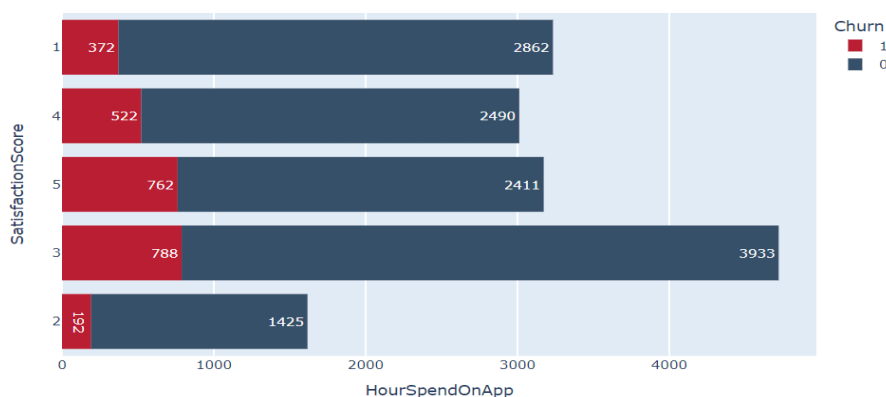
A holistic analysis could involve examining churn rates across different CityTiers while controlling for other relevant variables. By employing regression analysis or machine learning models, we can assess the relative importance of order amount in predicting churn within the context of CityTier and other features.

Moreover, qualitative research methods like customer interviews or focus groups can offer valuable insights into the underlying motivations behind churn among CityTier 3 customers. Understanding their preferences, pain points, and satisfaction levels can inform targeted retention strategies aimed at mitigating churn effectively.

In summary, while CityTier 3 customers demonstrate a high average order amount, it's essential to explore the nuanced relationship between this metric and churn in conjunction with other variables. By conducting a comprehensive analysis and leveraging qualitative insights, the company can develop tailored retention strategies to address the specific needs of customers in CityTier 3 and enhance overall customer loyalty and satisfaction

## 4-Is Customer with High SatisfactionScore have high HourSpendOnApp?

**HourSpendOnApp Vs SatisfactionScore**



Although there appears to be a trend where individuals with a higher satisfaction score spend more time on the app compared to those with a lower satisfaction score, it's essential to conduct a deeper analysis to ascertain if there's a significant relationship between satisfaction score and app usage time.

While the data suggests a potential correlation, it's crucial to investigate other factors that may influence app usage independently of satisfaction score. For instance, demographic variables, such as age, gender, or occupation, could impact how individuals engage with the app regardless of their satisfaction level.

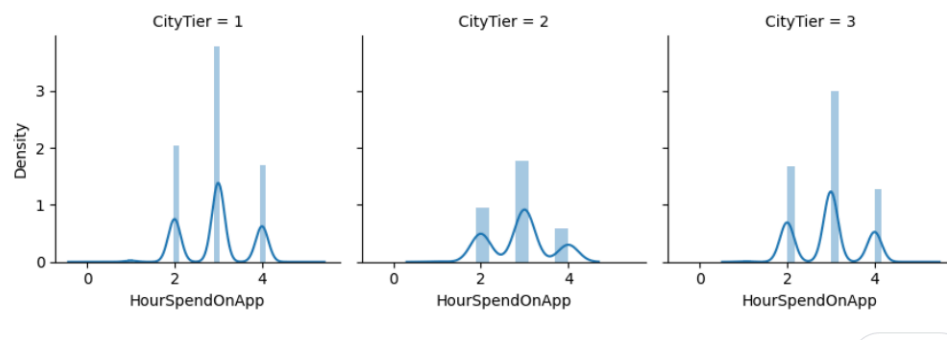
Moreover, external factors like seasonality, promotional campaigns, or changes in app features could also affect app usage patterns, regardless of satisfaction score. By accounting for these variables in our analysis, we can better understand the nuanced relationship between satisfaction score and app usage time.

