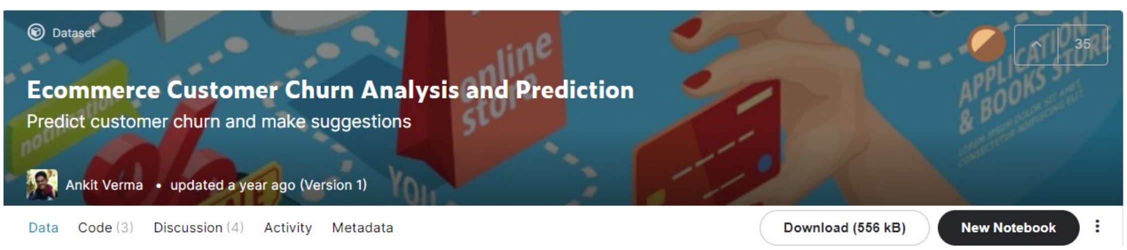
Rakamin Batch 19A Final Project -- EDA

Group 1 Members :

- Yosafat Respati
- Ridho Fajar
- Li'izza Diana M
- Teguh Tri Arvianto
- Vito Rihaldijiran
- M Supian Noor
- Rexy Anggala Putra

I. About This Project

In this project, from the Ecommerce Customer Churn Analysis and Prediction Dataset (https://www.kaggle.com/ankitverma2010/ecommerce-customer-churn-analysis-and-prediction), We want to do the Exploratory Data Analysis using Data Manipulation Libraries such as Numpy, Pandas, and Seaborn. The goal of this project is We want to Give Business Insights from the dataset about the correlation between users and ecommerce stuffs so in the future we can make a model that can predict the customer that has the possibility to churn or not.



II. What is Exploratory Data Analysis?

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns,to spot anomalies,to test hypothesis and to check assumptions with the help of summary statistics and graphical representations. (source = https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15)

III. Execution Plan

In this project, There are 2 Major Steps in order to get the business insights, which are :

- Descriptive Statistics
- Univariate Analysis
- Multivariate Analysis
- Business Insights

Step 1 : Descriptive Statistics

In this step, We will explore the Ecommerce Dataset using Descriptive statistics and check whether there are outliers, missing values in it.

Part 1 : Import the Libraries

The libraries that We used in this project :

- Numpy : for working with arrays and numerical operation
- Pandas : for data manipulation and analysis (also data cleaning)
- Matplotlib : for Data Visualizations
- Seaborn : for Data Visualizations

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

Part 2 : Load and Explore the Dataset

The Next step is Load the dataset with pandas and then take a look of the glimpse of the dataset such as missing values, outliers, aggregate summary, etc. The first one is we load the Data Dictionary. It contains about informations of each column in the dataset

```
df_dict = pd.read_excel('E_Commerce_Dataset.xlsx', sheet_name = 'Data Dict', header=1, usecols=[1,2,3])
df_dict
```

```
      Data                     Variable  \
0   E Comm                   CustomerID
1   E Comm                        Churn
2   E Comm                       Tenure
3   E Comm         PreferredLoginDevice
4   E Comm                     CityTier
5   E Comm              WarehouseToHome
6   E Comm         PreferredPaymentMode
7   E Comm                       Gender
8   E Comm                HourSpendOnApp
9   E Comm     NumberOfDeviceRegistered
10  E Comm             PreferedOrderCat
11  E Comm            SatisfactionScore
12  E Comm                MaritalStatus
13  E Comm              NumberOfAddress
14  E Comm                     Complain
15  E Comm  OrderAmountHikeFromlastYear
16  E Comm                   CouponUsed
17  E Comm                   OrderCount
18  E Comm            DaySinceLastOrder
19  E Comm               CashbackAmount
```

```
                                          Discerption
0                                  Unique customer ID
1                                          Churn Flag
2                     Tenure of customer in organization
3                     Preferred login device of customer
4                                           City tier
5   Distance in between warehouse to home of customer
6                   Preferred payment method of customer
7                                  Gender of customer
8   Number of hours spend on mobile application or...
9   Total number of deceives is registered on part...
10  Preferred order category of customer in last m...
11           Satisfactory score of customer on service
12                         Marital status of customer
13  Total number of added added on particular cust...
14          Any complaint has been raised in last month
15         Percentage increases in order from last year
16  Total number of coupon has been used in last m...
17  Total number of orders has been places in last...
18                   Day Since last order by customer
19                     Average cashback in last month
```

And then we load the main dataset that contains about 20 columns. Dont forget to specify the sheet name since it has 2 sheets in it.

```
df_main = pd.read_excel('E_Commerce_Dataset.xlsx', sheet_name = 'E
Comm')
df_main.head()
```

```
   CustomerID  Churn  Tenure PreferredLoginDevice  CityTier
WarehouseToHome  \
0       50001      1     4.0         Mobile Phone         3
6.0
1       50002      1     NaN                Phone         1
8.0
2       50003      1     NaN                Phone         1
30.0
3       50004      1     0.0                Phone         3
15.0
4       50005      1     0.0                Phone         1
12.0

  PreferredPaymentMode  Gender  HourSpendOnApp
NumberOfDeviceRegistered  \
0          Debit Card  Female             3.0
3
1                 UPI    Male             3.0
4
2          Debit Card    Male             2.0
4
3          Debit Card    Male             2.0
```

```
4
4                   CC    Male             NaN
3

     PreferedOrderCat  SatisfactionScore MaritalStatus
NumberOfAddress  \
0  Laptop & Accessory                  2        Single
9
1              Mobile                  3        Single
7
2              Mobile                  3        Single
6
3  Laptop & Accessory                  5        Single
8
4              Mobile                  5        Single
3

   Complain  OrderAmountHikeFromlastYear  CouponUsed  OrderCount  \
0         1                         11.0         1.0         1.0
1         1                         15.0         0.0         1.0
2         1                         14.0         0.0         1.0
3         0                         23.0         0.0         1.0
4         0                         11.0         1.0         1.0

   DaySinceLastOrder  CashbackAmount
0                5.0          159.93
1                0.0          120.90
2                3.0          120.28
3                3.0          134.07
4                3.0          129.60
```

```
df_main.shape
```

```
(5630, 20)
```

from the shape function, we know that the dataset has 5630 rows and 20 columns.

```
df_main.info()
```

```
<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                       5366 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   HourSpendOnApp               5375 non-null   float64
 9   NumberOfDeviceRegistered     5630 non-null   int64  
 10  PreferedOrderCat             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)
memory usage: 879.8+ KB
```

By using df.info(), you will know the information about the columns type

And then We check the missing values. Check it by using isnull()

```
df_main.isnull().sum()
```

```
CustomerID                       0
Churn                            0
Tenure                         264
PreferredLoginDevice             0
CityTier                         0
WarehouseToHome                251
PreferredPaymentMode             0
Gender                           0
HourSpendOnApp                 255
NumberOfDeviceRegistered         0
PreferedOrderCat                 0
SatisfactionScore                0
MaritalStatus                    0
NumberOfAddress                  0
Complain                         0
OrderAmountHikeFromlastYear    265
CouponUsed                     256
OrderCount                     258
DaySinceLastOrder              307
CashbackAmount                   0
dtype: int64
```

```
df_main.isnull().sum()/len(df_main)*100
```

```
CustomerID                     0.000000
Churn                          0.000000
Tenure                         4.689165
PreferredLoginDevice           0.000000
CityTier                       0.000000
```

```
WarehouseToHome              4.458259
PreferredPaymentMode         0.000000
Gender                       0.000000
HourSpendOnApp               4.529307
NumberOfDeviceRegistered     0.000000
PreferedOrderCat             0.000000
SatisfactionScore            0.000000
MaritalStatus                0.000000
NumberOfAddress              0.000000
Complain                     0.000000
OrderAmountHikeFromlastYear  4.706927
CouponUsed                   4.547069
OrderCount                   4.582593
DaySinceLastOrder            5.452931
CashbackAmount               0.000000
dtype: float64
```

After checking the dataset, We Know that 7 columns have missing values in it and average is 4-5% of total values of the datasets. if we want to build ML Model, we should take care this by simply drop the dataset or impute them

The Next Step is We check is there any Duplication in the dataset. we use the df.duplicated to do that.

```
df_main[df_main.duplicated(keep=False)]
```

```
Empty DataFrame
Columns: [CustomerID, Churn, Tenure, PreferredLoginDevice, CityTier,
WarehouseToHome, PreferredPaymentMode, Gender, HourSpendOnApp,
NumberOfDeviceRegistered, PreferedOrderCat, SatisfactionScore,
MaritalStatus, NumberOfAddress, Complain, OrderAmountHikeFromlastYear,
CouponUsed, OrderCount, DaySinceLastOrder, CashbackAmount]
Index: []
```

after the checking, there is no Duplicate Value in the dataset.

