



CPE 232 Data Models Final Project

จัดทำโดย

65070501048 รัชชิพงษ์ สกุลจีน

65070501072 ชนกันต์ จิตรกสิกร

65070501074 ณภัทร สินจินดาววงศ์

65070501075 ณัฐชนน บุญยะโท

65070501084 ภูมิรพี เมืองน้อย

เสนอ

ผศ.ดร.สันติธรรม พรหมอ่อน

รายงานนี้เป็นส่วนหนึ่งของรายวิชา Data Models (CPE 232)

ภาคเรียนที่ 2 ปีการศึกษา 2566

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

Introduction

รายงานนี้เป็นส่วนหนึ่งของรายวิชา Data Models (CPE 232) โดยเป็นการจัดทำ EDA, Visualization และ Modeling โดยใช้ชุดข้อมูล Ecommerce Customer Churn จำนวน 5639 例 在ในการทำงาน

Data explanation

ข้อมูลของบริษัท Online E-commerce เป็นข้อมูลที่บอกถึงพฤติกรรมการซื้อสินค้าออนไลน์ของลูกค้า และประวัติการซื้อของเหล่านั้น โดยข้อมูลมีทั้งหมด 5,630 rows และ 20 columns ซึ่งได้แก่

- CustomerID : ID ของลูกค้าแต่ละคน
- Churn : เลขที่บอกถึงการ Churn ของลูกค้าโดย 0 หมายถึงลูกค้ายังเป็นสมาชิกของบริษัทอยู่ และ 1 หมายถึงลูกค้ายกเลิกการเป็นสมาชิกของบริษัทไปแล้ว
- Tenure : ระยะเวลาที่ลูกค้าอยู่กับองค์กรนี้
- PreferredLoginDevice : อุปกรณ์ที่ลูกค้าเลือกใช้ในการ Login
- CityTier : ระดับของพุทธิกรรมในการบริโภคของประชากรในเมืองนั้น โดยจะมีตั้งแต่ระดับ 1 ไปจนถึง ระดับ 3
- WarehouseToHome : ระยะห่างระหว่างที่เก็บสินค้ากับบ้านลูกค้า
- PreferredPaymentMode : วิธีที่ลูกค้าเลือกใช้ในการชำระเงิน
- Gender : เพศของลูกค้า
- HourSpendOnApp : จำนวนชั่วโมงที่ลูกค้าใช้งานแอปบนโทรศัพท์หรือในเว็บไซต์
- NumberOfDeviceRegistered : จำนวนอุปกรณ์ทั้งหมดที่ลูกค้าลงทะเบียนไว้
- PreferredOrderCat : หมวดหมู่สินค้าที่ลูกค้าซื้อบ่อยในเดือนก่อนหน้า
- SatisfactionScore : ความพึงพอใจของลูกค้า
- MaritalStatus : สถานภาพการสมรสของลูกค้า
- NumberOfAddress : จำนวนที่ร่วมอยู่ของลูกค้า
- Complain : ลูกค้าได้มีการ complain หรือไม่ในเดือนก่อนหน้าโดย 0 หมายถึง ไม่มีการ complain เลยในเดือนก่อนหน้า 1 หมายถึงมีการ complain ในเดือนก่อนหน้า

- OrderAmountHikeFromlastYear : จำนวน order รวมที่ลูกค้ารายนี้สั่งเพิ่มขึ้นจากปีที่แล้วคิดเป็นเปอร์เซ็นต์
- CouponUsed : จำนวนคูปองทั้งหมดที่ลูกค้ารายนี้ใช้ในเดือนก่อนหน้า
- OrderCount : จำนวน order ทั้งหมดที่ลูกค้าสั่งในเดือนก่อนหน้า
- DaySinceLastOrder : จำนวนวันที่ผ่านไปนับตั้งแต่ลูกค้าสั่งซื้อครั้งล่าสุด
- CashbackAmount : จำนวนเงินเฉลี่ยที่ลูกค้าได้รับคืนในเดือนที่แล้ว

Data preparation process and results

ในการเตรียมข้อมูล หรือการ clean data จะต้องมีการทำความเข้าใจข้อมูลก่อนว่ามีความหมายว่าอย่างไร จากนั้นนำข้อมูลมาตรวจสอบแต่ละคอลัมน์เป็นข้อมูลประเภท มีค่าว่างหรือไม่

df.isnull().any()		df.dtypes	
CustomerID	False	CustomerID	int64
Churn	False	Churn	int64
Tenure	True	Tenure	float64
PreferredLoginDevice	False	PreferredLoginDevice	object
CityTier	False	CityTier	int64
WarehouseToHome	True	WarehouseToHome	float64
PreferredPaymentMode	False	PreferredPaymentMode	object
Gender	False	Gender	object
HourSpendOnApp	True	HourSpendOnApp	float64
NumberofDeviceRegistered	False	NumberofDeviceRegistered	int64
PreferedOrderCat	False	PreferedOrderCat	object
SatisfactionScore	False	SatisfactionScore	int64
MaritalStatus	False	MaritalStatus	object
NumberofAddress	False	NumberofAddress	int64
Complain	False	Complain	int64
OrderAmountHikeFromlastYear	True	OrderAmountHikeFromlastYear	float64
CouponUsed	True	CouponUsed	float64
OrderCount	True	OrderCount	float64
DaySinceLastOrder	True	DaySinceLastOrder	float64
CashbackAmount	False	CashbackAmount	float64
			dtype: object

การ clean data มีขั้นตอนดังนี้

- ตรวจสอบคอลัมน์ที่ไม่มีค่า และวิเคราะห์ข้อมูล

df_clean.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

จะเห็นได้ว่า มีคอลัมน์ที่มีค่าว่างซึ่งการจัดการค่าว่างนี้สามารถทำได้โดยการนำค่าเฉลี่ย หรือค่ามัธยฐานมาแทนที่ซึ่งเมื่อเปรียบเทียบค่าเฉลี่ยและค่ามัธยฐานของแต่ละคอลัมน์ที่ไม่มีค่าพบว่ามีค่าที่ใกล้เคียงกันในที่นี้จึงเลือกใช้ค่ามัธยฐานเนื่องจากโดยส่วนใหญ่มีค่าน้อยกว่า ค่าเฉลี่ย

```
Tenure -> median : 9.0 | mean : 10.189899366380917
WarehouseToHome -> median : 14.0 | mean : 15.639895891429633
HourSpendOnApp -> median : 3.0 | mean : 2.9315348837209303
OrderAmountHikeFromlastYear -> median : 15.0 | mean : 15.707921714818266
CouponUsed -> median : 1.0 | mean : 1.7510234462225531
OrderCount -> median : 2.0 | mean : 3.0080044676098288
DaySinceLastOrder -> median : 3.0 | mean : 4.543490512868683
```

- เติมข้อมูลส่วนที่หายไป

```
[33] for i in df.columns:
    if df_clean[i].isnull().sum() > 0:
        df_clean[i].fillna(df[i].median(), inplace=True)
```

จากขั้นตอนที่ผ่านมาได้มีการเลือกค่ามัธยฐานในการเติมในส่วนที่ข้อมูลขาดหายไป

Exploratory Data Analysis (EDA) and visualization of data

ในส่วนแรกจะเริ่มด้วยการปรับชนิดของข้อมูลแต่ละคอลัมน์ให้ถูกต้อง โดยมีการนำคอลัมน์ CustomerID ออก และแก้ไขคอลัมน์ Churn CityTier และ Complain ให้เป็น object

```
[252] df_EDA.drop(columns="CustomerID", inplace=True)

[254] df_EDA['Churn'] = df_EDA['Churn'].astype('object')
      df_EDA['CityTier'] = df_EDA['CityTier'].astype('object')
      df_EDA['Complain'] = df_EDA['Complain'].astype('object')
```

จากนั้นจะทำการแบ่งคอลัมน์เป็นสองประเภทคือ Categorical และ Numerical โดยแยกจากชนิดข้อมูลที่เป็น object หรือไม่

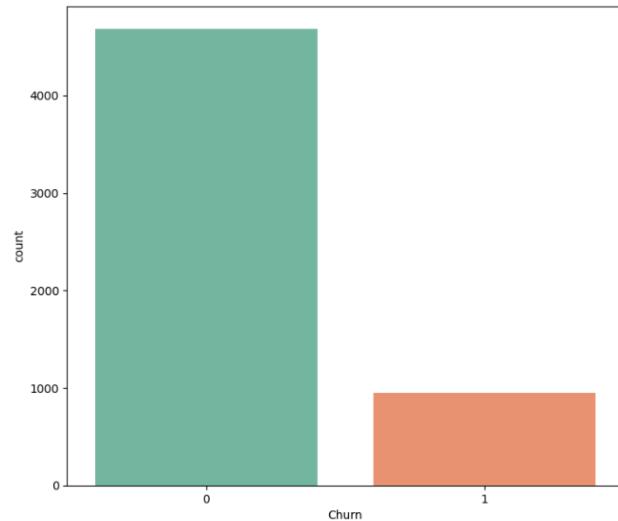
```
cat_col = []
num_col = []
for column in df_EDA.columns:
    if df_EDA[column].dtype=='object':
        cat_col.append(column)
    else:
        num_col.append(column)
```