To assess the strength of the relationship, statistical methods such as correlation analysis or regression modeling can provide insights into the extent to which satisfaction score predicts app usage time while controlling for confounding variables.

Furthermore, qualitative research methods like user surveys or interviews can offer additional context by uncovering the underlying motivations and preferences driving app usage behavior. By combining quantitative and qualitative approaches, we can gain a comprehensive understanding of the factors influencing app engagement and satisfaction among users.

In summary, while there may be a general trend suggesting a positive relationship between satisfaction score and app usage time, further analysis is needed to determine the extent of this relationship and identify other influential factors. By conducting a rigorous analysis and considering both quantitative and qualitative insights, we can develop actionable strategies to enhance user engagement and satisfaction on the app.

## 5-Which CityTier has the most HourSpendOnApp?



Although individuals from city tier 1 demonstrate the highest average hours spent on the app, it's crucial to delve deeper into the data to understand the underlying factors driving this trend and its potential implications.

While the data suggests a correlation between city tier and app usage time, other variables may influence app engagement independently of city tier. For example, demographic factors like age, income level, or occupation could impact how individuals from different city tiers interact with the app.

Moreover, contextual factors such as lifestyle preferences, access to technology, or cultural norms may also play a role in shaping app usage patterns across city tiers. By considering these variables in our analysis, we can better understand the nuanced relationship between city tier and app usage time.

To assess the robustness of the relationship, statistical techniques like regression analysis or segmentation modeling can help identify the key drivers of app engagement within each city tier. By controlling for confounding variables and exploring interactions between predictors, we can uncover the unique factors contributing to higher app usage in city tier 1.

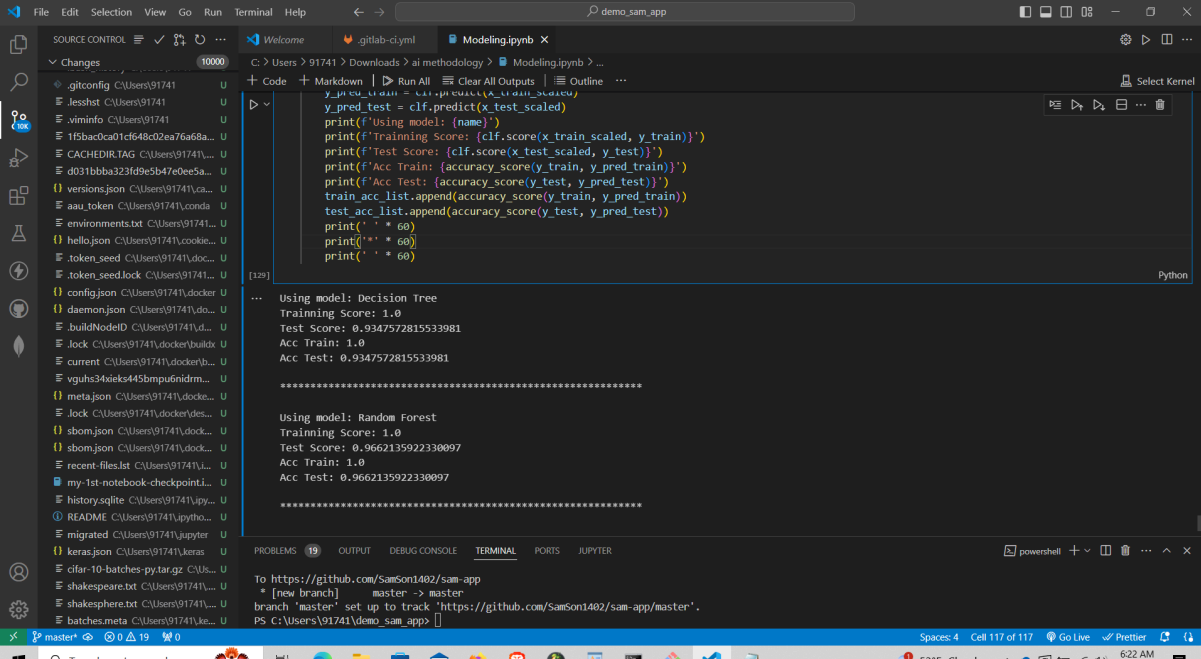
Furthermore, qualitative research methods such as focus groups or user interviews can provide valuable insights into the motivations and behaviors of app users across different city tiers. By capturing the perspectives of users directly, we can gain a deeper understanding of their needs, preferences, and pain points related to app usage.