After that we subset the dataset into just numerical columns. I want to know the statistical numbers by using describe method

```
df_main_num = df_main[['Tenure', 'WarehouseToHome', 'HourSpendOnApp',
'NumberOfDeviceRegistered',
                       'SatisfactionScore', 'NumberOfAddress',
'OrderAmountHikeFromlastYear',
                       'CouponUsed', 'OrderCount',
'DaySinceLastOrder', 'CashbackAmount']]

nums = df_main[['Tenure', 'WarehouseToHome', 'HourSpendOnApp',
'NumberOfDeviceRegistered',
                       'SatisfactionScore', 'NumberOfAddress',
'OrderAmountHikeFromlastYear',
```

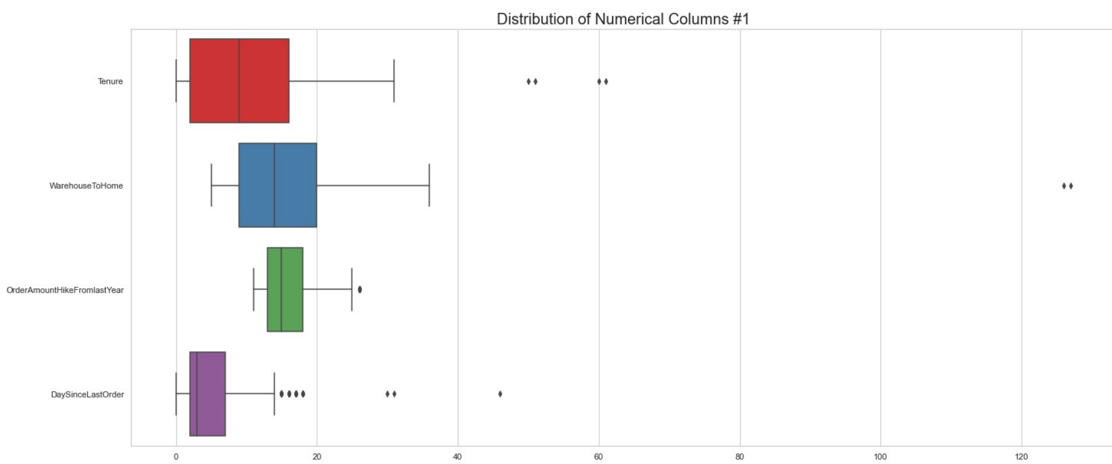
```
                       'CouponUsed', 'OrderCount',
'DaySinceLastOrder', 'CashbackAmount']]

df_main_num.describe(include = 'all')

            Tenure  WarehouseToHome  HourSpendOnApp
NumberOfDeviceRegistered  \
count  5366.000000      5379.000000     5375.000000
5630.000000
mean     10.189899        15.639896        2.931535
3.688988
std       8.557241         8.531475        0.721926
1.023999
min       0.000000         5.000000        0.000000
1.000000
25%       2.000000         9.000000        2.000000
3.000000
50%       9.000000        14.000000        3.000000
4.000000
75%      16.000000        20.000000        3.000000
4.000000
max      61.000000       127.000000        5.000000
6.000000

       SatisfactionScore  NumberOfAddress  OrderAmountHikeFromlastYear
\
count        5630.000000      5630.000000                  5365.000000

mean            3.066785         4.214032                    15.707922

std             1.380194         2.583586                     3.675485

min             1.000000         1.000000                    11.000000

25%             2.000000         2.000000                    13.000000

50%             3.000000         3.000000                    15.000000

75%             4.000000         6.000000                    18.000000

max             5.000000        22.000000                    26.000000


        CouponUsed   OrderCount  DaySinceLastOrder  CashbackAmount
count  5374.000000  5372.000000        5323.000000     5630.000000
mean      1.751023     3.008004           4.543491      177.223030
std       1.894621     2.939680           3.654433       49.207036
min       0.000000     1.000000           0.000000        0.000000
25%       1.000000     1.000000           2.000000      145.770000
50%       1.000000     2.000000           3.000000      163.280000
```

```
75%       2.000000     3.000000             7.000000      196.392500
max      16.000000    16.000000            46.000000      324.990000
```

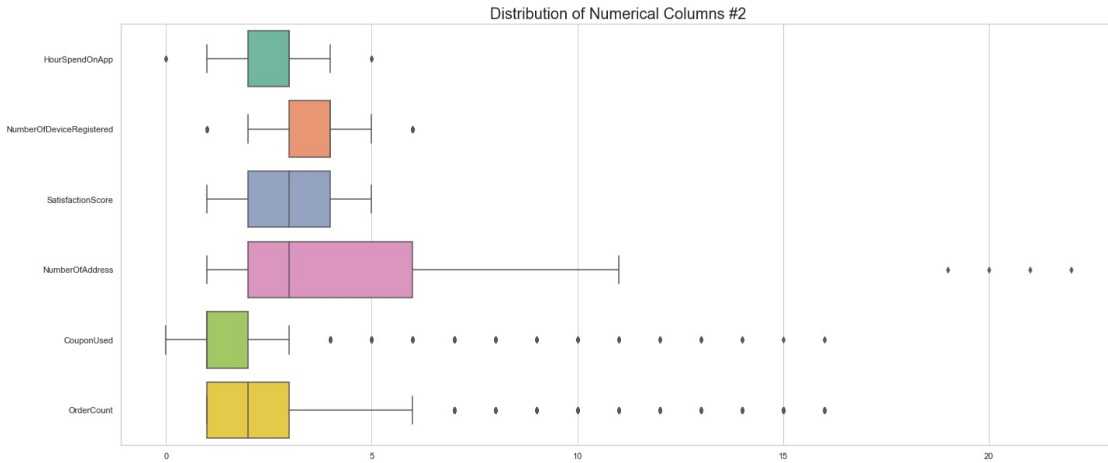
with this feature, We Know the count, mean, percentile, etc . for the example, the total count of the dataset of each column is +-5630 and has various standard deviation.

The next step is I want to know the glimpse of distribution of the numerical columns using seaborn boxplot. actually this will be done in Step 2 but We'll do it now anyway.

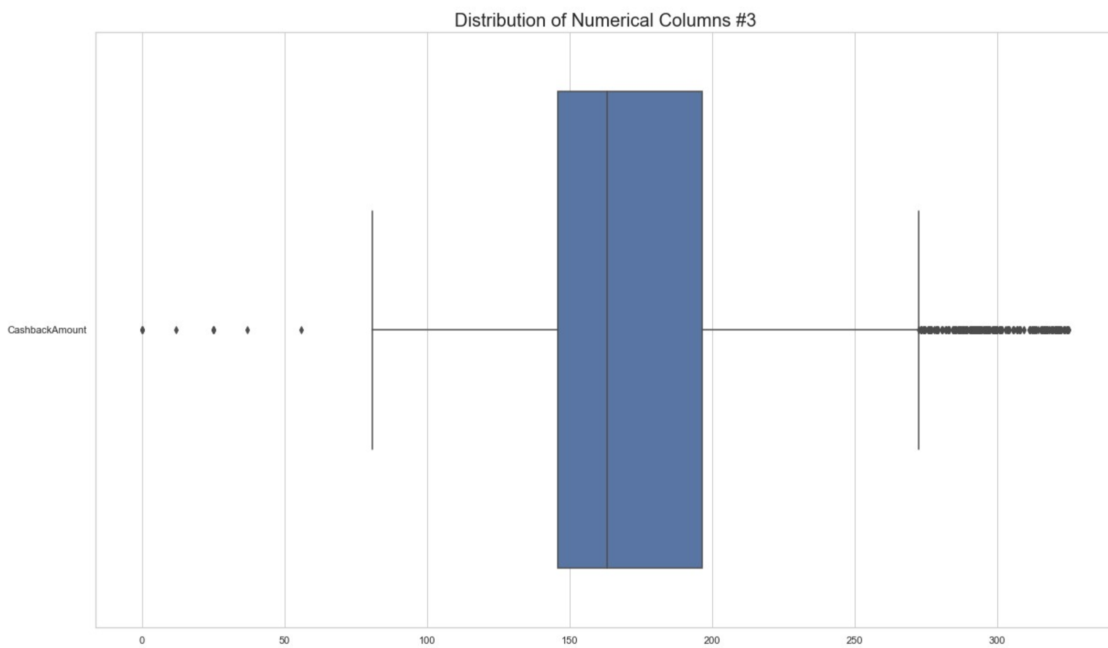
```
df1 = df_main_num[['Tenure', 'WarehouseToHome',
'OrderAmountHikeFromlastYear', 'DaySinceLastOrder']]
df2 = df_main_num[['HourSpendOnApp', 'NumberOfDeviceRegistered',
'SatisfactionScore', 'NumberOfAddress', 'CouponUsed'
                        , 'OrderCount']]
df3 = df_main_num[['CashbackAmount']]

sns.set_theme(style="whitegrid")
plt.figure(figsize = (23,10))
p = sns.boxplot(data=df1,orient="h", palette = 'Set1')
plt.title('Distribution of Numerical Columns #1', fontsize = 20)
plt.show()
```




```
sns.set_theme(style="whitegrid")
plt.figure(figsize = (23,10))
p = sns.boxplot(data=df2, orient="h", palette = 'Set2')
plt.title('Distribution of Numerical Columns #2', fontsize = 20)
plt.show()
```




```
plt.figure(figsize = (20,12))
sns.set_theme(style="whitegrid")
p = sns.boxplot(data=df3, orient = 'h')
plt.title('Distribution of Numerical Columns #3', fontsize = 20)
plt.show()
```



Observations

Questions

- Apakah ada kolom dengan tipe data kurang sesuai, atau nama kolom dan isinya kurang sesuai?
- Apakah ada kolom yang memiliki nilai kosong? Jika ada, apa saja?
- Apakah ada kolom yang memiliki nilai summary agak aneh? (min,mean, median, max, unique, top, freq)

Answers

1. No, there is no unmatch type between the data type and the column name, and all the values are matched.
2. Yes, there are many columns that have missing values, which are Tenure, Warehouse to Home, Hours Spend on app,OrderAmountHikeFromlastYear, CouponUsed , OrderCount, DaySinceLastOrder and the proportion are between 4%-5% of all values in the dataset
3. We think that all summary values is not proper yet to build the model in it because there are columns that have missing values and outliers, for the next step if we want to build the model we should do some treatment to it.