สำหรับการ visualization แบ่งเป็น 2 แบบดังนี้

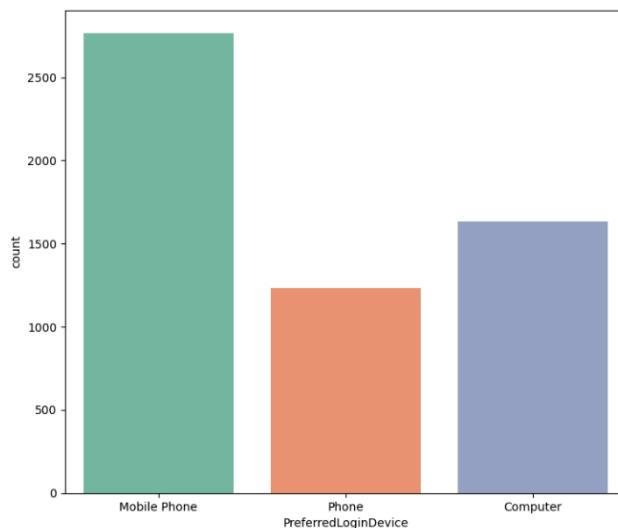
1. Univariate Analysis

1.1. Categorical Data

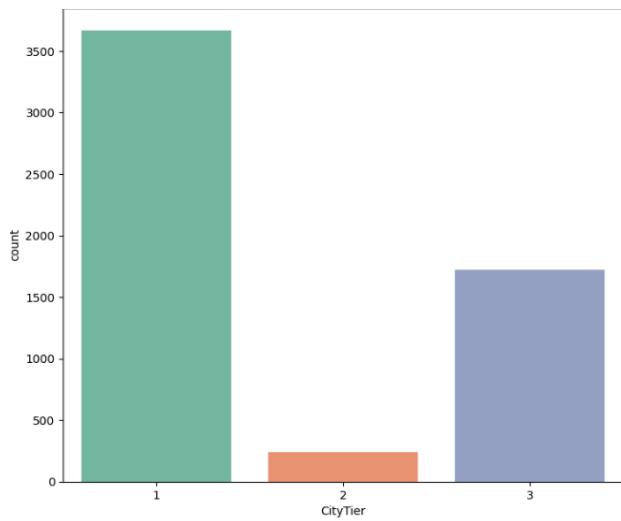
1.1.1. Churn : กราฟแสดงสัดส่วนของสถานะการ Churn ของลูกค้า



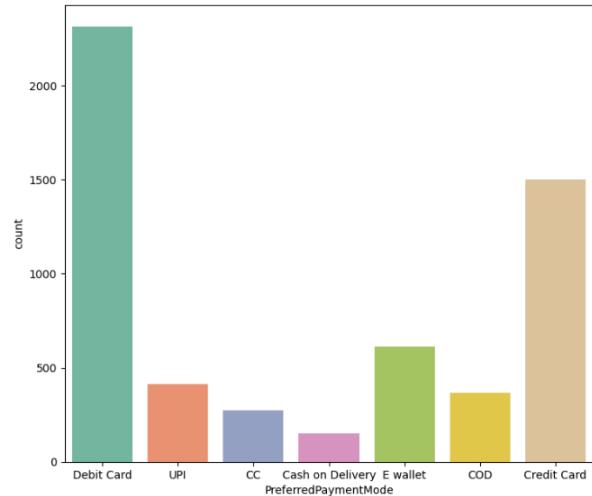
1.1.2. Preferred Login Device : กราฟแสดงสัดส่วนของเครื่องมือต่างๆ ที่ลูกค้าใช้ในการ login



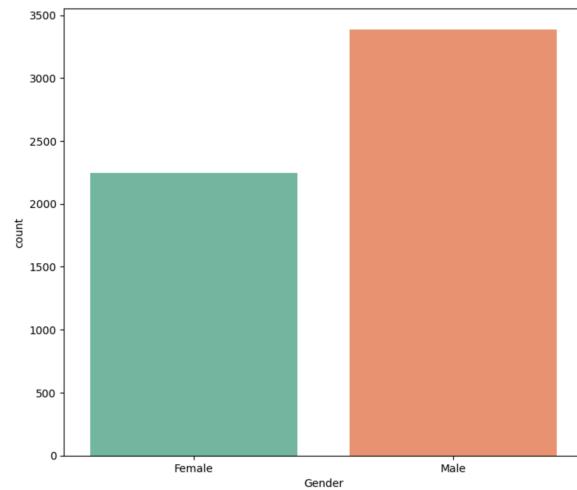
1.1.3. City Tier : กราฟแสดงระดับของเมืองที่ลูกค้าอยู่



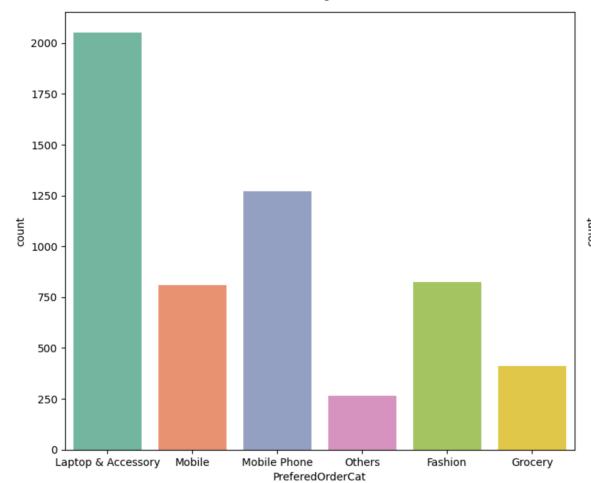
1.1.4. Preferred Payment Mode : กราฟแสดงวิธีที่ลูกค้าเลือกใช้ในการชำระเงิน



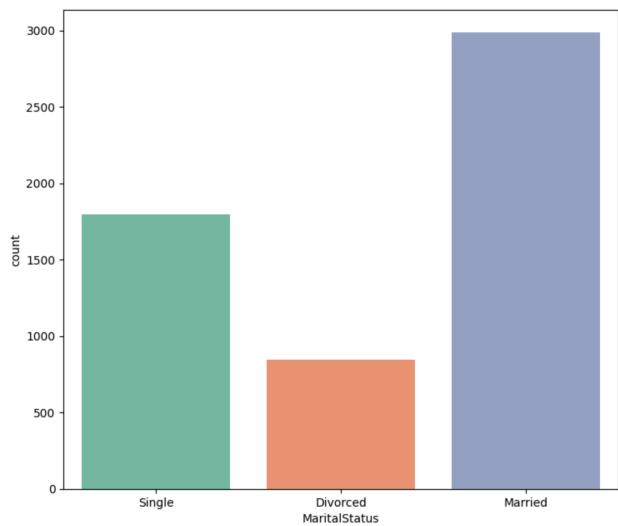
1.1.5. Gender : กราฟแสดงสัดส่วนเพศของลูกค้าที่ใช้บริการ



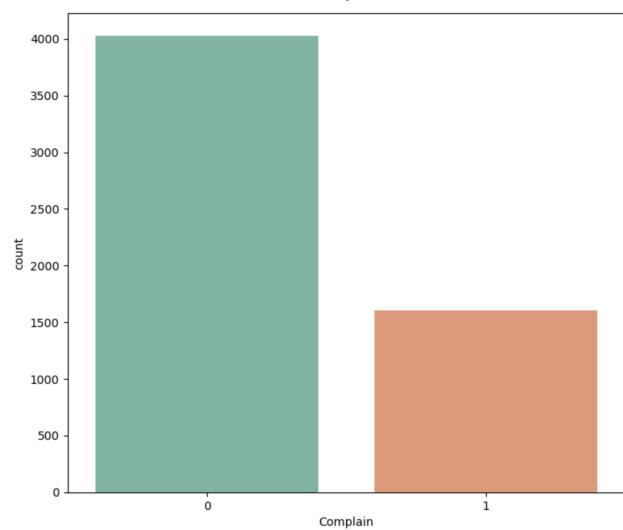
1.1.6. Preferred Order Cat : กราฟแสดงหมวดหมู่ของสินค้าที่ลูกค้าซื้อบ่อย



