

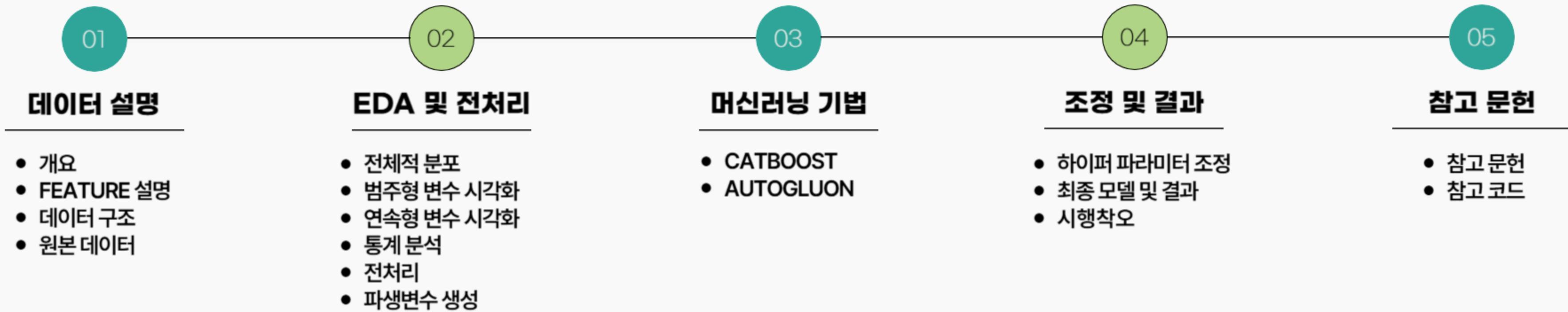
아이엠뱅크 2차 프로젝트

머신러닝(ML) 성능 극대화 프로젝트

송현서

2024-12-11

목차



데이터 설명

개요



Binary Classification with a Bank Churn Dataset

Playground Series - Season 4, Episode 1
2024년 첫 번째로 개최된 Platground Series.

다른 원본 데이터를 기반으로 딥러닝 모델을 통해 생성된 데이터(21.65mb)를 기반으로 고객의 이탈 여부의 예측 목표.
평가 지표: ROC-AUC

개요

평가지표: ROC-AUC

- ROC (Receiver Operating Characteristic) 곡선:

이진 분류 문제에서 모델의 분류 성능을 평가하는 시각화 도구.

X축: FPR (False Positive Rate) - 잘못된 경보율.

Y축: TPR (True Positive Rate) - 민감도(정답 비율).

분류기의 판단 기준(Threshold)에 따라 X축과 Y축 값이 변화하며 곡선이 그려짐.

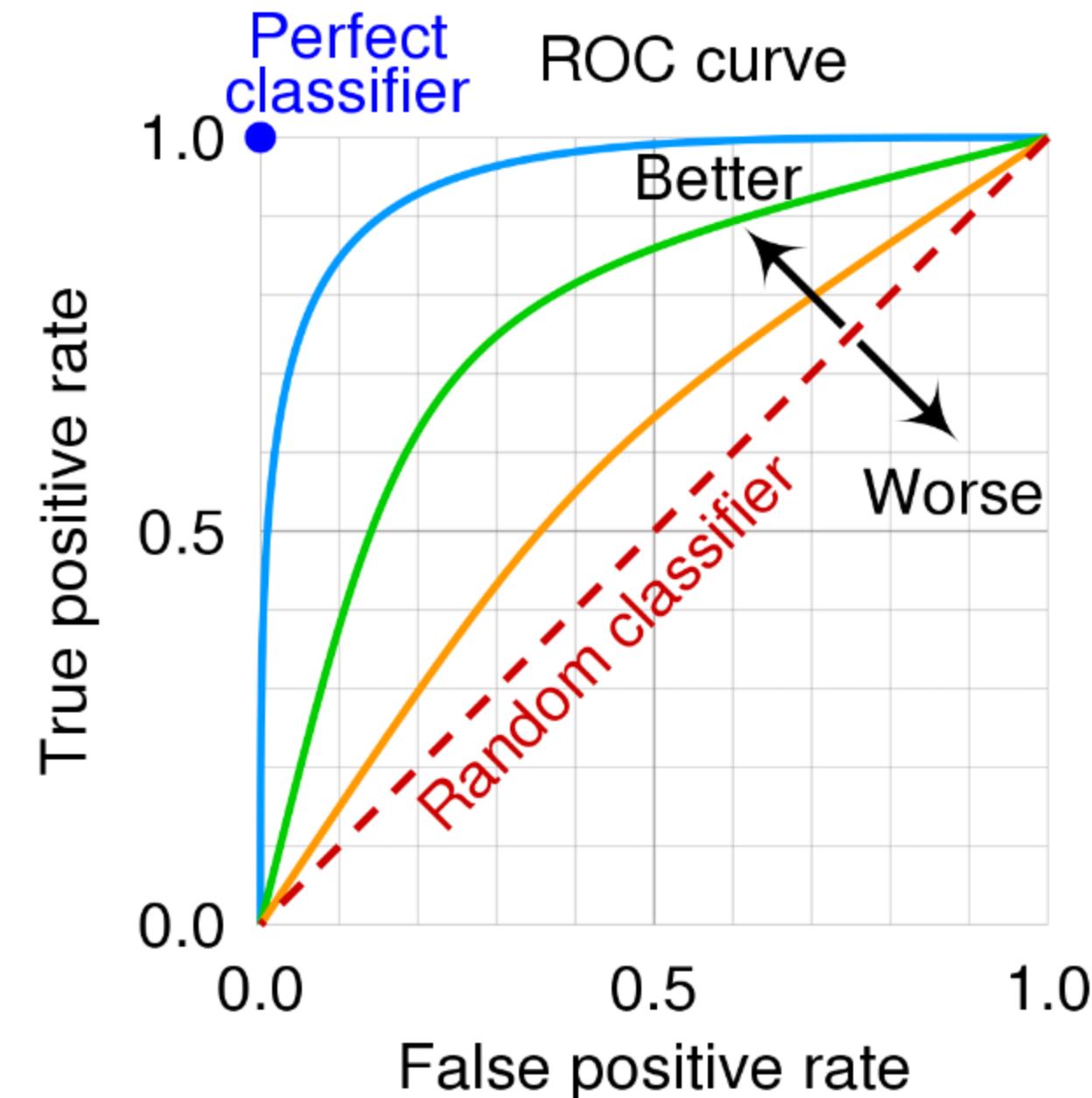
- AUC (Area Under the Curve):