Step 2. Univariate Analysis

In this step, we will use Seaborn for visualize the dataset. we will gain many insights from this tool. but before we start, What is Seaborn, actually?

Part 1 : Seaborn

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data.(source = http://seaborn.pydata.org/introduction.html)

Part 2 : Univariate Analysis

Univariate analysis is the simplest form of analyzing data. “Uni” means “one”, so in other words your data has only one variable. It doesn't deal with causes or relationships (unlike regression) and it's major purpose is to describe; It takes data, summarizes that data and finds patterns in the data.

```
df_main.head()
```

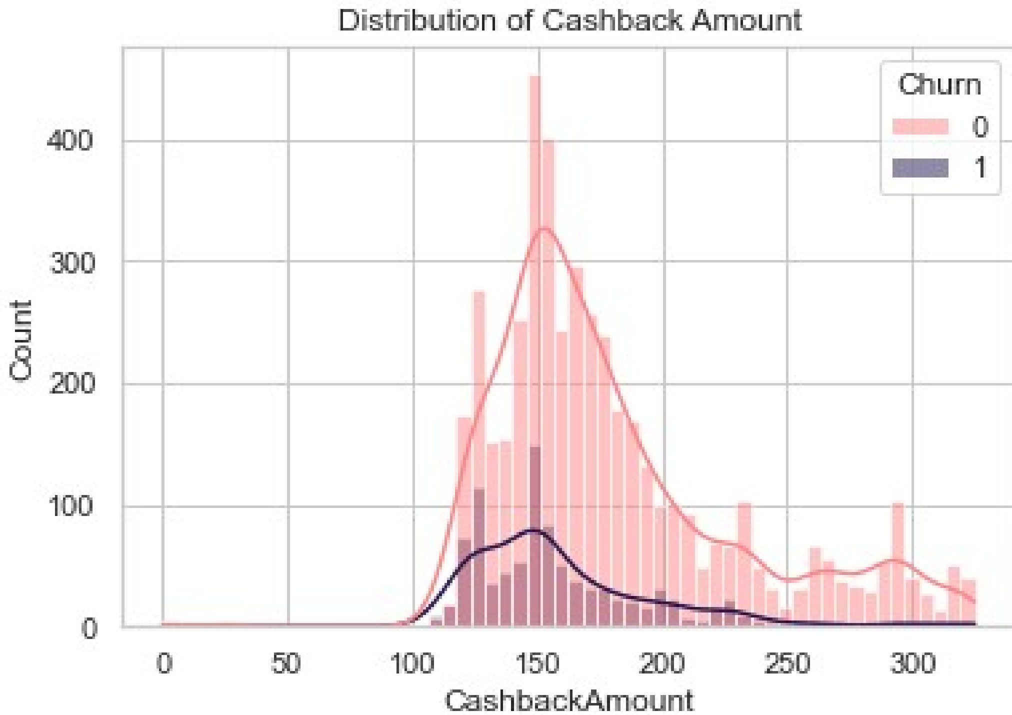
```
   CustomerID  Churn  Tenure PreferredLoginDevice  CityTier  
WarehouseToHome  \
0       50001      1     4.0         Mobile Phone         3  
6.0   
1       50002      1     NaN                Phone         1  
8.0   
2       50003      1     NaN                Phone         1  
30.0   
3       50004      1     0.0                Phone         3  
15.0   
4       50005      1     0.0                Phone         1  
12.0   

  PreferredPaymentMode  Gender  HourSpendOnApp  
NumberOfDeviceRegistered  \
0          Debit Card  Female             3.0  
3
```

```
1                   UPI    Male             3.0
4
2            Debit Card    Male             2.0
4
3            Debit Card    Male             2.0
4
4                    CC    Male             NaN
3

     PreferedOrderCat  SatisfactionScore MaritalStatus
NumberOfAddress  \
0  Laptop & Accessory                  2        Single
9
1              Mobile                  3        Single
7
2              Mobile                  3        Single
6
3  Laptop & Accessory                  5        Single
8
4              Mobile                  5        Single
3

   Complain  OrderAmountHikeFromlastYear  CouponUsed  OrderCount  \
0         1                         11.0         1.0         1.0
1         1                         15.0         0.0         1.0
2         1                         14.0         0.0         1.0
3         0                         23.0         0.0         1.0
4         0                         11.0         1.0         1.0

   DaySinceLastOrder  CashbackAmount
0                5.0          159.93
1                0.0          120.90
2                3.0          120.28
3                3.0          134.07
4                3.0          129.60
```

Part 3 : Histogram

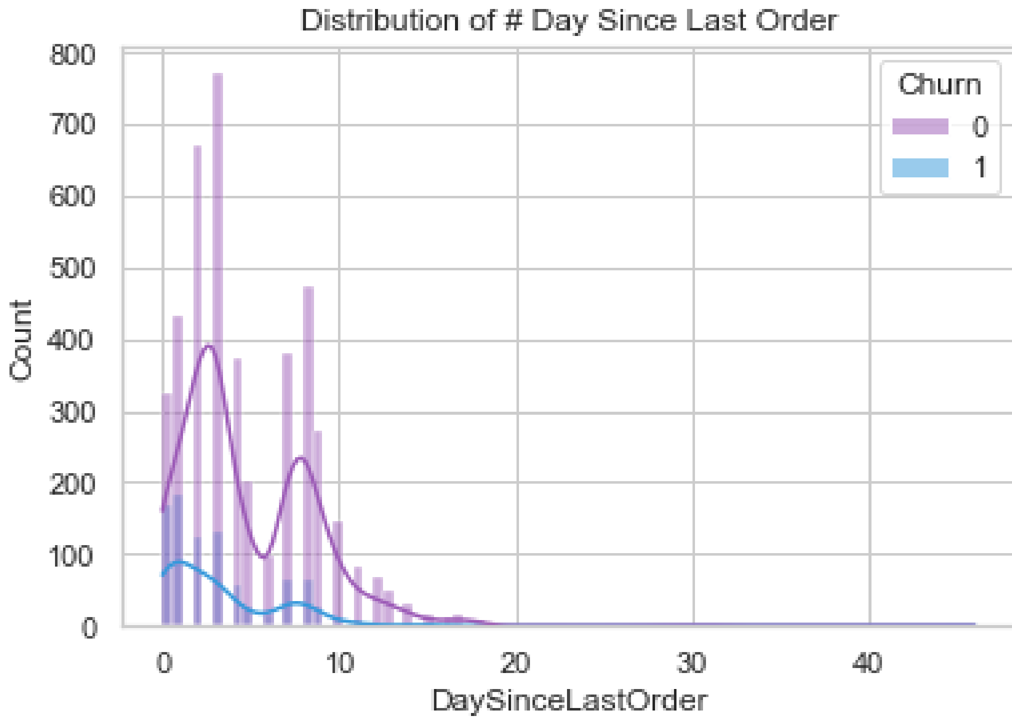
```
flatui = ['#FF8484', '#231651']
sns.histplot(data=df_main, x="CashbackAmount", hue="Churn", kde =
True, palette = flatui)
plt.title('Distribution of Cashback Amount')
```

```
Text(0.5, 1.0, 'Distribution of Cashback Amount')
```

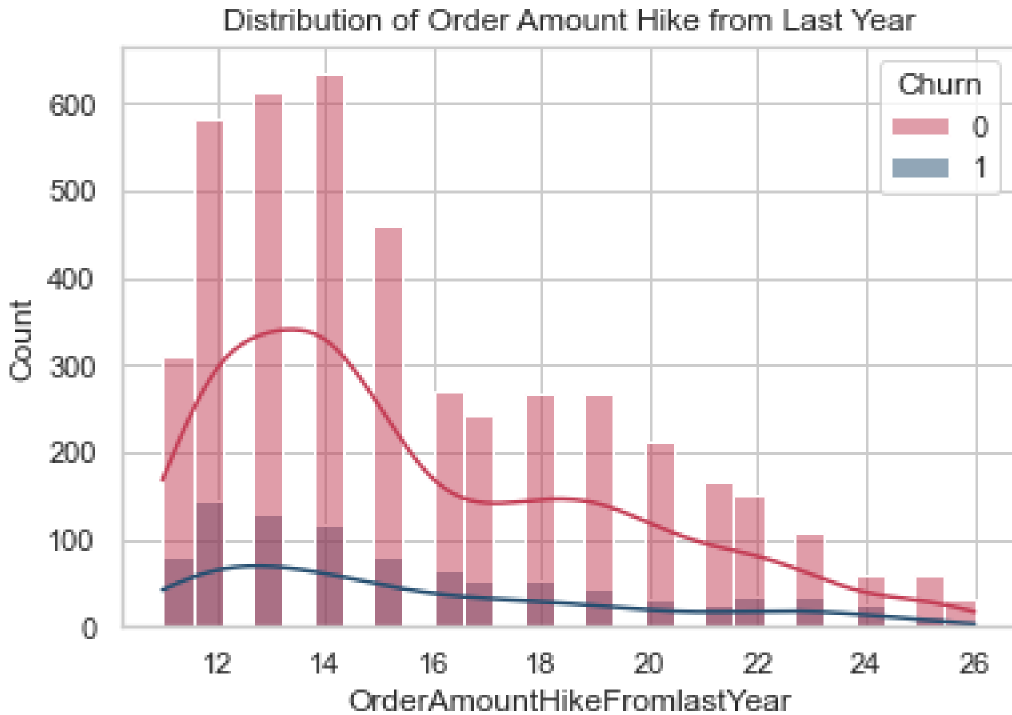



```
flatui = ["#9b59b6", "#3498db"]
sns.histplot(data=df_main, x="DaySinceLastOrder", hue = 'Churn', kde = True, palette = flatui)
plt.title('Distribution of # Day Since Last Order')
```

```
Text(0.5, 1.0, 'Distribution of # Day Since Last Order')
```