In summary, while city tier 1 may exhibit higher average app usage time, further analysis is necessary to elucidate the underlying drivers and implications of this



trend. By employing a combination of quantitative and qualitative methods, we can develop targeted strategies to enhance app engagement and satisfaction among users across all city tiers

## Modelling



```
...  
y_pred_test = clf.predict(x_test_scaled)  
print(f'Using model: {name}')  
print(f'Training Score: {clf.score(x_train_scaled, y_train)}')  
print(f'Test Score: {clf.score(x_test_scaled, y_test)}')  
print(f'Acc Train: {accuracy_score(y_train, y_pred_train)}')  
print(f'Acc Test: {accuracy_score(y_test, y_pred_test)}')  
train_acc_list.append(accuracy_score(y_train, y_pred_train))  
test_acc_list.append(accuracy_score(y_test, y_pred_test))  
print(' ' * 60)  
print('* ' * 60)  
print(' ' * 60)  
[129]  
Python  
...  
Using model: Decision Tree  
Training Score: 1.0  
Test Score: 0.9347572815533981  
Acc Train: 1.0  
Acc Test: 0.9347572815533981  
.....  
Using model: Random Forest  
Training Score: 1.0  
Test Score: 0.9662135922330097  
Acc Train: 1.0  
Acc Test: 0.9662135922330097  
.....  
To https://github.com/SamSon1402/sam-app  
* [new branch] master -> master  
branch 'master' set up to track 'https://github.com/SamSon1402/sam-app/master'.  
PS C:\Users\91741\demo_sam_app>
```

**The Random Forest model** exhibits high accuracy on both the training and test datasets, achieving a perfect score of 1.0 on the training data and an impressive accuracy of 0.9662 on the test data. This indicates that the model is performing exceptionally well in predicting the target variable.

The discrepancy between the training and test accuracies is minimal, suggesting that the model has generalised well to unseen data and is not overfitting. This is further supported by the high test accuracy, which indicates that the model is effectively capturing the underlying patterns in the data.

The high accuracy scores imply that the features used by the model are informative and relevant for predicting the target variable. However, it's essential to consider

other performance metrics such as precision, recall, and F1-score to assess the model's performance comprehensively.

Overall, based on the provided accuracy scores, the Random Forest model appears to be a robust and effective classifier for the given dataset. Further evaluation using additional metrics and techniques may provide deeper insights into the model's performance and potential areas for improvement

**The Decision Tree model demonstrates** excellent performance on the training dataset, achieving a perfect score of 1.0, indicating that it perfectly predicts the target variable on the data it was trained on. However, there is a slight decrease in performance on the test dataset, with an accuracy score of 0.9348.

While the model performs slightly worse on the test dataset compared to the training dataset, the difference in accuracy is minimal. This suggests that the model has generalized well to unseen data and is not overfitting to the training data.

The high accuracy scores on both the training and test datasets indicate that the features used by the model are informative and relevant for predicting the target variable. However, as with any machine learning model, it's essential to consider other evaluation metrics such as precision, recall, and F1-score to gain a comprehensive understanding of its performance.

In summary, the Decision Tree model appears to be a robust classifier for the given dataset, demonstrating strong predictive capabilities. Further analysis and evaluation using additional metrics may provide deeper insights into its performance and potential areas for improvement