1.1.7. Marital Status : กราฟแสดงสัดส่วนสถานภาพสมรสของลูกค้า

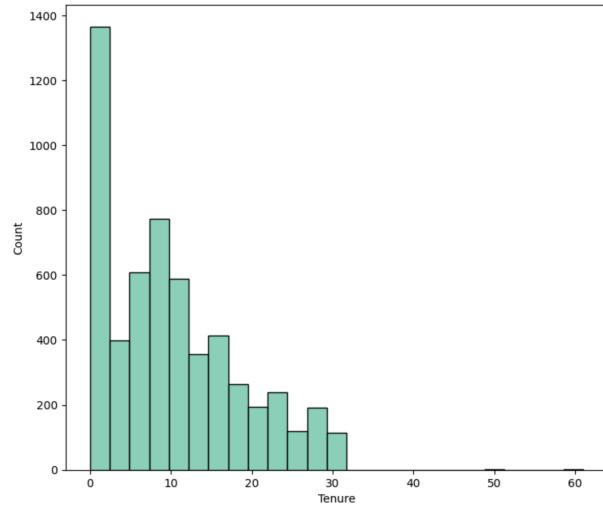


1.1.8. Complain : กราฟแสดงสัดส่วนของสถานะการ Complain ของลูกค้าในช่วง 1 เดือนก่อนหน้า

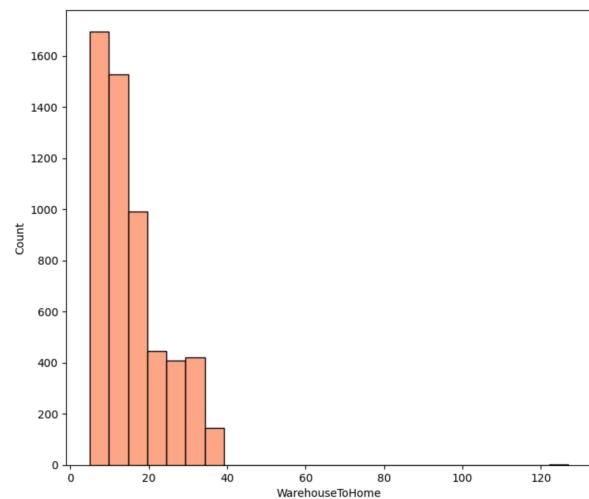


1.2. Numerical Data

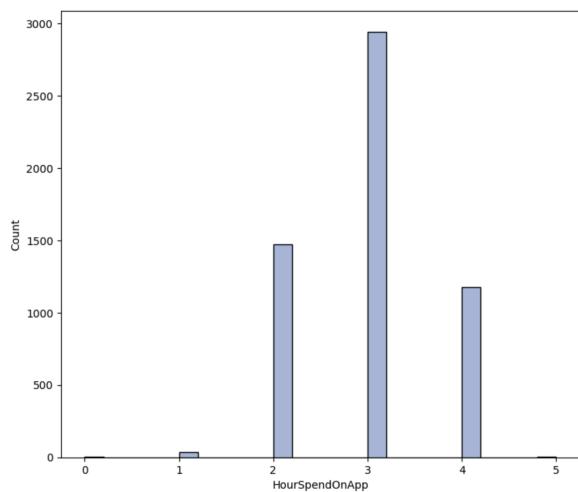
1.2.1. Tenure : กราฟแสดงสัดส่วนของระยะเวลาในการใช้บริการแพลตฟอร์ม



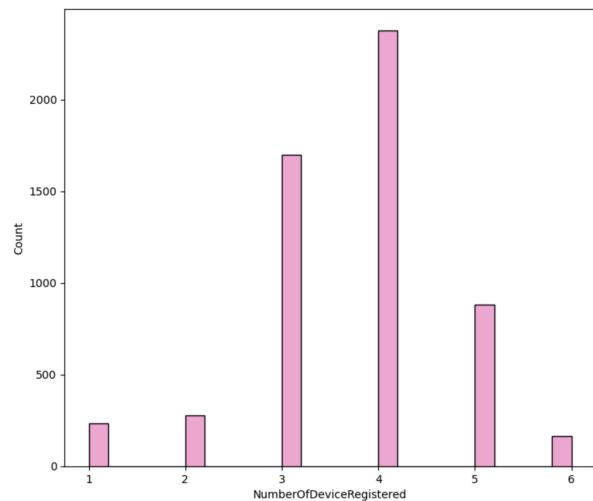
1.2.2. Warehouse To Home : กราฟแสดงข้อมูลระยะห่างระหว่างที่เก็บสินค้ากับบ้านของลูกค้า



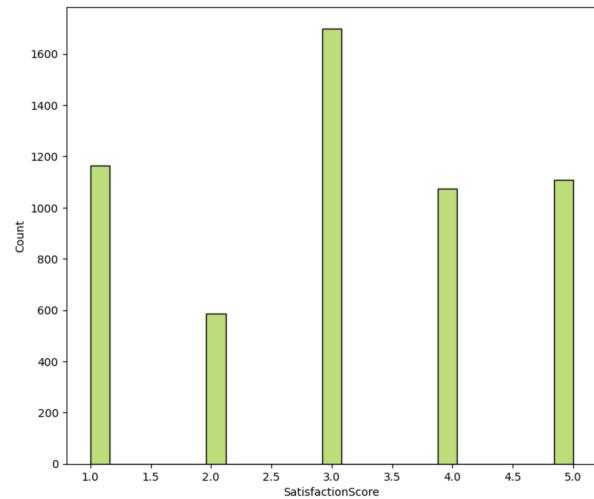
1.2.3. Hour Spend On App : กราฟแสดงสัดส่วนเวลาที่ลูกค้าใช้แอป



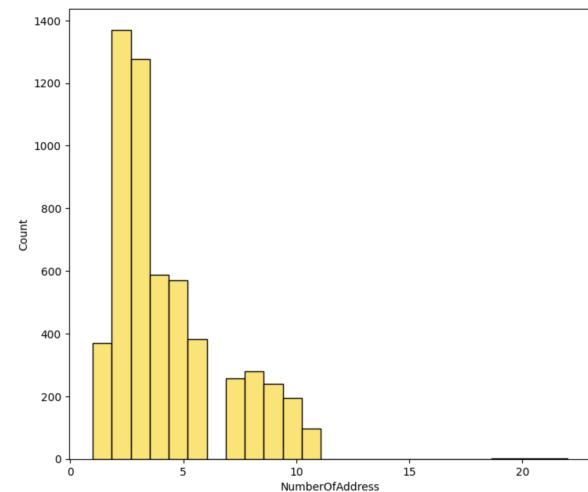
1.2.4. Number Of Device Registered : กราฟแสดงสัดส่วนจำนวนอุปกรณ์ที่ลูกค้าลงทะเบียนไว้



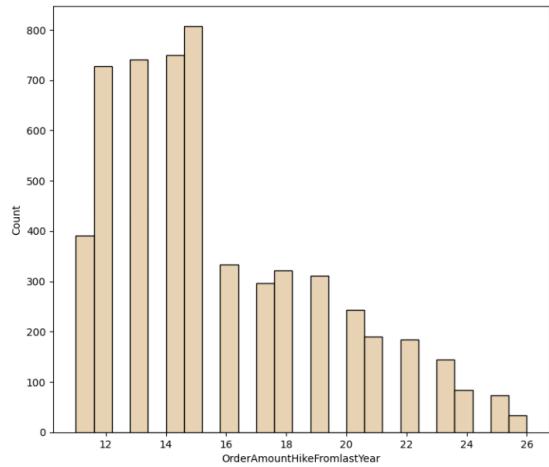
1.2.5. Satisfaction Score : กราฟแสดงคะแนนความพึงพอใจของลูกค้า



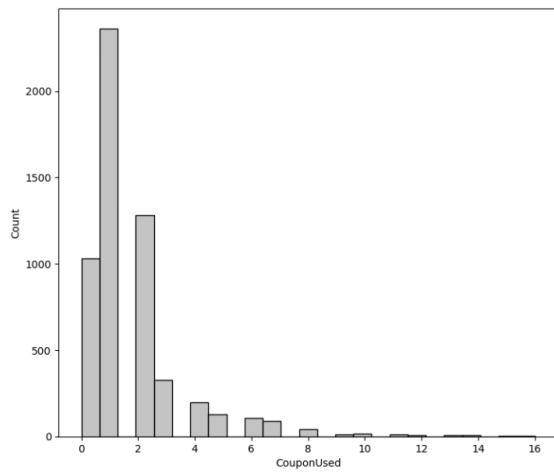
1.2.6. Number Of Address : กราฟแสดงข้อมูลจำนวนที่อยู่ทั้งหมดของลูกค้า



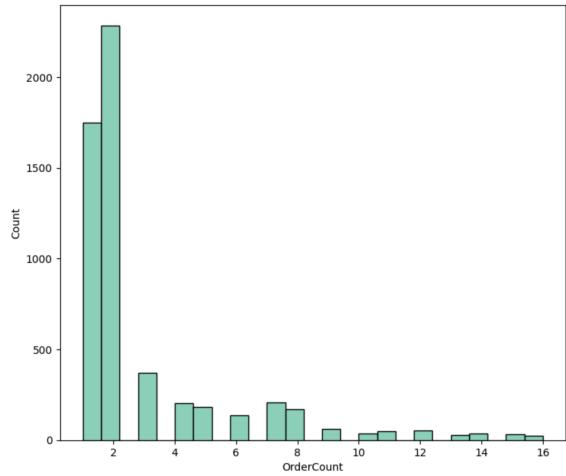
1.2.7. OrderAmountHike From Last year : กราฟแสดงสัดส่วนของจำนวนออร์เดอร์ที่เพิ่มขึ้นของลูกค้าแต่ละคนเมื่อเทียบกับปีที่แล้ว