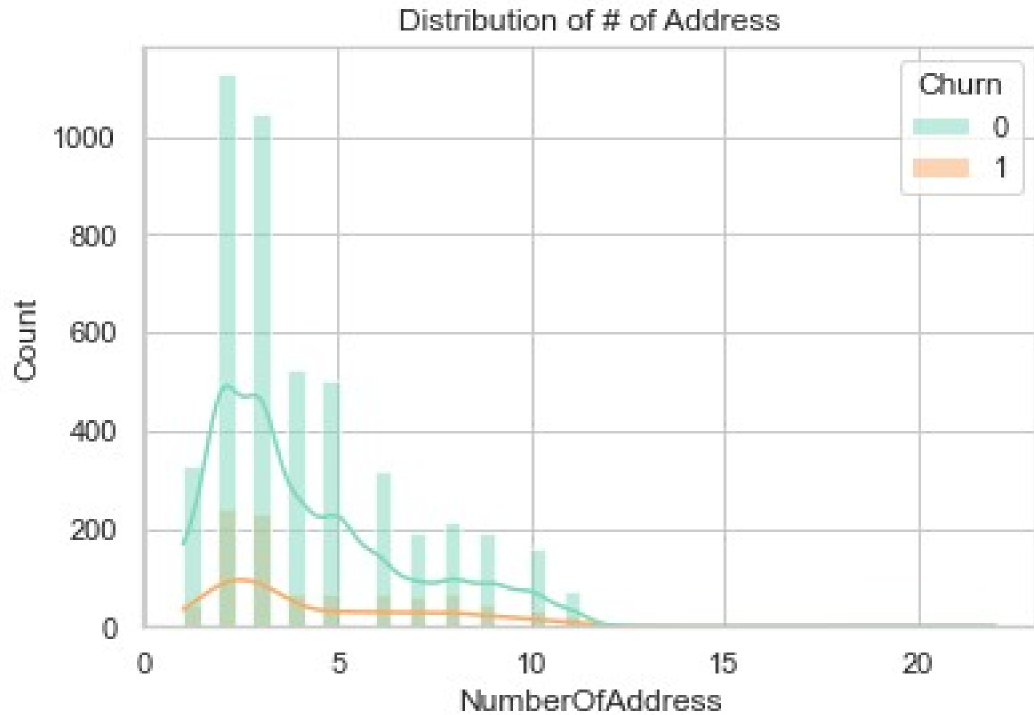



```
flatui = ["#C33C54", "#254E70"]
sns.histplot(data=df_main, x="OrderAmountHikeFromlastYear", hue =
'Churn', kde = True, palette = flatui)
plt.title('Distribution of Order Amount Hike from Last Year')

Text(0.5, 1.0, 'Distribution of Order Amount Hike from Last Year')
```



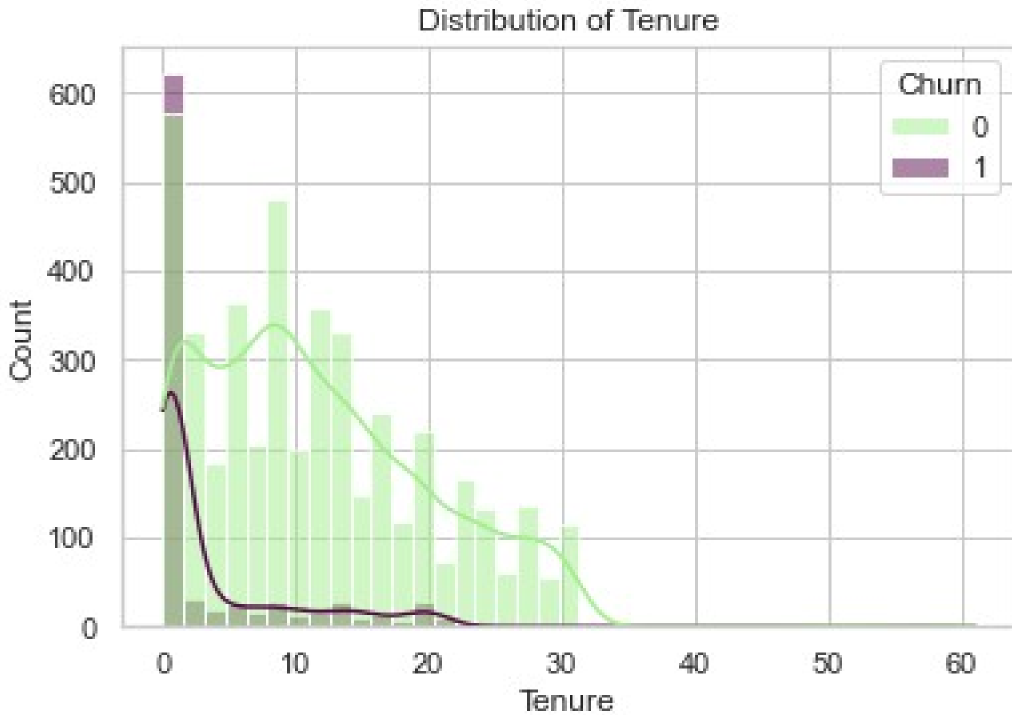

```
flatui = ["#7FD8BE", "#FCAB64"]
sns.histplot(data=df_main, x="NumberOfAddress",hue = 'Churn', kde =
True, palette = flatui)
plt.title('Distribution of # of Address')
```

```
Text(0.5, 1.0, 'Distribution of # of Address')
```




```
flatui = ["#A1EF8B", "#5A0B4D"]
sns.histplot(data=df_main, x="Tenure",hue = 'Churn', kde = True,
palette = flatui)
plt.title('Distribution of Tenure')
```

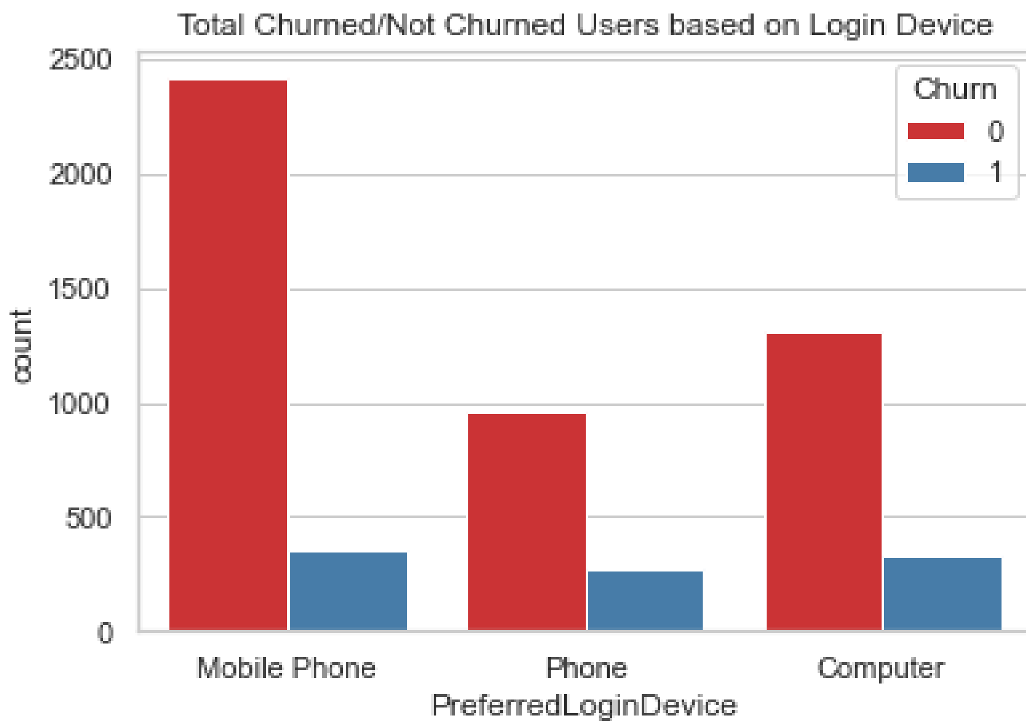
```
Text(0.5, 1.0, 'Distribution of Tenure')
```



