단위 프로젝트 EDA 보고서

1. 데이터 설명

사용 데이터 출처

출처	Kaggle
Link	https://www.kaggle.com/datasets/ankitverma2010/ecommerce-customer-churn-analysis-and-prediction
Dataset Name	E-commerce Customer Churn Analysis and Prediction
Total row	5630
Total columns	20

변수 설명

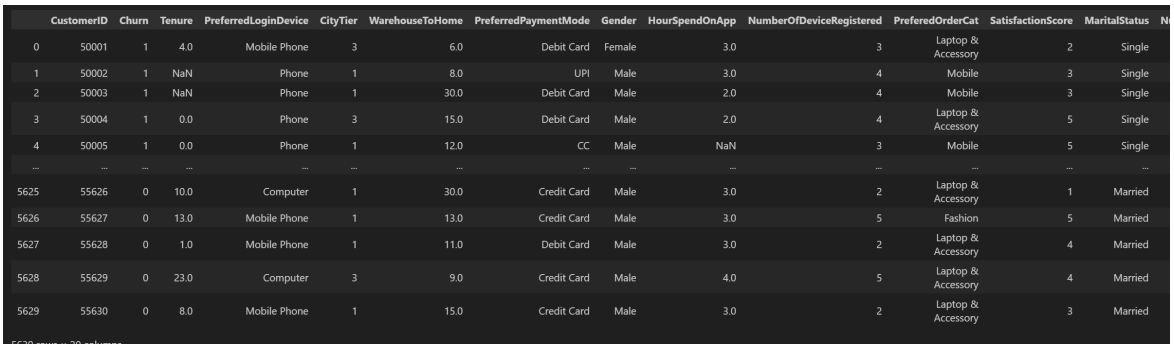
변수	type	NA	고유값	설명
CustomerID	int64	X	50001 ~ 55630	Unique customer ID
Churn	int64	X	1, 0	Churn Flag
Tenure	float64	O	4. nan 0. 13. 11. 9. 19. 20. 14. 8. 18. 5. 2. 30. 1. 23. 3. 29. 6. 26. 28. 7. 24. 25. 10. 15. 22. 27. 16. 12. 21. 17. 50. 60. 31. 51. 61.	Tenure of customer in organization
PreferredLoginDevice	object	X	'Mobile Phone' ,'Phone' ,'Computer'	Preferred login device of customer
CityTier	int64	X	3, 1, 2	City tier
WarehouseToHome	float64	O	6. 8. 30. 15. 12. 22. 11. 9. 31. 18. 13. 20. 29. 28. 26. 14. nan 10. 27. 17. 23. 33. 19. 35. 24. 16. 25. 32. 34. 5. 21. 126. 7. 36. 127.	Distance in between warehouse to home of customer

변수	type	NA	고유값	설명
PreferredPaymentMode	object	X	'Debit Card' 'UPI' 'CC' 'Cash on Delivery' 'E wallet' 'COD' 'Credit Card'	Preferred payment method of customer
Gender	object	X	'Female' 'Male'	Gender of customer
HourSpendOnApp	float64	O	3. 2. nan 1. 0. 4. 5.	Number of hours spend on mobile application or website
NumberOfDeviceRegistered	int64	X	3 4 5 2 1 6	Total number of deceives is registered on particular customer
PreferedOrderCat	object	X	'Laptop & Accessory' 'Mobile' 'Mobile Phone' 'Others' 'Fashion' 'Grocery'	Preferred order category of customer in last month
SatisfactionScore	int64	X	2 3 5 4 1	Satisfactory score of customer on service
MaritalStatus	object	X	Single' 'Divorced' 'Married'	Marital status of customer
NumberOfAddress	int64	X	9 7 6 8 3 2 4 10 1 5 19 21 11 20 22	Total number of added added on particular customer
Complain	int64	X	1 0	Any complaint has been raised in last month
OrderAmountHikeFromlastYear	float64	X	11. 15. 14. 23. 22. 16. 12. nan 13. 17. 18. 24. 19. 20. 21. 25. 26.	Percentage increases in order from last year