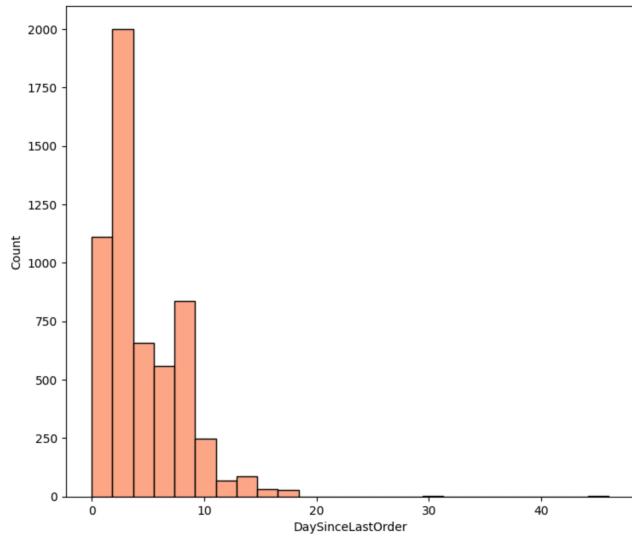
1.2.8. CouponUsed : กราฟแสดงสัดส่วนของจำนวนการใช้คูปองของลูกค้าแต่ละคนในช่วง 1 เดือนก่อนหน้านี้



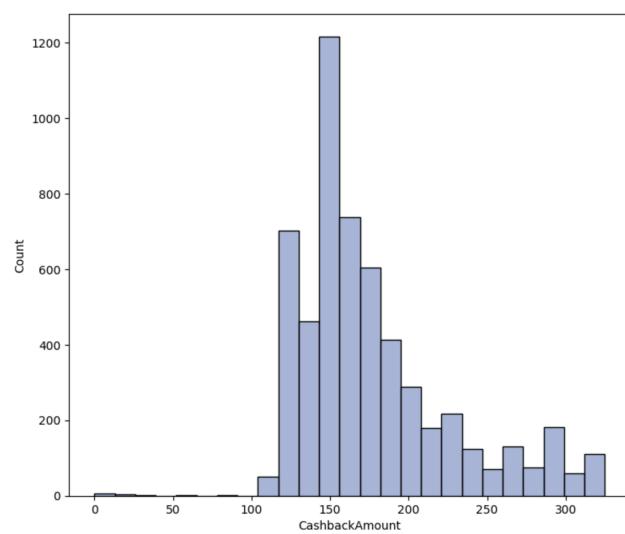
1.2.9. OrderCount : กราฟแสดงสัดส่วนของจำนวนออร์เดอร์รวมทั้งหมดในช่วง 1 เดือน ที่ผ่านมา



1.2.10. DaySinceLastOrder : กราฟแสดงสัดส่วนระยะเวลาตั้งแต่การซื้อครั้งล่าสุด



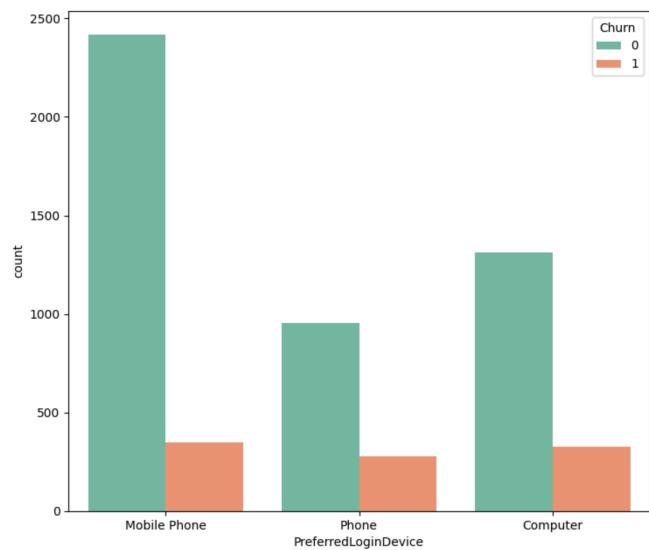
1.2.11. CashBackAmount : กราฟแสดงสัดส่วนของจำนวนเงินที่ลูกค้าได้รับคืน



2. Churn Analysis

2.1. Categorical Data

2.1.1. Preferred Login Device : ในกลุ่มลูกค้าที่ใช้ Mobile phone มีจำนวนคนที่ Churn ทั้งหมด 348 คน และคนที่ยังไม่ Churn ทั้งหมด 2417 คน
กลุ่มลูกค้าที่ใช้ Computer มีจำนวนคนที่ Churn ทั้งหมด 324 คน และคนที่ยังไม่ Churn ทั้งหมด 1310 คน
กลุ่มลูกค้าที่ใช้ Phone มีจำนวนคนที่ Churn ทั้งหมด 276 คน และคนที่ยังไม่ Churn ทั้งหมด 955 คน

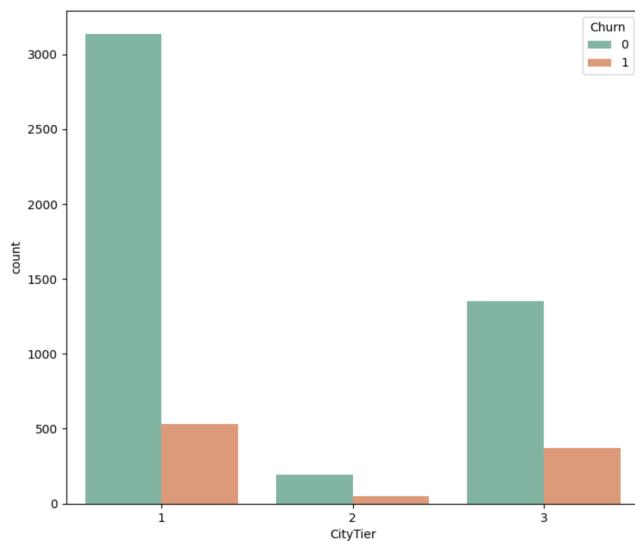


2.1.2. City Tier :

ในกลุ่มลูกค้าที่อาศัยอยู่ในเมืองระดับ 1 มีคนที่ Churn ทั้งหมด 532 คน และมีคนที่ยังไม่ Churn ทั้งหมด 3134 คน

ในกลุ่มลูกค้าที่อาศัยอยู่ในเมืองระดับ 2 มีคนที่ Churn ทั้งหมด 48 คน และมีคนที่ยังไม่ Churn ทั้งหมด 194 คน

ในกลุ่มลูกค้าที่อาศัยอยู่ในเมืองระดับ 3 มีคนที่ Churn ทั้งหมด 368 คน และมีคนที่ยังไม่ Churn ทั้งหมด 1354 คน



2.1.3. PreferredPaymentMode :

ในกลุ่มลูกค้าที่ชำระค่าสินค้าด้วยวิธี CC มีคน Churn ทั้งหมด 59 คน และมีคนที่ยังไม่ Churn 214 คน

ในกลุ่มลูกค้าที่ชำระค่าสินค้าด้วยวิธี COD มีคน Churn ทั้งหมด 105 คน และมีคนที่ยังไม่ Churn 260 คน

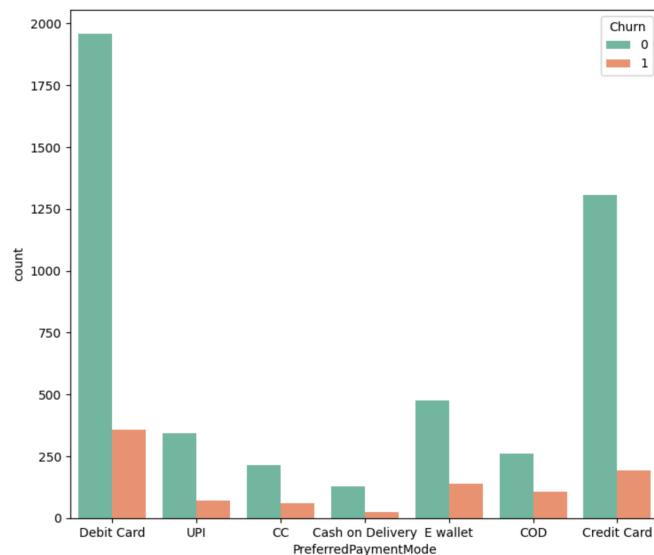
ในกลุ่มลูกค้าที่ชำระค่าสินค้าด้วยวิธี Cash on Delivery มีคน Churn ทั้งหมด 23 คน และมีคนที่ยังไม่ Churn 126 คน

ในกลุ่มลูกค้าที่ชำระค่าสินค้าด้วยวิธี Credit Card มีคน Churn ทั้งหมด 193 คน และมีคนที่ยังไม่ Churn 1308 คน

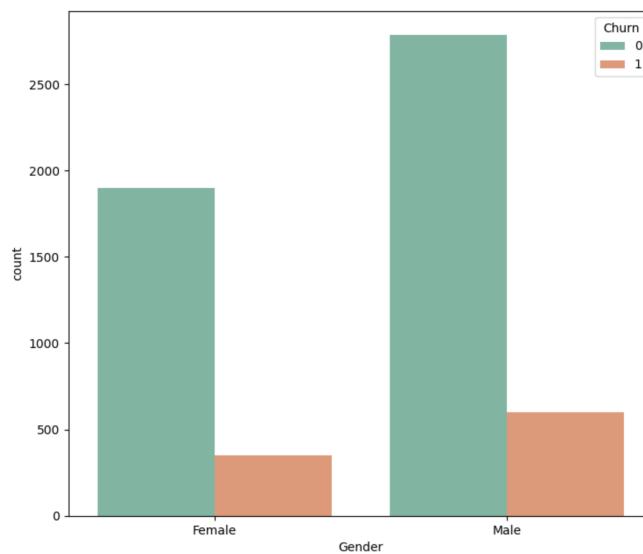
ในกลุ่มลูกค้าที่ชำระค่าสินค้าด้วยวิธี Debit Card มีคน Churn ทั้งหมด 356 คน และมีคนที่ยังไม่ Churn 1958 คน