Part 4 : Count Plot

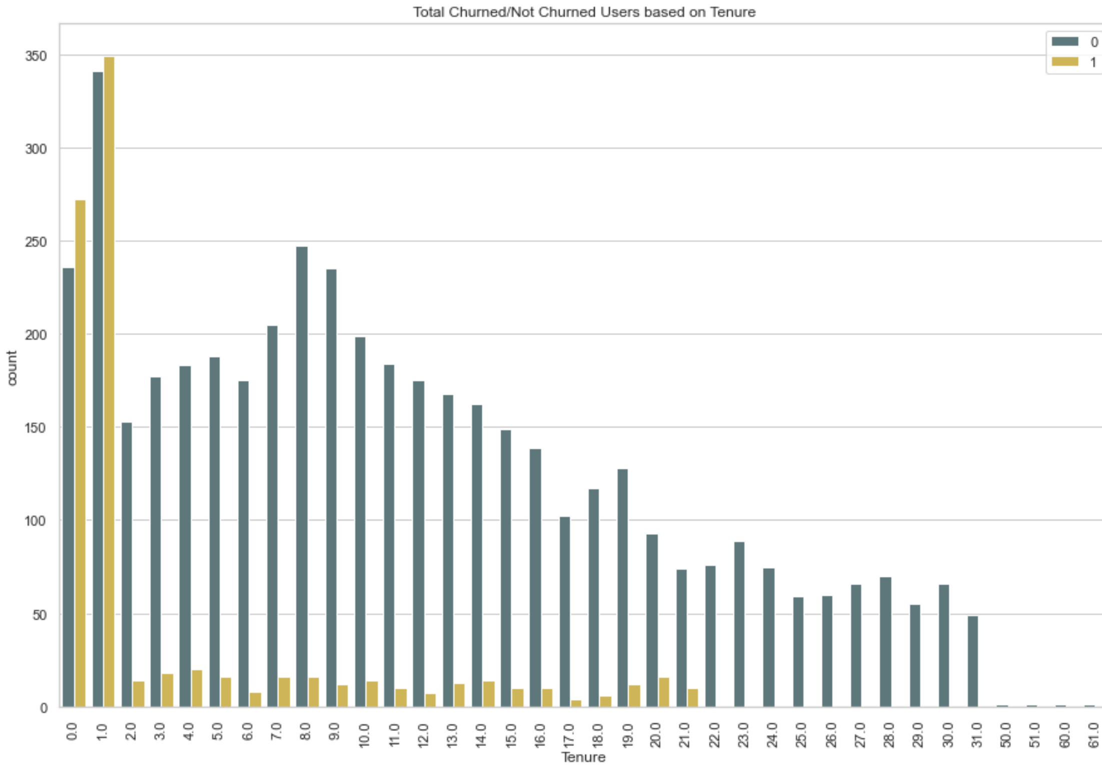
```
ax = sns.countplot(x="PreferredLoginDevice", hue="Churn", data=
df_main, palette = 'Set1')
plt.title('Total Churned/Not Churned Users based on Login Device')
```

```
Text(0.5, 1.0, 'Total Churned/Not Churned Users based on Login
Device')
```



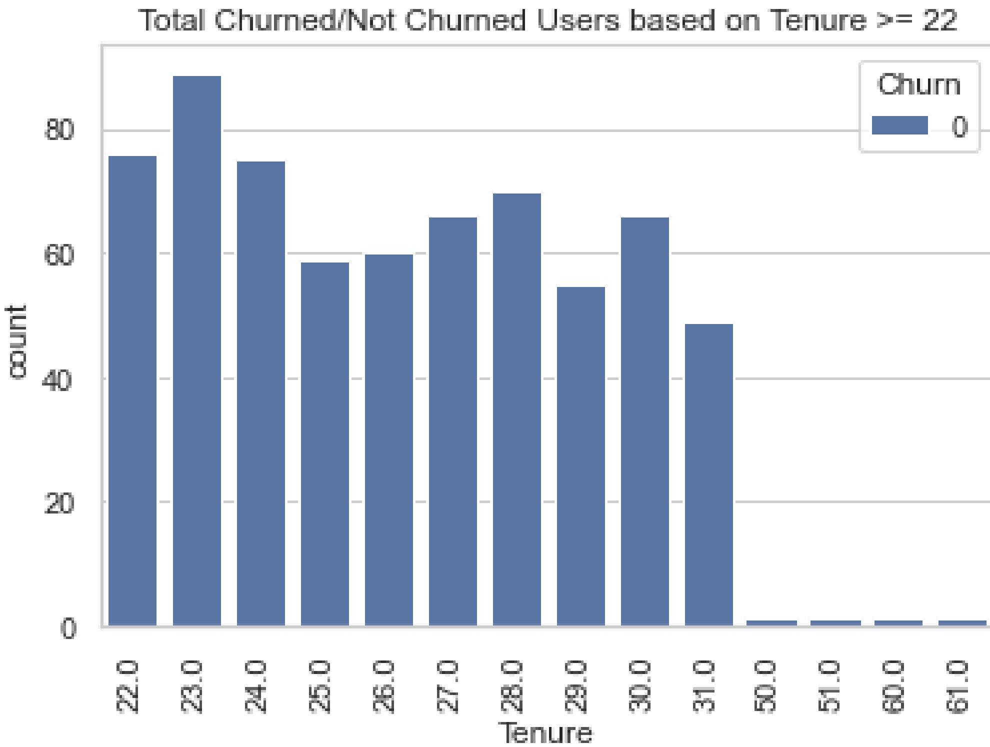

```
paletui = ['#587B7F', '#E2C044']
plt.figure(figsize = (15,10))
ax = sns.countplot(x="Tenure", hue="Churn", data= df_main, palette =
paletui)
ax.tick_params(axis='x', rotation=90)
plt.legend(loc='upper right')
plt.title('Total Churned/Not Churned Users based on Tenure')
```

```
Text(0.5, 1.0, 'Total Churned/Not Churned Users based on Tenure')
```



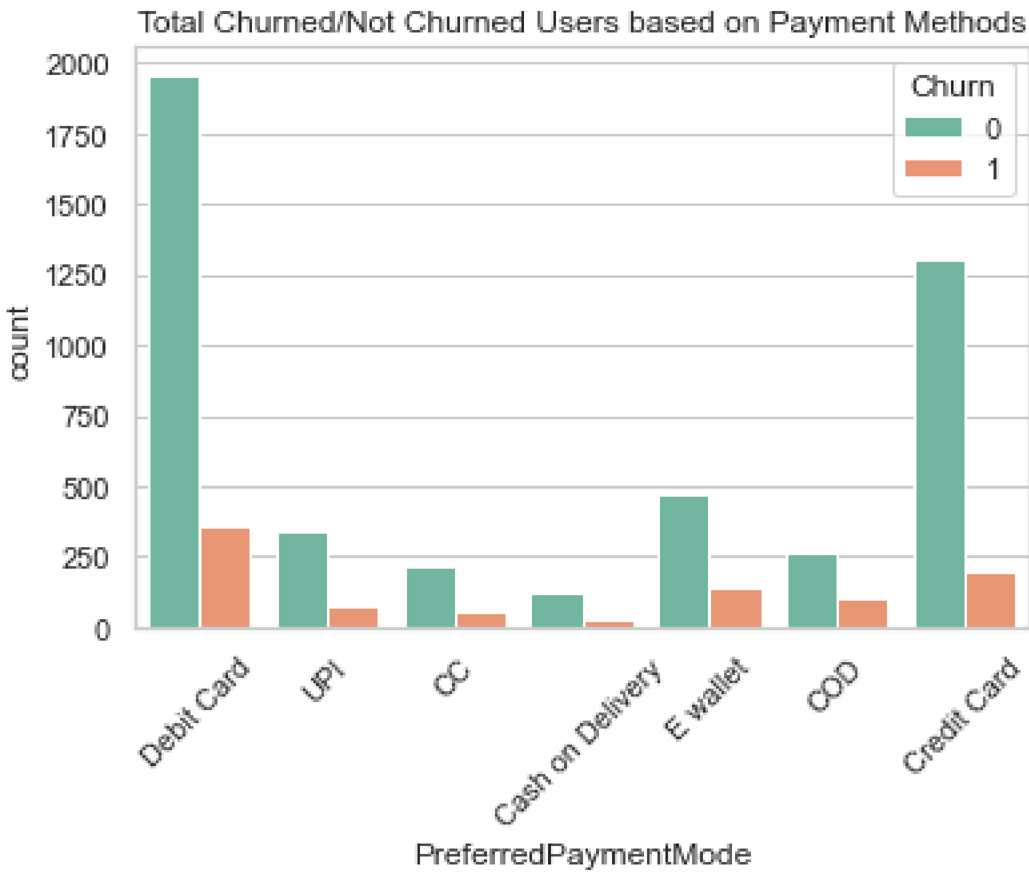

```
df_tenure_22 = df_main[df_main['Tenure']>=22]
ax = sns.countplot(x = 'Tenure', hue = 'Churn', data = df_tenure_22)
ax.tick_params(axis='x', rotation=90)
plt.title('Total Churned/Not Churned Users based on Tenure >= 22')
```

```
Text(0.5, 1.0, 'Total Churned/Not Churned Users based on Tenure >=
22')
```