변수	type	NA	고유값	설명
CouponUsed	float64	O	1. 0. 4. 2. 9. 6. 11. nan 7. 12. 10. 5. 3. 13. 15. 8. 14. 16.	Total number of coupon has been used in last month
OrderCount	float64	O	1. 6. 2. 15. 4. 7. 3. 9. nan 11. 5. 12. 10. 8. 13. 14. 16.	Total number of orders has been places in last month
DaySinceLastOrder	float64	O	5. 0. 3. 7. 2. 1. 8. 6. 4. 15. 9. 11. 10. nan 13. 12. 17. 16. 14. 30. 46. 18. 31.	Day Since last order by customer
CashbackAmount	float64	X	159.93 ~ 173.78	Average cashback in last month

```
<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
```

데이터 확인



	CustomerID	Churn	Tenure	PreferredLoginDevice	CityTier	WarehouseToHome	PreferredPaymentMode	Gender	HourSpendOnApp	NumberOfDeviceRegistered	PreferedOrderCat	SatisfactionScore	MaritalStatus	N
0	50001	1	4.0	Mobile Phone	3	6.0	Debit Card	Female	3.0	3	Laptop & Accessory	2	Single	
1	50002	1	NaN	Phone	1	8.0	UPI	Male	3.0	4	Mobile	3	Single	
2	50003	1	NaN	Phone	1	30.0	Debit Card	Male	2.0	4	Mobile	3	Single	
3	50004	1	0.0	Phone	3	15.0	Debit Card	Male	2.0	4	Laptop & Accessory	5	Single	
4	50005	1	0.0	Phone	1	12.0	CC	Male	NaN	3	Mobile	5	Single	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
5625	55626	0	10.0	Computer	1	30.0	Credit Card	Male	3.0	2	Laptop & Accessory	1	Married	
5626	55627	0	13.0	Mobile Phone	1	13.0	Credit Card	Male	3.0	5	Fashion	5	Married	
5627	55628	0	1.0	Mobile Phone	1	11.0	Debit Card	Male	3.0	2	Laptop & Accessory	4	Married	
5628	55629	0	23.0	Computer	3	9.0	Credit Card	Male	4.0	5	Laptop & Accessory	4	Married	
5629	55630	0	8.0	Mobile Phone	1	15.0	Credit Card	Male	3.0	2	Laptop & Accessory	3	Married	

5630 rows × 20 columns

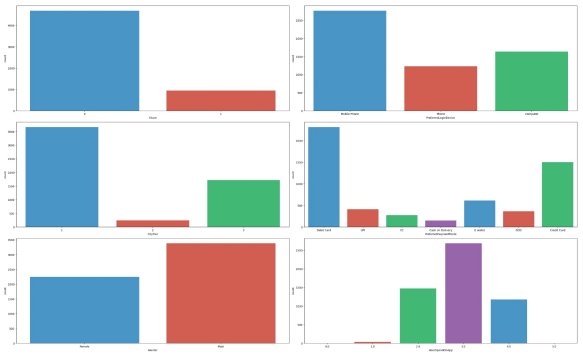
- 총 5630개의 행과 20개의 컬럼이 존재하는 것을 알 수 있다.
- 결측치가 존재하고, 문자형 데이터가 있는 것을 보아, 데이터가 전처리가 필요하다고 판단하였다