ROC 곡선 아래 면적.

0~1 사이의 값으로, 1에 가까울수록 모델 성능이 우수함.

모델의 종합적인 성능을 한눈에 확인할 수 있는 값.

- 모델 비교가 쉬워 평가지표로 자주 쓰인다.



데이터 설명

Feature 설명

- Train

| | id | CustomerId | Surname | CreditScore | Geography | Gender | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited |
|--------|-----------|-------------------|----------------|--------------------|------------------|---------------|------------|---------------|----------------|----------------------|------------------|-----------------------|------------------------|---------------|
| 0 | 0 | 15674932 | Okwudilichukwu | 668 | France | Male | 33.0 | 3 | 0.00 | 2 | 1.0 | 0.0 | 181449.97 | 0 |
| 1 | 1 | 15749177 | Okwudiliolisa | 627 | France | Male | 33.0 | 1 | 0.00 | 2 | 1.0 | 1.0 | 49503.50 | 0 |
| 2 | 2 | 15694510 | Hsueh | 678 | France | Male | 40.0 | 10 | 0.00 | 2 | 1.0 | 0.0 | 184866.69 | 0 |
| 3 | 3 | 15741417 | Kao | 581 | France | Male | 34.0 | 2 | 148882.54 | 1 | 1.0 | 1.0 | 84560.88 | 0 |
| 4 | 4 | 15766172 | Chiemenam | 716 | Spain | Male | 33.0 | 5 | 0.00 | 2 | 1.0 | 1.0 | 15068.83 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 165029 | 165029 | 15667085 | Meng | 667 | Spain | Female | 33.0 | 2 | 0.00 | 1 | 1.0 | 1.0 | 131834.75 | 0 |
| 165030 | 165030 | 15665521 | Okechukwu | 792 | France | Male | 35.0 | 3 | 0.00 | 1 | 0.0 | 0.0 | 131834.45 | 0 |
| 165031 | 165031 | 15664752 | Hsia | 565 | France | Male | 31.0 | 5 | 0.00 | 1 | 1.0 | 1.0 | 127429.56 | 0 |
| 165032 | 165032 | 15689614 | Hsiung | 554 | Spain | Female | 30.0 | 7 | 161533.00 | 1 | 0.0 | 1.0 | 71173.03 | 0 |
| 165033 | 165033 | 15732798 | Ulyanov | 850 | France | Male | 31.0 | 1 | 0.00 | 1 | 1.0 | 0.0 | 61581.79 | 1 |

Data Shape: (165034, 14)

CustomerId: 각 고객마다 고유한 ID

Surname: 고객의 성

CreditScore: 고객의 신용 점수의 수치화

Geography: 고객의 거주 국가

Gender: 고객의 성별

Age: 고객의 나이

Tenure: 고객이 은행과 거래한 기간(년)

Balance: 고객의 계좌 잔액

NumOfProducts: 고객이 사용하는 은행상품 수

HasCrCard: 고객의 신용카드 보유 여부

IsActiveMember: 고객의 활동 고객 여부

EstimatedSalary: 고객의 예측 소득

Exited: 고객 이탈 여부 (Target)

1 = yes, 0 = no

데이터 설명

데이터 구조

`['Surname', 'Geography', 'Gender',
'HasCrCard', 'IsActiveMember']`

범주형 데이터

이 중 'Surname', 'Geography', 'Gender'는 문자로 이루어진 범주형 변수
이지만 'HasCrCard', 'IsActiveMember'의 경우 1과 0으로 이루어진
이진 범주형 변수이다. 특히, Surname의 경우 높은 Cardinality를 보일
가능성이 크다.