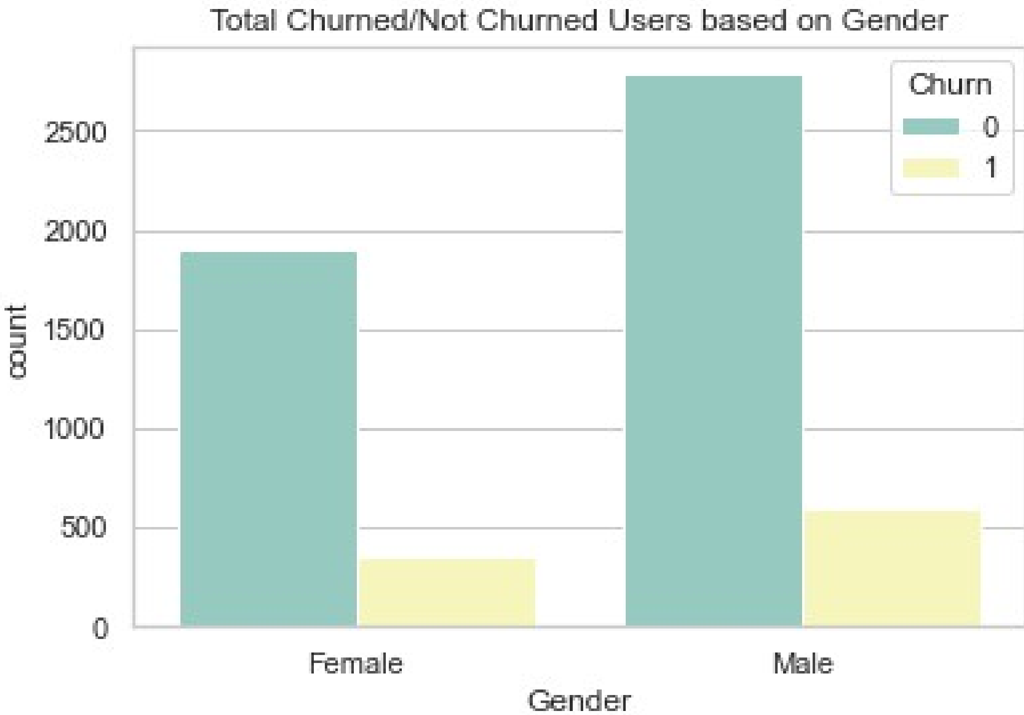

```
p =sns.countplot(x="PreferredPaymentMode", hue="Churn", data= df_main,
palette = 'Set2')
p.tick_params(axis='x', rotation=45)
plt.title('Total Churned/Not Churned Users based on Payment Methods')
```

```
Text(0.5, 1.0, 'Total Churned/Not Churned Users based on Payment
Methods')
```



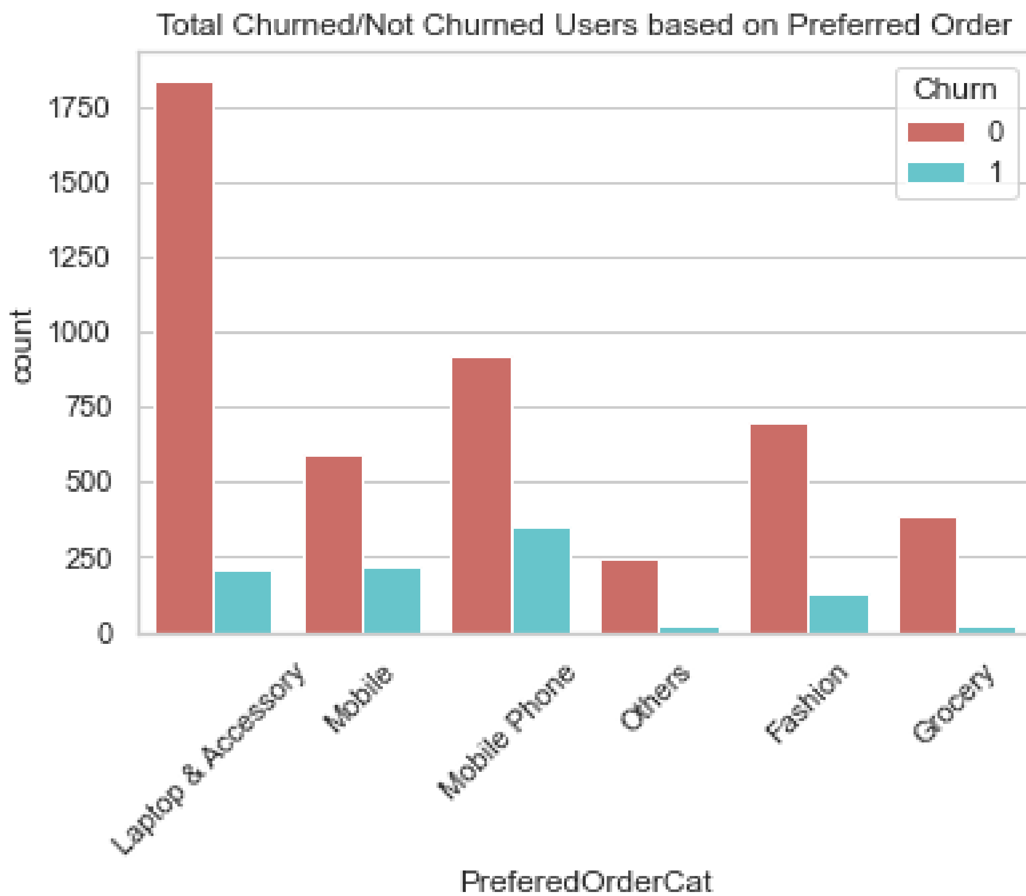

```
sns.countplot(x="Gender", hue="Churn", data= df_main, palette =
'Set3')
plt.title('Total Churned/Not Churned Users based on Gender')
```

```
Text(0.5, 1.0, 'Total Churned/Not Churned Users based on Gender')
```



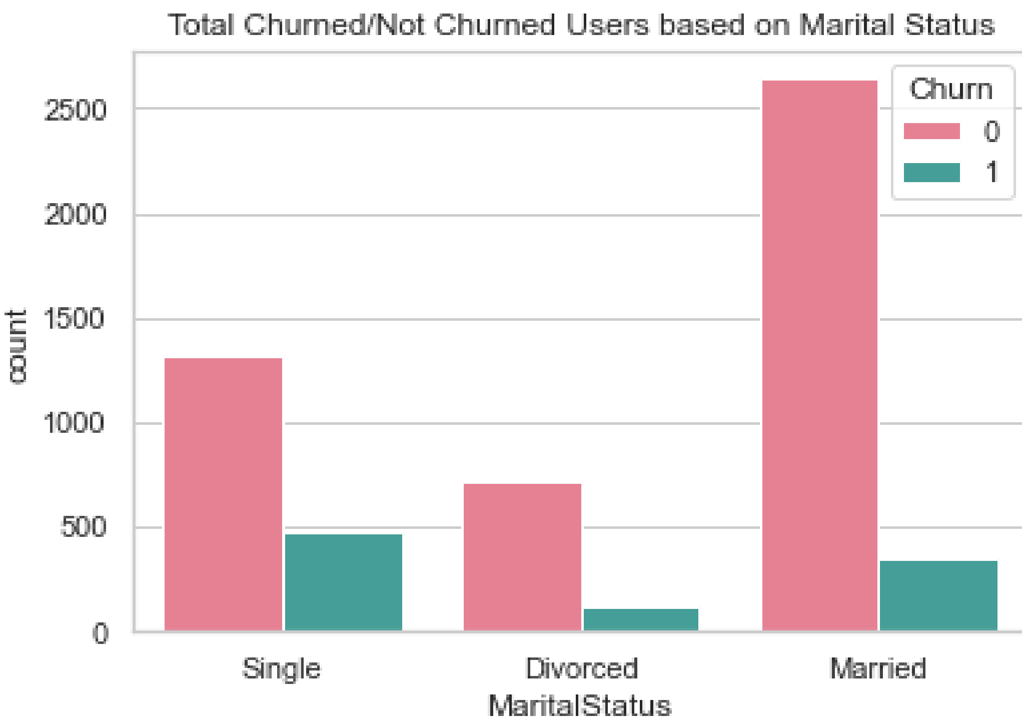

```
ax =sns.countplot(x="PreferedOrderCat", hue="Churn", data= df_main,
palette = 'hls')
ax.tick_params(axis='x', rotation=45)
plt.title('Total Churned/Not Churned Users based on Preferred Order')
```

```
Text(0.5, 1.0, 'Total Churned/Not Churned Users based on Preferred
Order')
```




```
ax =sns.countplot(x="MaritalStatus", hue="Churn", data= df_main,
palette = 'husl')
plt.title('Total Churned/Not Churned Users based on Marital Status')
```

```
Text(0.5, 1.0, 'Total Churned/Not Churned Users based on Marital
Status')
```



Observations

Question

Gunakan visualisasi untuk melihat distribusi masing-masing kolom (feature maupun target). Tuliskan hasil observasinya, misalnya jika ada suatu kolom yang distribusinya menarik (misal skewed, bimodal, ada outlier, ada nilai yang mendominasi, kategorinya terlalu banyak, dsb). Jelaskan juga apa yang harus di-follow up saat data pre-processing.

Answers

Mean The distribution column is positive skewed, for example Power Since Last Order, Order Amount Hike from Last Year, Number of Address, and Tenure. Meanwhile, the Cashback Column has an almost normal distribution. For Outliers, it can be seen in the plot below, namely the box plot that there are many outliers in the tenure section where the marital status is single and the preferred login device is a mobile phone.