ในกลุ่มลูกค้าที่ชำระค่าสินค้าด้วยวิธี E-wallet มีคน Churn ทั้งหมด 140 คน และมีคนที่ยังไม่ Churn 474 คน

ในกลุ่มลูกค้าที่ชำระค่าสินค้าด้วยวิธี UPI มีคน Churn ทั้งหมด 72 คน และมีคนที่ยังไม่ Churn 342 คน



2.1.4. Gender : ในกลุ่มผู้ใช้งานที่เป็นผู้หญิงมีคนที่ Churn ทั้งหมด 348 คนและไม่ Churn 1898 คน และในกลุ่มลูกค้าที่เป็นผู้ชายมีคนที่ Churn ทั้งหมด 600 คนและไม่ Churn 2784 คน



2.1.5. Preferred Order Cat :

ในกลุ่มลูกค้าที่ต้องการซื้อสินค้าประเภท Fashion มีคน Churn ทั้งหมด

128 คน และมีคนยังไม่ 698 คน

ในกลุ่มลูกค้าที่ต้องการซื้อสินค้าประเภท Grocery มีคน Churn ทั้งหมด

20 คน และมีคนยังไม่ 390 คน

ในกลุ่มลูกค้าที่ต้องการซื้อสินค้าประเภท Laptop & Accessory มีคน

Churn ทั้งหมด 210 คน และมีคนยังไม่ 1840 คน

ในกลุ่มลูกค้าที่ต้องการซื้อสินค้าประเภท Mobile มีคน Churn ทั้งหมด

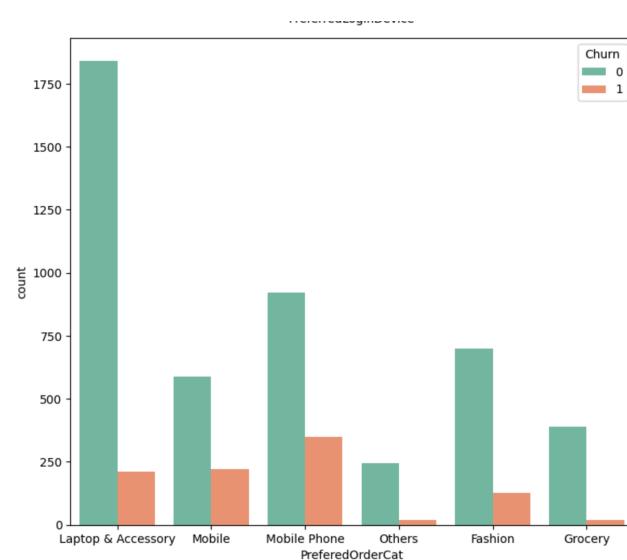
220 คน และมีคนยังไม่ 589 คน

ในกลุ่มลูกค้าที่ต้องการซื้อสินค้าประเภท Mobile Phone มีคน Churn

ทั้งหมด 350 คน และมีคนยังไม่ 921 คน

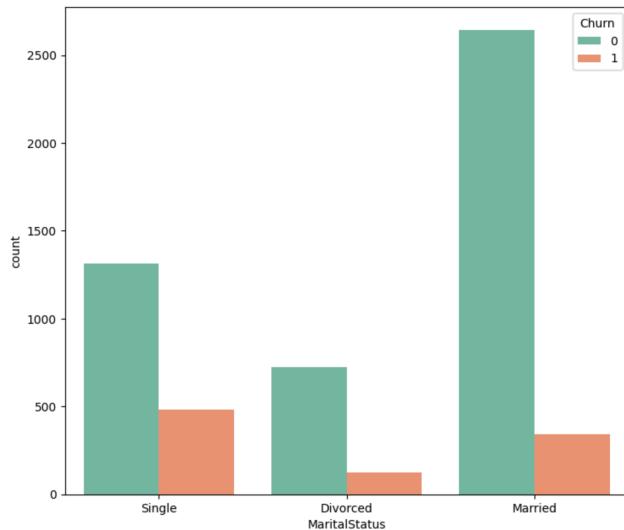
ในกลุ่มลูกค้าที่ต้องการซื้อสินค้าประเภท Others มีคน Churn ทั้งหมด 20

คน และมีคนยังไม่ 244 คน

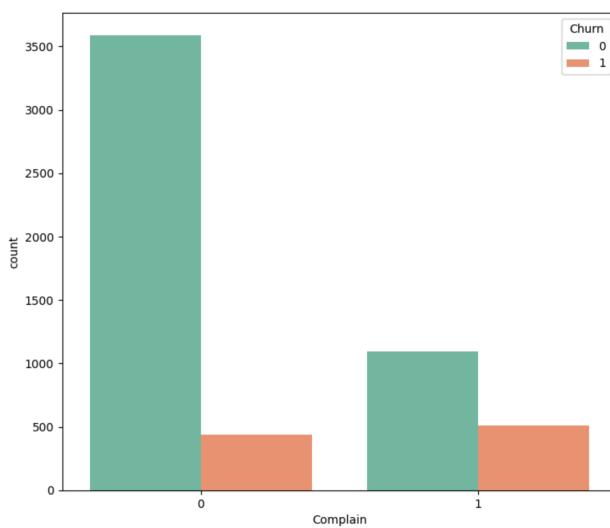


2.1.6. Martial Status :

ในกลุ่มลูกค้าที่สถานะสมรส มีคนที่ Churn ทั้งหมด 480 คนและไม่ Churn 1316 คน ในกลุ่มลูกค้าที่หย่ากันแล้ว มีคนที่ Churn 124 คนและไม่ Churn 724 และในกลุ่มลูกค้าที่แต่งงานแล้ว มีคนที่ Churn 344 คนและไม่ Churn 2642 คน

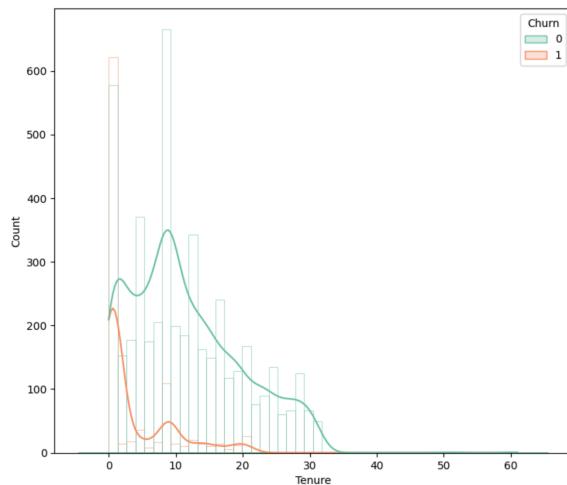


2.1.7. Complain : ในกลุ่มลูกค้าที่ไม่ได้ Complain มีคนที่ Churn ทั้งหมด 440 คน และไม่ Churn 3586 คน และ ในกลุ่มลูกค้าที่ Complain มีคนที่ Churn ทั้งหมด 508 คน และไม่ Churn 1096 คน

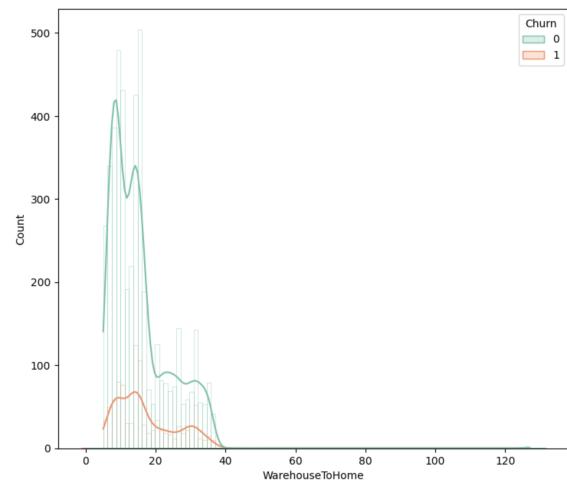


2.2. Numerical Data

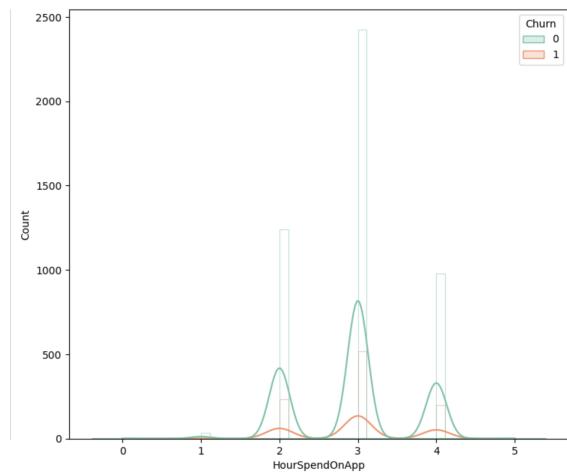
2.2.1. Tenure : ยิ่งระยะเวลาการในการใช้บริการนานขึ้น โอกาสการ Churn จะมีแนวโน้มที่จะลดลงเรื่อยๆ