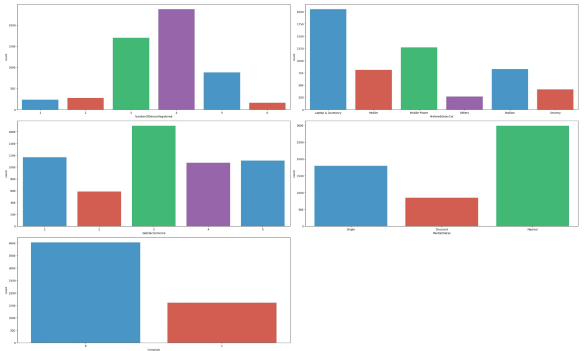
2. 데이터 샘플링

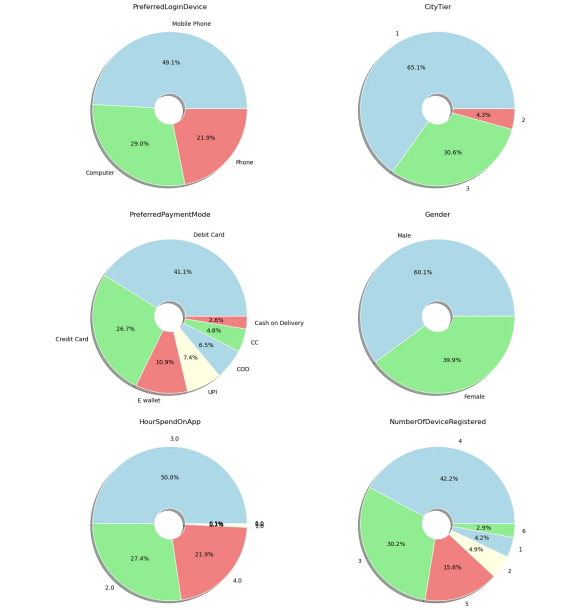
데이터 특징

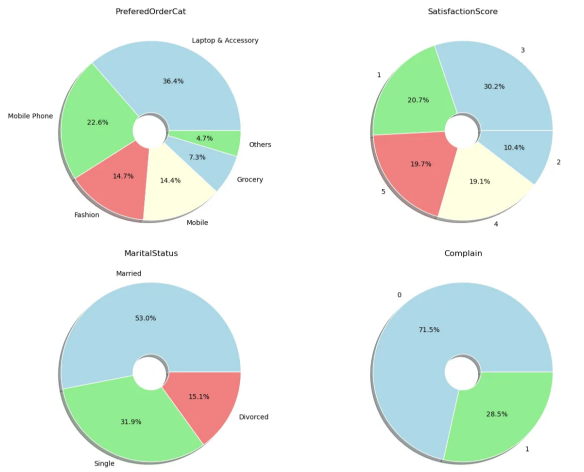
- **barchart & piechart**

데이터의 특성 이해	잠재적 이상값 (outliers) 감지	모델링 전 변수 중요성 평가	데이터 불균형 확인
각 변수의 고유값 분포를 파악함으로써, 데이터셋이 어떻게 구성되어 있는지, 값들이 얼마나 고르게 분포하는지 등을 알 수 있습니다.	고유값들이 극단적으로 분포한 경우, 데이터셋에 이상값이 포함되었는지 확인할 수 있습니다.	변수의 고유값 분포는 어떤 변수가 예측 모델에서 중요한 역할을 할 수 있는지 미리 평가할 수 있게 해줍니다	각 변수의 고유값 분포를 통해 데이터가 불균형하게 분포되어 있는지를 쉽게 확인할 수 있습니다. 예를 들어, 특정 값이 전체 데이터의 대부분을 차지할 수 있습니다.
	이상값을 미리 감지하여 데이터 정제를 할 수 있고, 잘못된 분석 결과를 방지할 수 있습니다.		