`['id', 'CustomerId', 'CreditScore', 'Age', 'Tenure', 'Balance',
'NumOfProducts', 'EstimatedSalary', 'Exited']`

연속형 데이터

연속형 변수의 경우 'id'는 의미가 없으므로 제거하는 편이 좋고, 'Exited' 역시
Target이므로 제거 후 학습을 하는 편이 좋다. 이외의 데이터는 사용할 모델에
따라 범주화 등 처리를 할 수 있다.

범주형 데이터가 많고, 범주형 데이터의 Cardinality가 높을 것으로
예상되므로, 적절한 전처리 및 파생 변수 생성,
그리고 모델 선정이 중요하다!

데이터 설명

원본 데이터



| RowNumber | CustomerId | Surname | CreditScore | Geography | Gender | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited | |
|-----------|------------|----------|-------------|-----------|---------|--------|--------|---------|---------------|-----------|----------------|-----------------|-----------|---|
| 0 | 1 | 15634602 | Hargrave | 619 | France | Female | 42 | 2 | 0.00 | 1 | 1 | 1 | 101348.88 | 1 |
| 1 | 2 | 15647311 | Hill | 608 | Spain | Female | 41 | 1 | 83807.86 | 1 | 0 | 1 | 112542.58 | 0 |
| 2 | 3 | 15619304 | Onio | 502 | France | Female | 42 | 8 | 159660.80 | 3 | 1 | 0 | 113931.57 | 1 |
| 3 | 4 | 15701354 | Boni | 699 | France | Female | 39 | 1 | 0.00 | 2 | 0 | 0 | 93826.63 | 0 |
| 4 | 5 | 15737888 | Mitchell | 850 | Spain | Female | 43 | 2 | 125510.82 | 1 | 1 | 1 | 79084.10 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| 9995 | 9996 | 15606229 | Obijiaku | 771 | France | Male | 39 | 5 | 0.00 | 2 | 1 | 0 | 96270.64 | 0 |
| 9996 | 9997 | 15569892 | Johnstone | 516 | France | Male | 35 | 10 | 57369.61 | 1 | 1 | 1 | 101699.77 | 0 |
| 9997 | 9998 | 15584532 | Liu | 709 | France | Female | 36 | 7 | 0.00 | 1 | 0 | 1 | 42085.58 | 1 |
| 9998 | 9999 | 15682355 | Sabbatini | 772 | Germany | Male | 42 | 3 | 75075.31 | 2 | 1 | 0 | 92888.52 | 1 |
| 9999 | 10000 | 15628319 | Walker | 792 | France | Female | 28 | 4 | 130142.79 | 1 | 1 | 0 | 38190.78 | 0 |

Bank Customer Churn Prediction

Binary Classification with a Bank Churn Dataset에서 사용된 데이터의 원본 데이터로, 해당 시리즈에서 사용된 데이터는 이 데이터를 기반으로 딥 러닝 모델을 통해 제작되었다.

즉, 이 데이터를 사용하면 데이터의 학습량을 늘릴 수 있어 유의미한 학습효과 향상이 기대된다.

EDA 및 전처리

EDA

기초통계

`['Surname', 'Geography', 'Gender',
 'HasCrCard', 'IsActiveMember']`

범주형 데이터

`['id', 'CustomerId', 'CreditScore', 'Age', 'Tenure', 'Balance',
 'NumOfProducts', 'EstimatedSalary', 'Exited']`