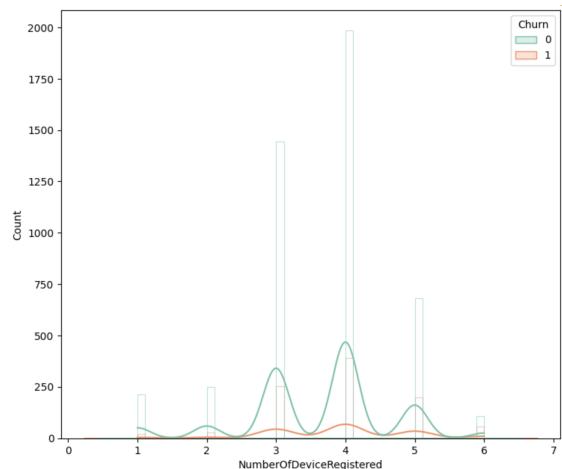
2.2.2. WarehouseToHome : โอกาสการ Churn จะมีแนวโน้มลดลงเมื่อระยะห่างมากขึ้น



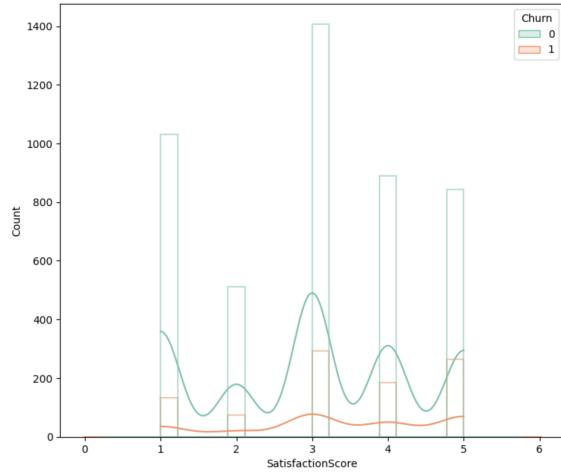
2.2.3. HourSpendOnApp : โอกาสการ Churn จะสูงที่สุด เมื่อชั่วโมงในการใช้แอปเท่ากับ 3 ชั่วโมง



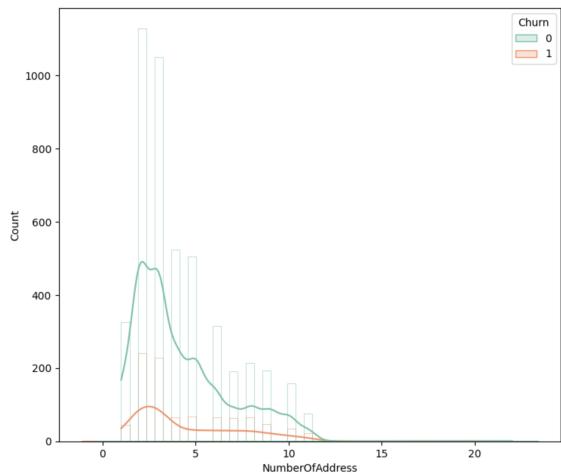
2.2.4. NumberOfDeviceRegistered : จำนวนอุปกรณ์ ในช่วง 3 - 5 อุปกรณ์ มีโอกาสการ Churn ใกล้เคียงกัน



2.2.5. SatisfactionScore : จากกราฟจะเห็นได้ว่าโอกาสการ Churn จะมีมากที่สุดเมื่อคะแนนมีค่าเท่ากับ 3

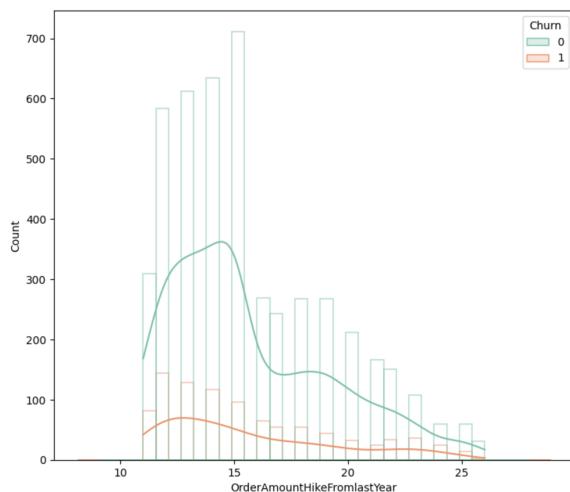


2.2.6. NumberOfAddress : โอกาสการ Churn มีมากเมื่อจำนวนที่อยู่น้อย และเมื่อจำนวนที่อยู่มากขึ้น แนวโน้มของการ Churn ลดลง

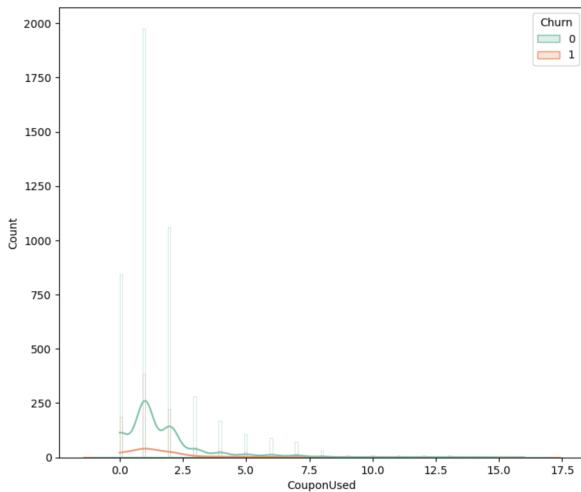


2.2.7. OrderAmountHikeFromLastYear :

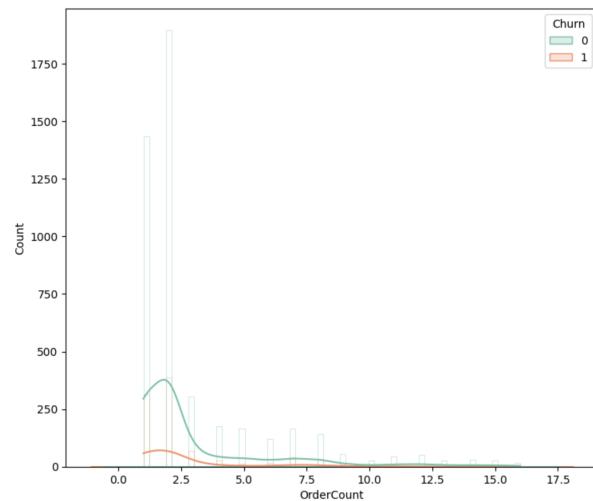
จะเห็นได้ว่าเมื่อเปอร์เซนต์จำนวนอโเดอร์ที่เพิ่มขึ้น อาจจะส่งผลให้โอกาสการ Churn ลดลง



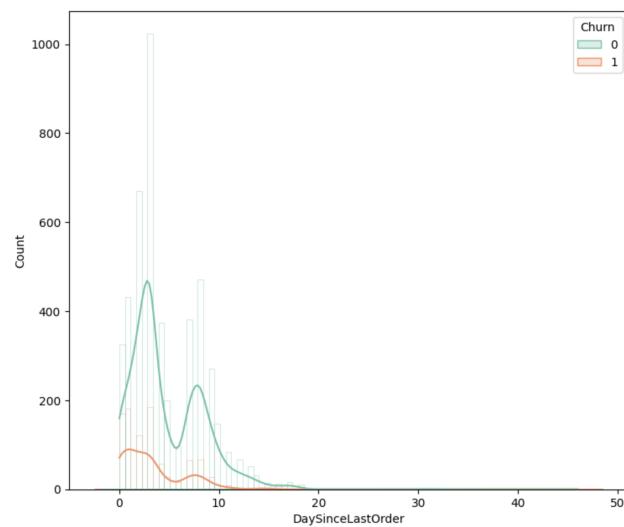
2.2.8. CouponUsed : เมื่อจำนวนคูปองที่ใช้มากขึ้น โอกาสการ Churn มีแนวโน้มที่ลดลง แต่เป็นแนวโน้มที่ไม่ชัดเจนมากนัก กล่าวคือจำนวนคูปองอาจไม่ได้ส่งผลต่อการ Churn มาก



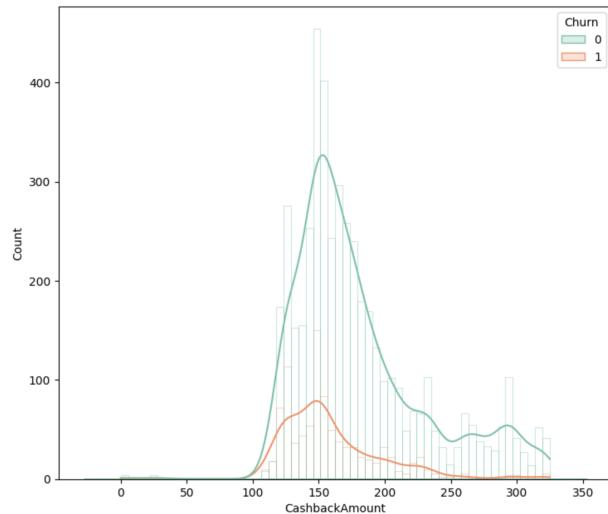
2.2.9. OrderCount : ในช่วงที่มีจำนวน order น้อย โอกาสการ Churn จะมากกว่า ช่วงที่ order เยอะ