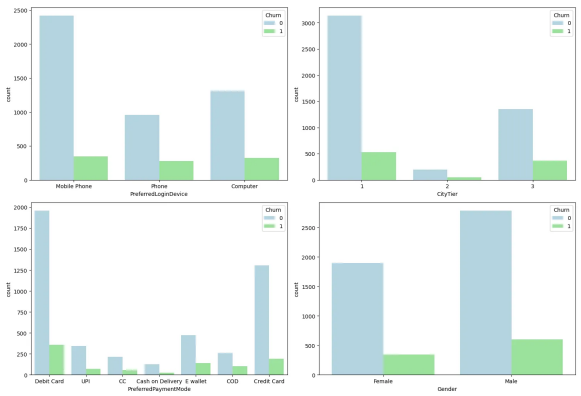


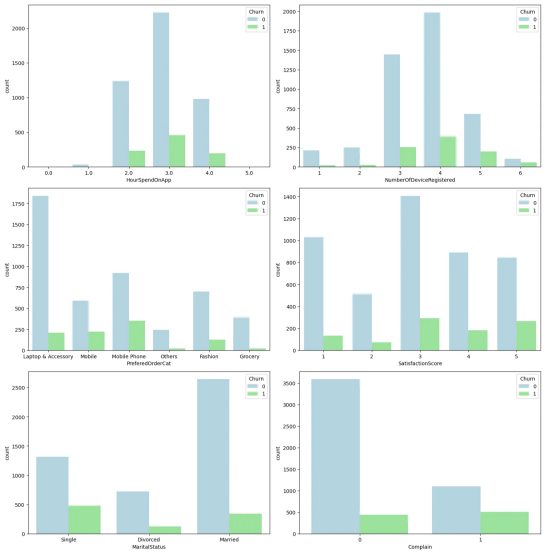




- 각 변수의 고유값을 0과 1로 나눠 그래프를 그림

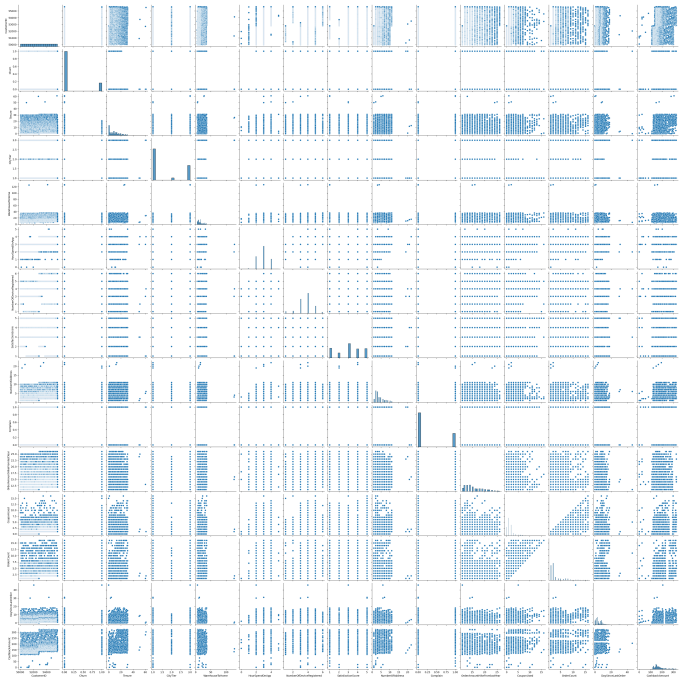
이탈 고객과 유지 고객의 분포 차이	변수의 예측력 평가
각 변수에 대해 이탈한 고객(1)과 이탈하지 않은 고객(0)의 분포를 직접 비교할 수 있습니다.	어떤 변수가 고객 이탈을 예측하는 데 중요한지 파악할 수 있습니다. 만약 이탈 여부에 따라 특정 변수의 값 분포가 크게 다르면, 그 변수는 중요한 예측 변수로 사용할 수 있습니다.
만약 어떤 변수에서 이탈 고객의 특정 값이 유독 높거나 낮다면, 그 변수가 고객 이탈에 중요한 영향을 미치는 변수일 수 있습니다.	