What must be followed up during data preprocessing:

- Determine the Threshold for outliers, whether you want to be discarded or left with a certain threshold
- Fill in Missing Values, either by imputation or other methods
- Perform one hot encoding or use pd.get_dummies() to handle features of type string
- if necessary, normalization of features is required before entering modeling

Step 3. Multivariate Analysis

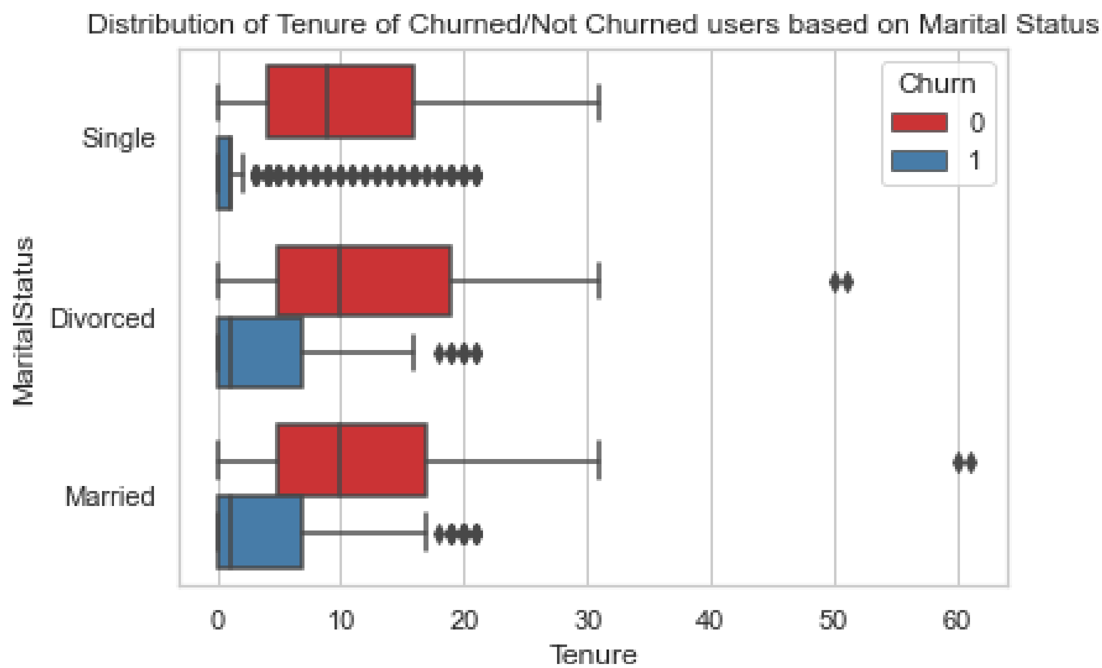
Part 1 : Multivariate Analysis

Multivariate analysis is conceptualized by tradition as the statistical study of experiments in which multiple measurements are made on each experimental unit and for which the relationship among multivariate measurements and their structure are important to the experiment's understanding.

Part 2 : Box Plot

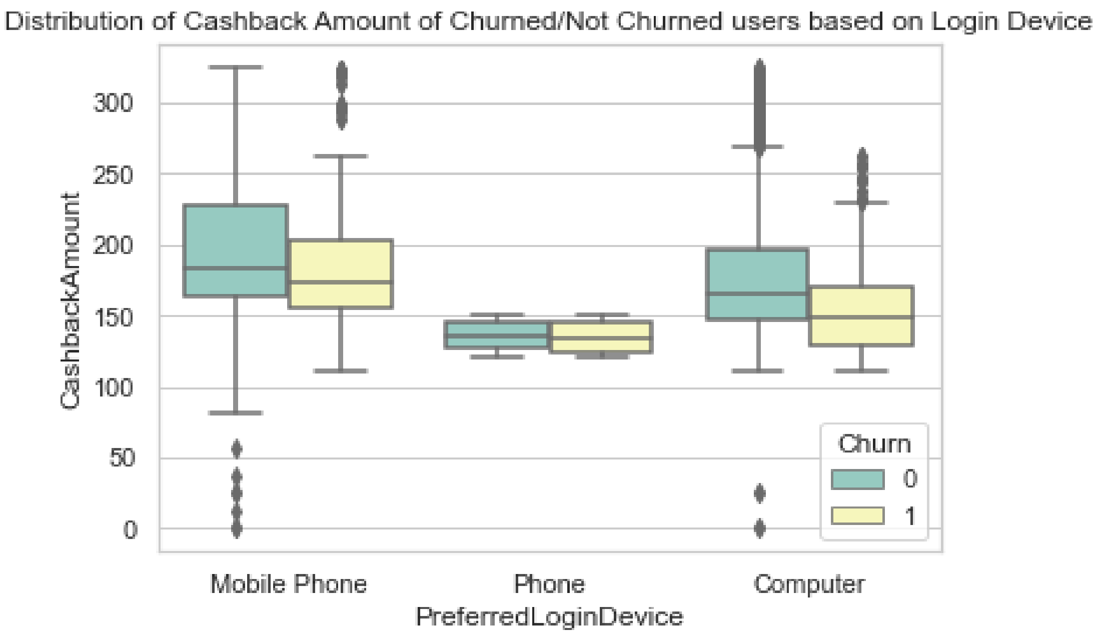
```
ax = sns.boxplot(x="Tenure", y="MaritalStatus", hue="Churn",
                 data=df_main, palette="Set1")
ax
plt.title('Distribution of Tenure of Churned/Not Churned users based
on Marital Status')
```

```
Text(0.5, 1.0, 'Distribution of Tenure of Churned/Not Churned users
based on Marital Status')
```




```
ax = sns.boxplot(x="PreferredLoginDevice", y="CashbackAmount",
hue="Churn",
                 data=df_main, palette="Set3")
ax
plt.title('Distribution of Cashback Amount of Churned/Not Churned
users based on Login Device')
```

```
Text(0.5, 1.0, 'Distribution of Cashback Amount of Churned/Not Churned
users based on Login Device')
```

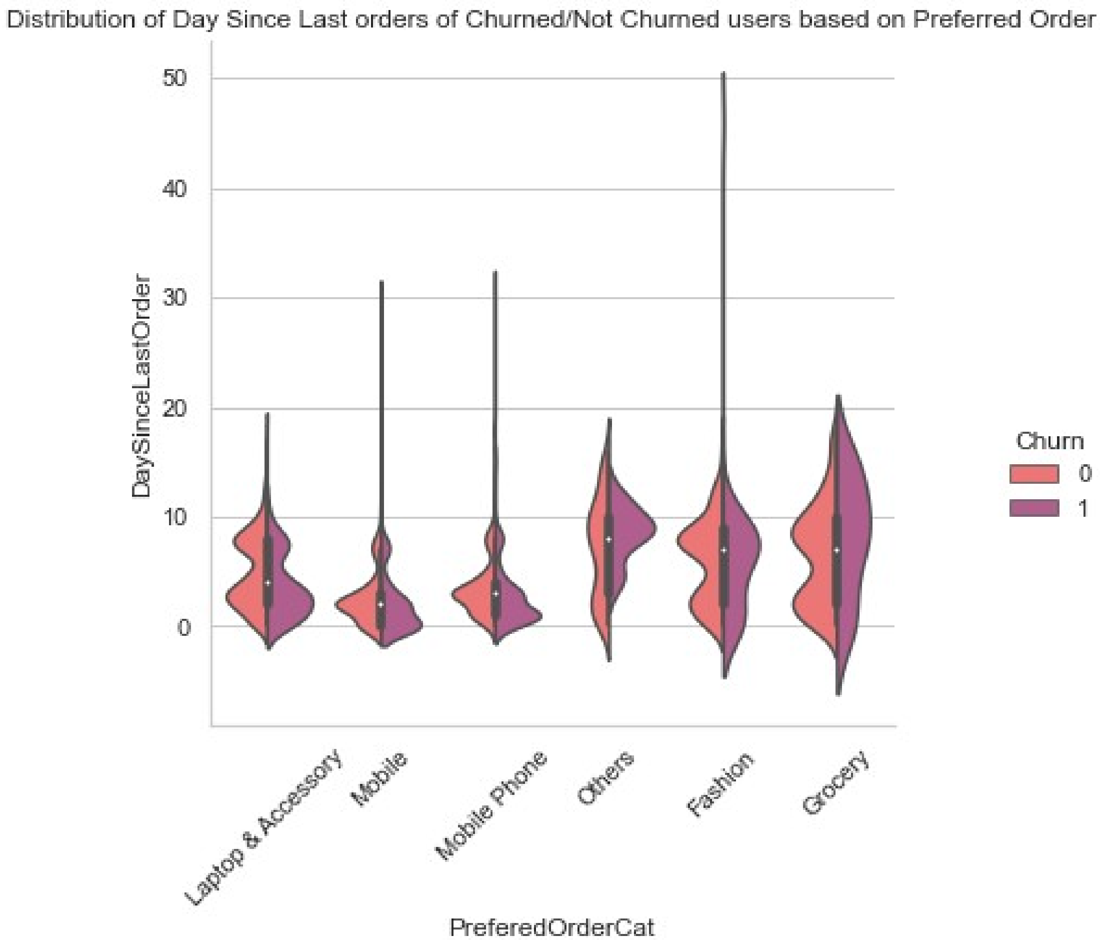


Part 3 : Violin Plot

```
flatui = ["#ff6361", "#bc5090"]
plt.figure(figsize = (25,25))
p = sns.catplot(x="PreferedOrderCat", y="DaySinceLastOrder", hue="Churn",
            kind="violin", split=True, data=df_main, palette = flatui)
p.set_xticklabels(rotation = 45)
plt.title('Distribution of Day Since Last orders of Churned/Not Churned users based on Preferred Order')
```

```
Text(0.5, 1.0, 'Distribution of Day Since Last orders of Churned/Not Churned users based on Preferred Order')
```