2.2.10. DaySinceLastOrder : เมื่อจำนวนวันมากขึ้น แนวโน้มการ Churn จะลดลง



2.2.11. CashbackAmount : เมื่อจำนวนเงินที่ลูกค้าได้รับ การ Churn
มีแนวโน้มที่จะลดลงตามลำดับ



Modeling method

- ทำการเตรียมข้อมูลโดยการแยก ข้อมูลออกเป็นตารางที่มี data type เป็น num (int, float) และที่มี data type เป็น object

```
[58] ecom = df_clean.copy()

[59] ecom_num = ecom.select_dtypes('number')
ecom_num.drop(columns='CustomerID', inplace=True)
ecom_num

[60] Churn Tenure CityTier WarehouseToHome HourSpendOnApp NumberOfDeviceRegistered SatisfactionScore NumberOfAddress Complain OrderAmountHikeFromlast
[61] 0 1 4.0 3 6.0 3.0 3 2 9 1
[62] 1 1 9.0 1 8.0 3.0 4 3 7 1
[63] 2 1 9.0 1 30.0 2.0 4 3 6 1
[64] 3 1 0.0 3 15.0 2.0 4 5 8 0
[65] 4 1 0.0 1 12.0 3.0 3 5 3 0
[66] ...
[67] 5625 0 10.0 1 30.0 3.0 2 1 6 0
[68] 5626 0 13.0 1 13.0 3.0 5 5 6 0
[69] 5627 0 1.0 1 11.0 3.0 2 4 3 1
[70] 5628 0 23.0 3 9.0 4.0 5 4 4 0
[71] 5629 0 8.0 1 15.0 3.0 2 3 4 0
5630 rows x 14 columns
```

- ข้อมูลที่มี data type เป็น num จะทำการตัด customer ID ออกไป
- ข้อมูลที่มี data type เป็น object จะใช้ฟังก์ชัน get_dummies เพื่อทำการแยกเป็น column ที่มีแค่เลข 0 กับ 1 จากนั้น drop column ที่ไม่ได้ใช้ออก แล้วรวมเข้าด้วยกัน

```
[59] ecom_obj = ecom.select_dtypes('object')
ecom_cat = pd.get_dummies(ecom_obj).astype(int)
ecom_cat.drop(columns='Gender_Male', inplace=True)
ecom_cat

[60] PreferredLoginDevice_Computer PreferredLoginDevice_Mobile Phone PreferredLoginDevice_Phone PreferredPaymentMode_CC PreferredPaymentMode_COD PreferredPay
[61] 0 0 1 0 0 0
[62] 1 0 0 1 0 0 0
[63] 2 0 0 1 0 0 0
[64] 3 0 0 1 0 0 0
[65] 4 0 0 1 1 1 0
[66] ...
[67] 5625 1 0 0 0 0 0
[68] 5626 0 1 0 0 0 0
[69] 5627 0 1 0 0 0 0
[70] 5628 1 0 0 0 0 0
[71] 5629 0 1 0 0 0 0
5630 rows x 20 columns

[60] ecom_prep = pd.concat([ecom_num, ecom_cat], axis=1)
ecom_prep

[61] Churn Tenure CityTier WarehouseToHome HourSpendOnApp NumberOfDeviceRegistered SatisfactionScore NumberOfAddress Complain OrderAmountHikeFromlast
[62] 0 1 4.0 3 6.0 3.0 3 2 9 1
[63] 1 1 9.0 1 8.0 3.0 4 3 7 1
[64] 2 1 9.0 1 30.0 2.0 4 3 6 1
[65] 3 1 0.0 3 15.0 2.0 4 5 8 0
[66] 4 1 0.0 1 12.0 3.0 3 5 3 0
[67] ...
[68] 5625 0 10.0 1 30.0 3.0 2 1 6 0
[69] 5626 0 13.0 1 13.0 3.0 5 5 6 0
[70] 5627 0 1.0 1 11.0 3.0 2 4 3 1
[71] 5628 0 23.0 3 9.0 4.0 5 4 4 0
[72] 5629 0 8.0 1 15.0 3.0 2 3 4 0
5630 rows x 34 columns
```

Decision tree

- ใช้ library sklearn เพื่อแบ่งข้อมูลสำหรับการ train และ test โดยกำหนดให้มีอัตราส่วนเป็น 70:30
- จำนวนทำการ plot decision tree

```
[65] from sklearn.model_selection import train_test_split
ecom_train, ecom_test = train_test_split(ecom_prep, test_size=0.3)

[66] from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(min_samples_leaf=10, max_depth=4)
tree.fit(ecom_train.drop(columns='Churn'),
         ecom_train['Churn'])

DecisionTreeClassifier(max_depth=4, min_samples_leaf=10)
```

- ทำการหาค่า feature important พบว่าข้อมูลจาก column “Tenure” มีอิทธิพลต่อการตัดสินใจมากที่สุด

```
[67] pd.DataFrame(dict(Feature=ecom_train.columns[1:],
                      Value=tree.feature_importances_))\
    .sort_values(by='Value', ascending=False)
```

Feature	Value
Tenure	0.578229
Complain	0.145070
NumberOfAddress	0.092046
CashbackAmount	0.063273
DaySinceLastOrder	0.045951
NumberOfDeviceRegistered	0.044129
SatisfactionScore	0.021221
CouponUsed	0.005525
PreferredPaymentMode_UPi	0.004556

- Classification report ให้ค่า accuracy อยู่ที่ 0.89

```
[71] from sklearn.metrics import classification_report

[72] res = tree.predict(ecom_test.drop(columns='Churn'))
print(classification_report(y_true=ecom_test['Churn'].values, y_pred=res))

precision    recall   f1-score   support
          0       0.93      0.95      0.94     1391
          1       0.72      0.65      0.69      298

accuracy                           0.89     1689
macro avg       0.82      0.80      0.81     1689
weighted avg    0.89      0.89      0.89     1689
```

Logistic Regression

- ทำการ import Logistic Regression model กำหนดค่า max_iter ไว้ที่ 10,000 เพื่อให้ algorithm ของ model มีเวลาคำนวณมากขึ้น

```
[75] from sklearn.linear_model import LogisticRegression  
  
[76] lrg = LogisticRegression(solver='lbfgs', max_iter=10000)  
      lrg.fit(ecom_train.drop(columns='Churn'),  
              ecom_train['Churn'])
```

↳ ▾ LogisticRegression
LogisticRegression(max_iter=10000)

- Classification report ให้ค่า accuracy อยู่ที่ 0.90

```
[77] res = lrg.predict(ecom_test.drop(columns='Churn'))  
      print(classification_report(y_true=ecom_test['Churn'].values, y_pred=res))
```

	precision	recall	f1-score	support
0	0.91	0.98	0.94	1391
1	0.84	0.54	0.66	298
accuracy			0.90	1689
macro avg	0.87	0.76	0.80	1689
weighted avg	0.90	0.90	0.89	1689

K-neighbors Classifier

- Classification report ให้ค่า accuracy อยู่ที่ 0.87

```
[78] from sklearn.neighbors import KNeighborsClassifier  
  
[79] knn = KNeighborsClassifier()  
      knn.fit(ecom_train.drop(columns='Churn'),  
              ecom_train['Churn'])
```

↳ ▾ KNeighborsClassifier
KNeighborsClassifier()

```
[80] res = knn.predict(ecom_test.drop(columns='Churn'))  
      print(classification_report(y_true=ecom_test['Churn'].values, y_pred=res))
```

	precision	recall	f1-score	support
0	0.90	0.95	0.92	1391
1	0.68	0.48	0.56	298
accuracy			0.87	1689
macro avg	0.79	0.72	0.74	1689
weighted avg	0.86	0.87	0.86	1689

Random Forest Classifier

- ทำการ import RandomForestClassifier เพื่อเรียกใช้ model จากนั้นเลือกใช้เกณฑ์ gini พร้อมกับกำหนดให้ค่า n_estimators = 100, random_state = 1 และ bootstrap = True

```
[81] from sklearn.ensemble import RandomForestClassifier  
  
[82] rf = RandomForestClassifier(  
    criterion='gini',  
    n_estimators=100,  
    random_state=1,  
    bootstrap=True  
)  
rf.fit(ecom_train.drop(columns='Churn'),  
      ecom_train['Churn'])
```

RandomForestClassifier
RandomForestClassifier(random_state=1)

- Classification report ให้ค่า accuracy อยู่ที่ 0.96

```
[83] res = rf.predict(ecom_test.drop(columns='Churn'))  
print(classification_report(y_true=ecom_test['Churn'].values, y_pred=res))
```

	precision	recall	f1-score	support
0	0.96	0.99	0.98	1391
1	0.95	0.82	0.88	298
accuracy			0.96	1689
macro avg	0.96	0.91	0.93	1689
weighted avg	0.96	0.96	0.96	1689