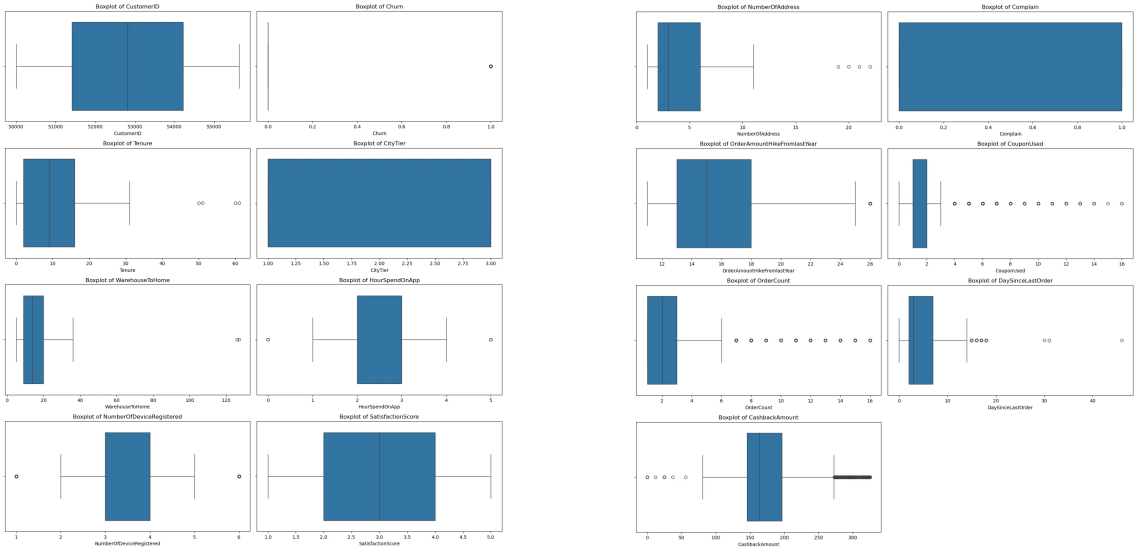
- **pairplot**

변수 간의 패턴과 분포 확인	범주형 변수와 결합하여 분석 가능
대각선에 각 변수의 **히스토그램**이 나타나고, 나머지 영역에는 변수들 간의 **산점도**가 나타납니다. 이를 통해 변수의 개별 분포와 쌍별 관계를 동시에 파악할 수 있습니다.	`pairplot` 에서 **hue** 인자를 사용하여 **범주형 변수**(예: `Churn`)에 따라 색깔을 다르게 설정할 수 있습니다. 이를 통해 범주에 따른 분포 차이나 패턴을 확인할 수 있습니다.
데이터가 어떻게 분포되어 있는지(정규 분포인지, 왜곡된 분포인지)를 확인할 수 있고, 변수 간의 패턴을 쉽게 볼 수 있습니다.	



- **Boxplot**

데이터의 분포와 중심 경향 파악	데이터 간 비교 용이	이상값(outliers) 탐지
박스플롯은 데이터의 **중앙값**, **사분위 범위**(IQR, Interquartile Range), **최댓값**, **최솟값**을 시각적으로 표현해 줍니다.	여러 그룹의 데이터를 나란히 그려 비교할 수 있습니다. 예를 들어, **범주형 변수에 따른 숫자형 변수의 분포 차이**를 쉽게 시각화할 수 있습니다.	박스플롯은 데이터의 이상값을 시각적으로 표시하여, 일반적인 분포에서 벗어난 값들을 쉽게 식별할 수 있습니다.
데이터가 **대칭적인지** 또는 **왜곡된 분포**(skewed distribution)를 가지고 있는지 쉽게 알 수 있습니다.		