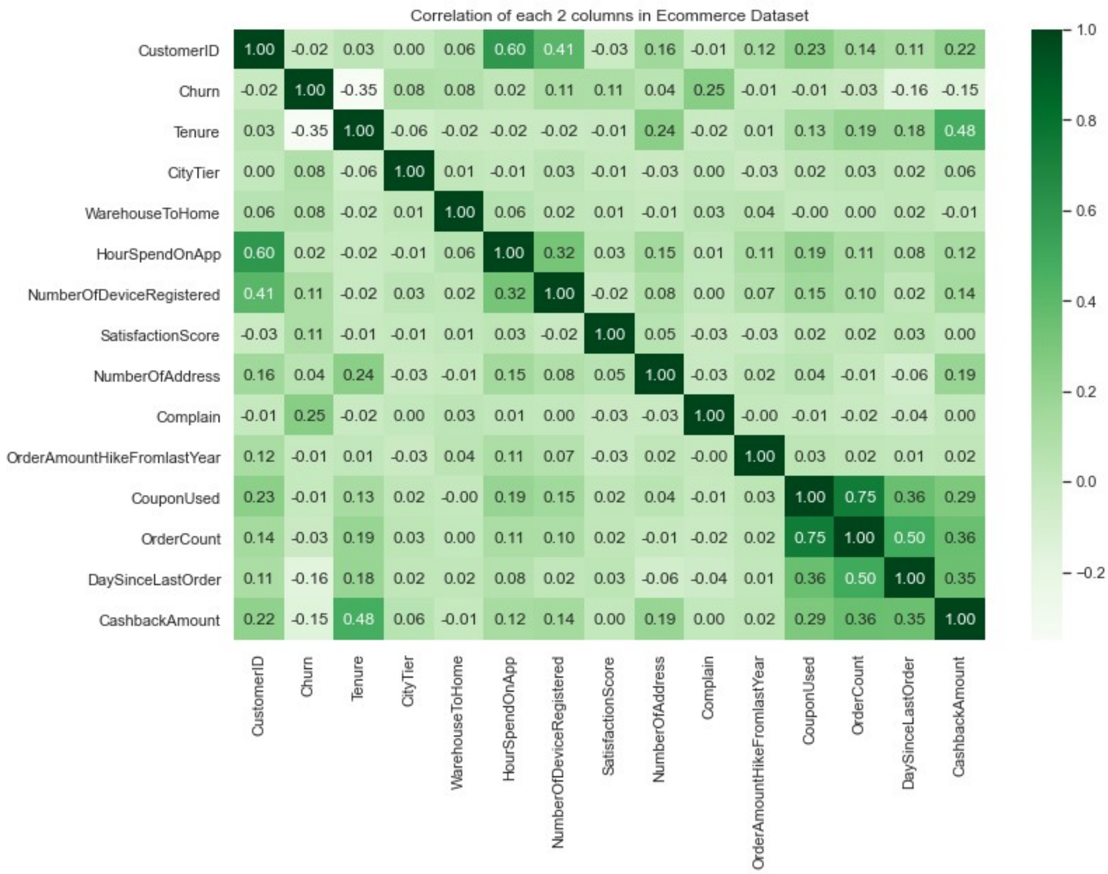
```
<Figure size 1800x1800 with 0 Axes>
```



Part 4 : Heatmap

```
plt.figure(figsize=(12,8))
sns.heatmap(df_main.corr(), cmap = 'Greens', annot = True, fmt =
'.2f')
plt.title('Correlation of each 2 columns in Ecommerce Dataset')
```

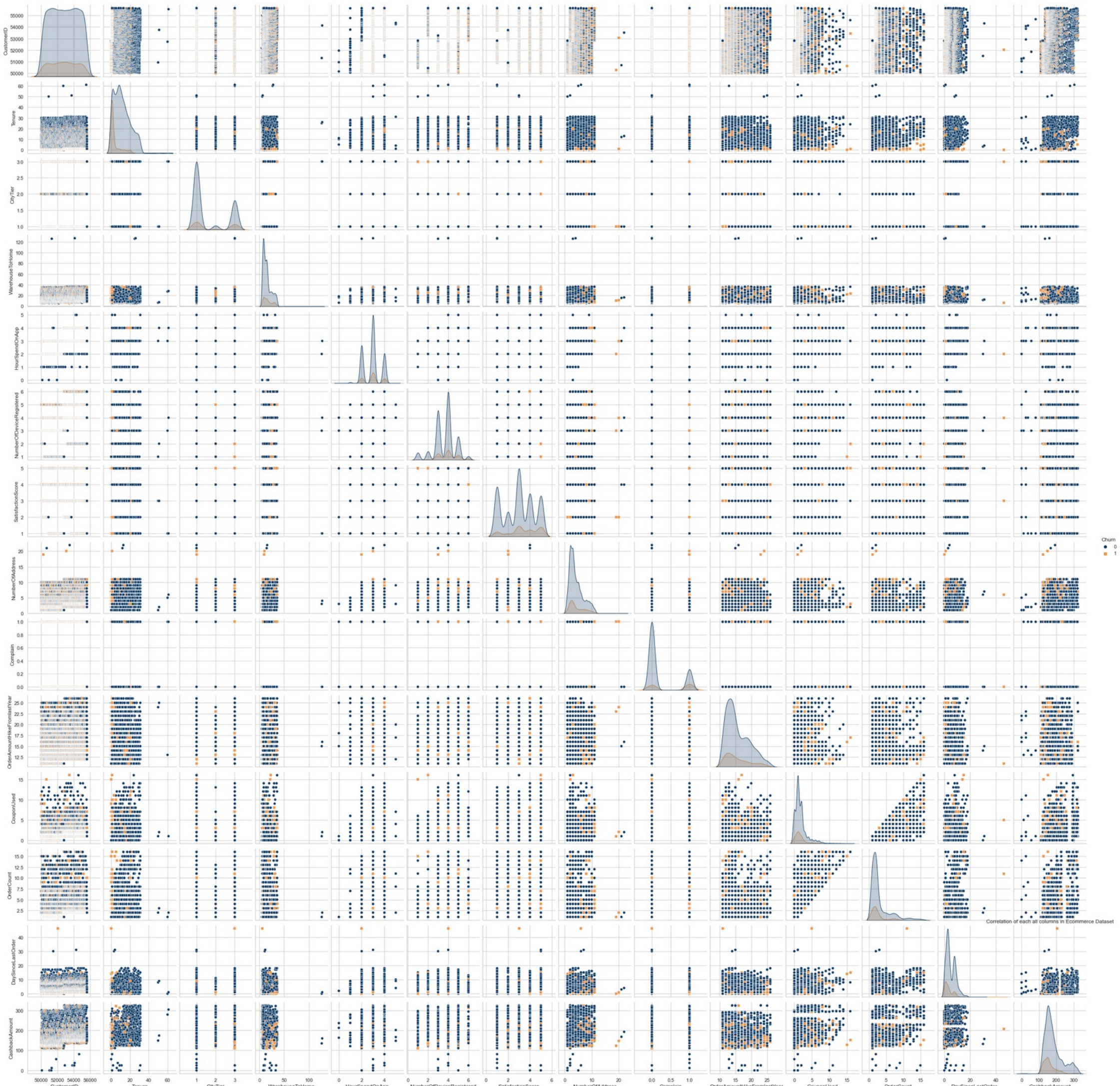
```
Text(0.5, 1.0, 'Correlation of each 2 columns in Ecommerce Dataset')
```



Part 5 : Pairplot

```
sns.pairplot(df_main, hue="Churn", markers=["o", "s"], palette =
['#0d3b66', '#ee964b'])
plt.title('Correlation of each all columns in Ecommerce Dataset')
```

```
Text(0.5, 1.0, 'Correlation of each all columns in Ecommerce Dataset')
```



Observations

Question

Lakukan multivariate analysis (seperti correlation heatmap dan category plots, sesuai yang diajarkan di kelas). Tuliskan hasil observasinya, seperti:

- Bagaimana korelasi antara masing-masing feature dan label. Kira-kira feature mana saja yang paling relevan dan harus dipertahankan?
- Bagaimana korelasi antar-feature, apakah ada pola yang menarik? Apa yang perlu dilakukan terhadap feature itu?

note : Tuliskan juga jika memang tidak ada feature yang saling berkorelasi

Answers

- The column that must be maintained is of course the one that has a correlation value = 1, and there are several columns that have a good correlation such as the correlation between coupon used and order count, Order count and day since last order, hours spend on app and Number of device registered, as well as Tenure and Cashback Amount.

- It is true that there are some interesting patterns, most of which arise from the relationship between numerical columns such as tenure and cashback, what needs to be done for preprocessing purposes is to check outliers and normalize to produce a robust model.

Step 4 : Business Insights

Observations

Question

Selain EDA, lakukan juga beberapa analisis dan visualisasi untuk menemukan suatu business insight. Tuliskan minimal 3 insight, dan berdasarkan insight tersebut jelaskan rekomendasinya untuk bisnis.

Answers

- Single men who use applications using mobile phones tend to be more likely to churn
- The amount of cashback is very influential on the churn rate. the more cashback given, the more likely the customer to churn. with the average of 150 - 200 cashback amount in 300 - 400 users, most likely the possibility in churn will decrease.
- Payments using debit card cause the most churn while payments via cash on delivery cause the least churn, but COD has a higher proportion between churn and non churn users than debit card. the ratio in Debit card between non churned and churned users is +- 1900 : 260 and the COD is +- 250:150
- based on the correlation heatmap, the value of the complaint has the largest positive correlation to churn, but the largest correlation to churn is tenure but the correlation is negative
- There are 2 factors that influence customers to Churn in terms of customer satisfaction, namely, the first in terms of payment methods, where if accumulated in total, customers with payment methods using Debit Cards have the largest churn quantity. However, if only grouped by type of payment method, customers with COD payment methods have the greatest churn potential, which is 28%. So that a review and improvement on the COD payment system is needed. In addition to payment methods, the churn rate for customers who submit complaints is quite high, above 31%.
- The churn trend often occurs in new customers, where customers churn with a ratio above 50% for the data value of Tenure <=1.
- The more devices registered on one Customer ID, the higher the potential for churn.
- As input, improvement in customer service is needed, because many customers who have just made orders, churn.
- There is a possibility that customers are dissatisfied with the cashback program provided, so they churn.

- There is no Churned Users when the tenure is equal or above 22