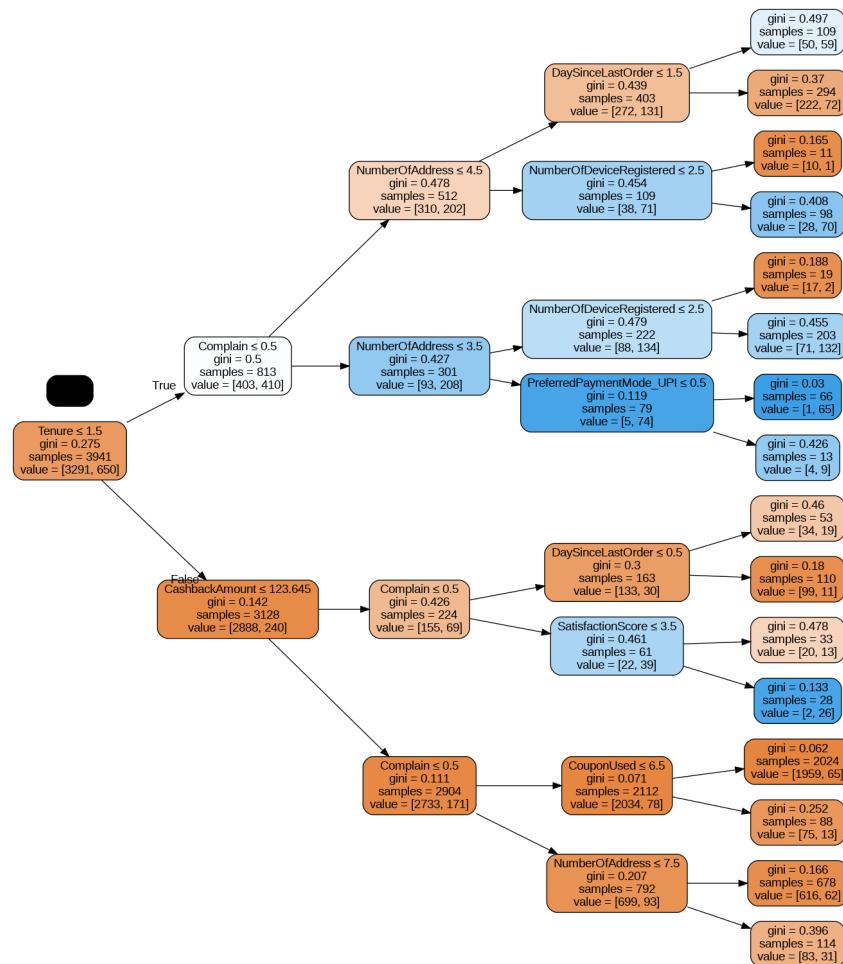
- ทำการสร้าง confusion matrix เพื่อดูผลการ predict

```
[84] from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay  
import matplotlib.pyplot as plt  
  
[85] cfm = confusion_matrix(ecom_test['Churn'], res)  
print(cfm)  
[[1378 13]  
 [ 53 245]]
```

```
[86] plt.figure(figsize=(20,15))  
cm_display = ConfusionMatrixDisplay(confusion_matrix = cfm, display_labels = [0, 1])  
cm_display.plot()  
plt.show()
```

Modeling results and discussion

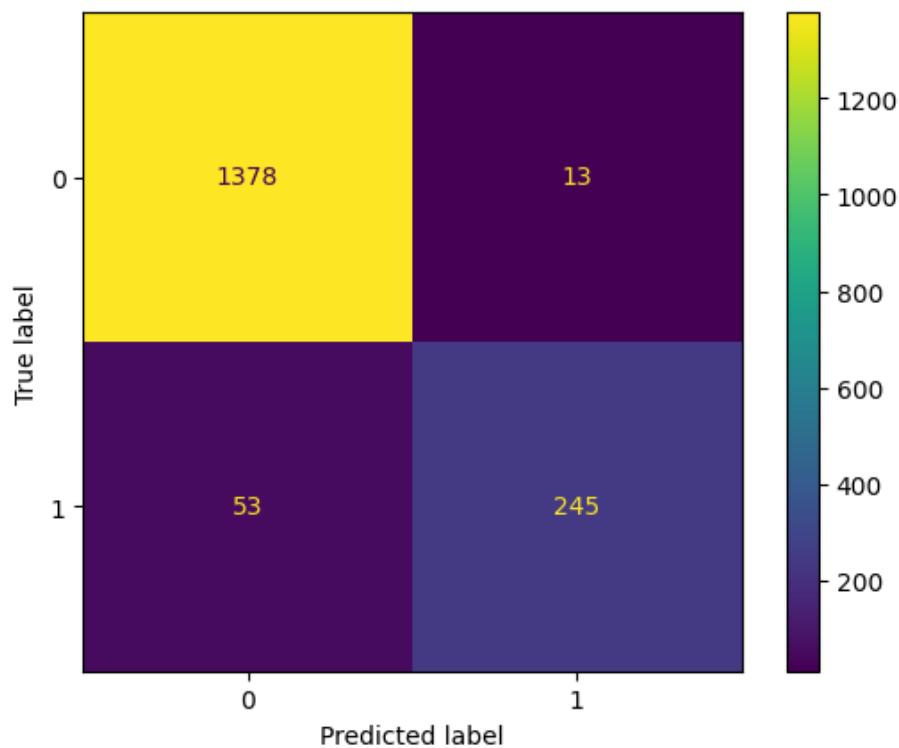
จากการทำ decision tree ทำให้เห็นว่า Tenure เป็นตัวแปรที่มีค่า feature importance สูงที่สุด กล่าวคือมีอิทธิพลต่อการ churn มากที่สุด รองลงมาเป็น Complain, NumberOfAddress, CashbackAmount, DaySinceLastOrder, NumberOfDeviceRegistered, SatisfactionScore, CouponUsed และ PreferredPaymentMode_UPi ส่วน column อื่น ๆ จะไม่มีอิทธิพลต่อการ churn และจากการทำ visualization ทำให้ทราบว่า ลูกค้าที่ใช้บริการมาในนานมากจะมีแนวโน้มจะเลิกใช้บริการ



ผลลัพธ์ที่ได้จากการทำ model พบว่าค่า accuracy ที่ได้จากการทำ decision tree มีค่าน้อยกว่า logistic regression เป็นเพราะความสัมพันธ์ของ column churn กับ column อื่น ๆ เป็นไปในทางเชิงเส้นมากกว่า ทำให้การทำ model แบบ linear สามารถ predict ได้แม่นยำมากกว่าการทำ decision tree ที่จะสนับสนุนความสัมพันธ์แบบ non-linear มากกว่า

ส่วน model แบบ K-Nearest Neighbor จะให้ค่า accuracy ที่ต่ำที่สุด แม้ในกรณีที่มีการปรับค่า n_neighbors หรือการใช้วิธีแบบ euclidean

ในขณะที่ model แบบ Random Forest จะให้ค่า accuracy ที่สูงที่สุด สามารถประเมินผลได้จาก การดู confusion matrix ซึ่งจะเห็นว่าถ้าค่าจริงเป็น 1 จะสามารถ predict ได้ว่าเป็น 1 จริง ๆ อยู่ที่ 245 samples จากทั้งหมด 298 samples คิดเป็น 82.21% ในขณะที่ ถ้าค่าจริงเป็น 0 จะสามารถ predict ได้ว่าเป็น 0 จริง ๆ อยู่ที่ 1378 samples จากทั้งหมด 1391 samples คิดเป็น 99.06% อัตราส่วนนี้แสดงให้ถึงค่า recall หรือค่า true positive rate



Conclusion

ในการทำงานสามารถแบ่งขั้นตอนได้เป็นสองส่วนหลัก ๆ ส่วนแรกคือการเตรียมข้อมูลและวิเคราะห์ข้อมูล และส่วนที่สองคือการทำ model สำหรับทำนาย

ส่วนแรกเริ่มจากการเตรียมข้อมูล โดยได้พบรหัสข้อมูลที่หายไปซึ่งแก้ไขด้วยการใช้ค่ามัธยฐาน จากนั้นทำการแสดงข้อมูลโดยวิเคราะห์สัดส่วนข้อมูลแต่ละคอลัมน์ก่อนจากนั้นนำแต่ละคอลัมน์มาวิเคราะห์กับค่า churn

ส่วนที่สองเป็นการทำ modeling โดยมีการทำทั้งหมด 4 แบบ คือ Decision tree, Logistic Regression, K-neighbors Classifier และ Random Forest Classifier ซึ่งจากการทำ decision tree ทำให้เห็นว่า Tenure เป็นตัวแปรที่มีค่า feature importance สูงที่สุด กล่าวคือมีอิทธิพลต่อการ churn มากที่สุด และเมื่อเปรียบเทียบ model ทั้งหมด จะพบว่า model แบบ Random Forest จะให้ค่า accuracy ที่สูงที่สุด

สุดท้ายในการทำ model สามารถพัฒนาเพิ่มด้วยการใช้วิธีการทำนายอื่น ๆ เพื่อเปรียบเทียบ วิธีที่มีประสิทธิภาพที่สุด