3. 전처리

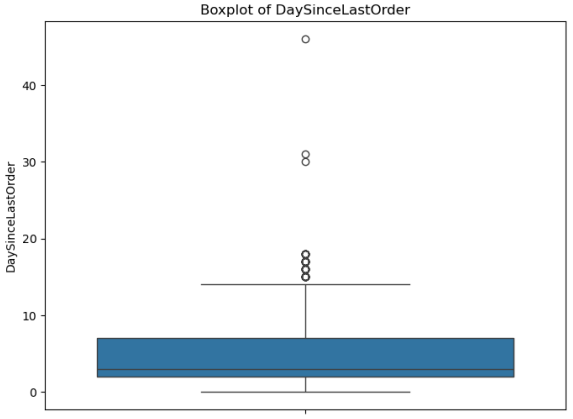
결측치 처리

```
import numpy as np

np.random.seed(42)  # 시드 고정
df['Tenure'].fillna(np.random.randint(61,72), inplace=True)
df['WarehouseToHome'].fillna(0, inplace=True)

df = df[df['HourSpendOnApp'].notna()]
df.drop(columns=['OrderAmountHikeFromlastYear'], inplace=True)
df['CouponUsed'].fillna(0, inplace=True)
df['OrderCount'].fillna(0, inplace=True)
df = df[df['DaySinceLastOrder'].notna()]
df = df[~df['DaySinceLastOrder'].isin([31, 46, 30])]
```

변수명	결측값 처리 방법	결측값 처리 방법 - 이유
Tenure	61과 71 사이의 임의의 대치	60개월 이후의 숫자가 없어서 사이트가 생기기 전에 가입되어 있던 고객일 가능성 有
WarehouseToHome	결측값을 모두 0으로 대치	자료에 0 의 값이 없어 매장 방문을 하지 않아 결측값으로 표시 될거라 판단
HourSpendOnApp	결측값이 있는 행을 제거	자료에 0이 이미 있어, 추정 불가로 행 제거
OrderAmountHikeFromlastYear	열을 데이터 프레임에서 제거	증가량이 미미하며, 감소한 추세가 보이지 않아 열 자체를 제거
CouponUsed	결측값을 모두 0으로 대치	자료에 0 의 값이 없어 쿠폰을 사용 하지 않아 결측값으로 표시 될거라 판단
DaySinceLastOrder	결측값이 있는 행을 제거	자료에 0이 이미 있어, 추정 불가로 행 제거
DaySinceLastOrder	30,31,46 인 행을 필러링 후 제거	boxplot 을 확인하여 봤을 때 30 이상의 수는 극단적 이상치라 성능 저하가 될 수 있다 판단



데이터 타입 변경

- **라벨 인코더**

```
abel_encoders = {
    'PreferredLoginDevice': LabelEncoder(),
    'PreferredPaymentMode': LabelEncoder(),
    'Gender': LabelEncoder(),
    'PreferedOrderCat': LabelEncoder(),
    'MaritalStatus': LabelEncoder()
}

for col, encoder in label_encoders.items():
    df[col] = encoder.fit_transform(df[col])
```

변수명	라벨 인코딩 후
PreferredLoginDevice	0~2
PreferredLoginDevice	0~6
Gender	0~1
PreferedOrderCat	0~5
MaritalStatus	0~2

- **스케일러 진행**

```
scalers = {}
numeric_columns = ['CityTier', 'WarehouseToHome', 'HourSpendOnApp',
                   'NumberOfDeviceRegistered', 'SatisfactionScore', 'Num
                    'CouponUsed', 'OrderCount','DaySinceLastOrder', 'Cas
for col in numeric_columns:
    scaler = StandardScaler()
    # fit_transform을 사용하여 데이터 스케일링
    df[col] = scaler.fit_transform(df[[col]])  # DataFrame 형식으로 전달
    # 각 스케일러를 저장
    scalers[col] = scaler
```

	스케일러 진행한 변수	
CashbackAmount	CityTier	WarehouseToHome
HourSpendOnApp	NumberOfDeviceRegistered	SatisfactionScore
NumberOfAddress	Complain	CouponUsed
OrderCount	DaySinceLastOrder	