연속형 데이터

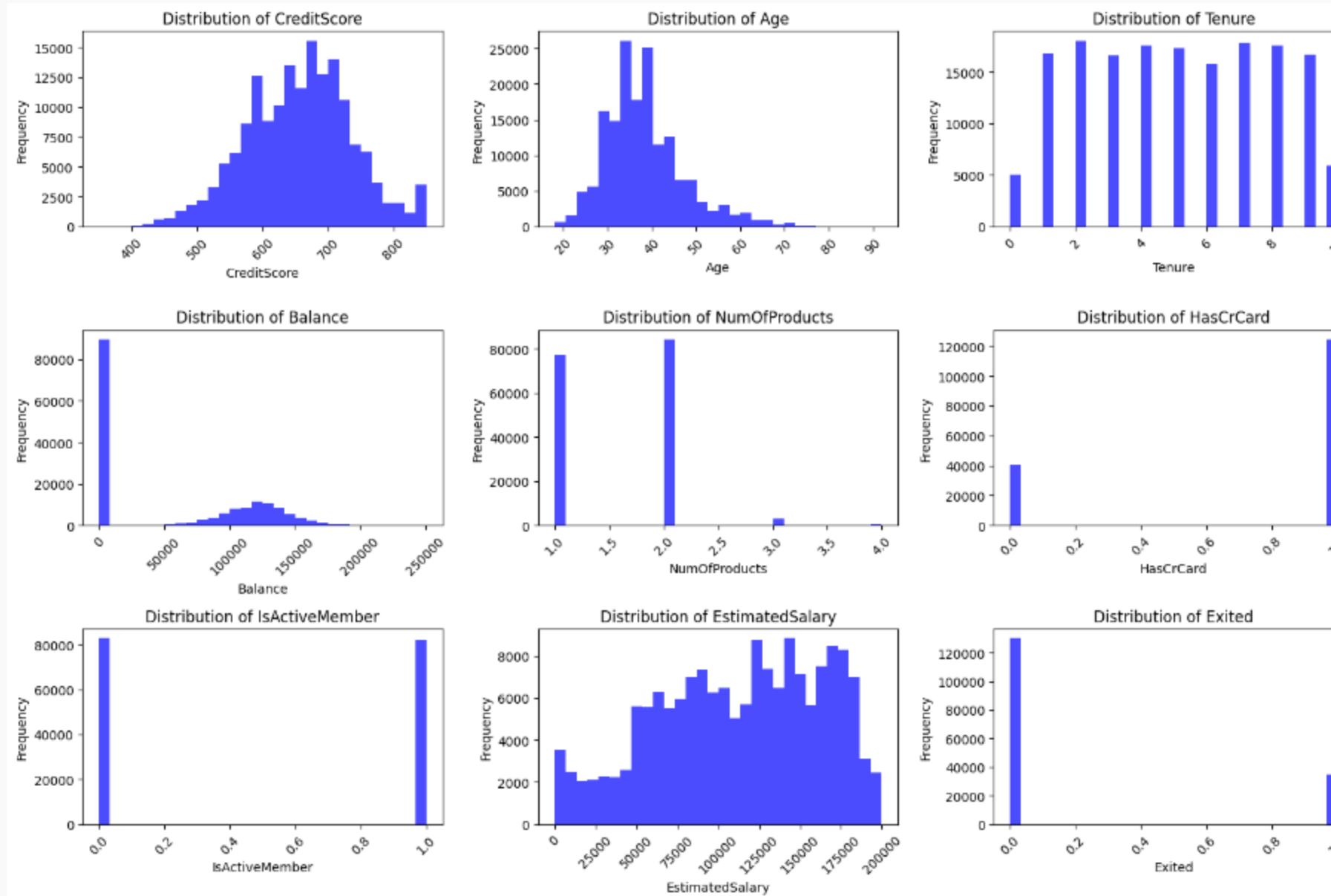
| | | type | count | unique_num | nulls |
|----------------|--|---------|--------|------------|-------|
| Surname | | object | 165034 | 2797 | 0 |
| Geography | | object | 165034 | 3 | 0 |
| Gender | | object | 165034 | 2 | 0 |
| HasCrCard | | float64 | 165034 | 2 | 0 |
| IsActiveMember | | float64 | 165034 | 2 | 0 |

| | | type | count | unique_num | mean | min | median | max | nulls |
|--|-----------------|---------|--------|------------|--------------|-------------|------------|-------------|-------|
| | id | int64 | 165034 | 165034 | 8.251650e+04 | 0.00 | 82516.5 | 165033.00 | 0 |
| | CustomerId | int64 | 165034 | 23221 | 1.569201e+07 | 15565701.00 | 15690169.0 | 15815690.00 | 0 |
| | CreditScore | int64 | 165034 | 457 | 6.564544e+02 | 350.00 | 659.0 | 850.00 | 0 |
| | Age | float64 | 165034 | 71 | 3.812589e+01 | 18.00 | 37.0 | 92.00 | 0 |
| | Tenure | int64 | 165034 | 11 | 5.020353e+00 | 0.00 | 5.0 | 10.00 | 0 |
| | Balance | float64 | 165034 | 30075 | 5.547809e+04 | 0.00 | 0.0 | 250898.09 | 0 |
| | NumOfProducts | int64 | 165034 | 4 | 1.554455e+00 | 1.00 | 2.0 | 4.00 | 0 |
| | EstimatedSalary | float64 | 165034 | 55298 | 1.125748e+05 | 11.58 | 117948.0 | 199992.48 | 0 |

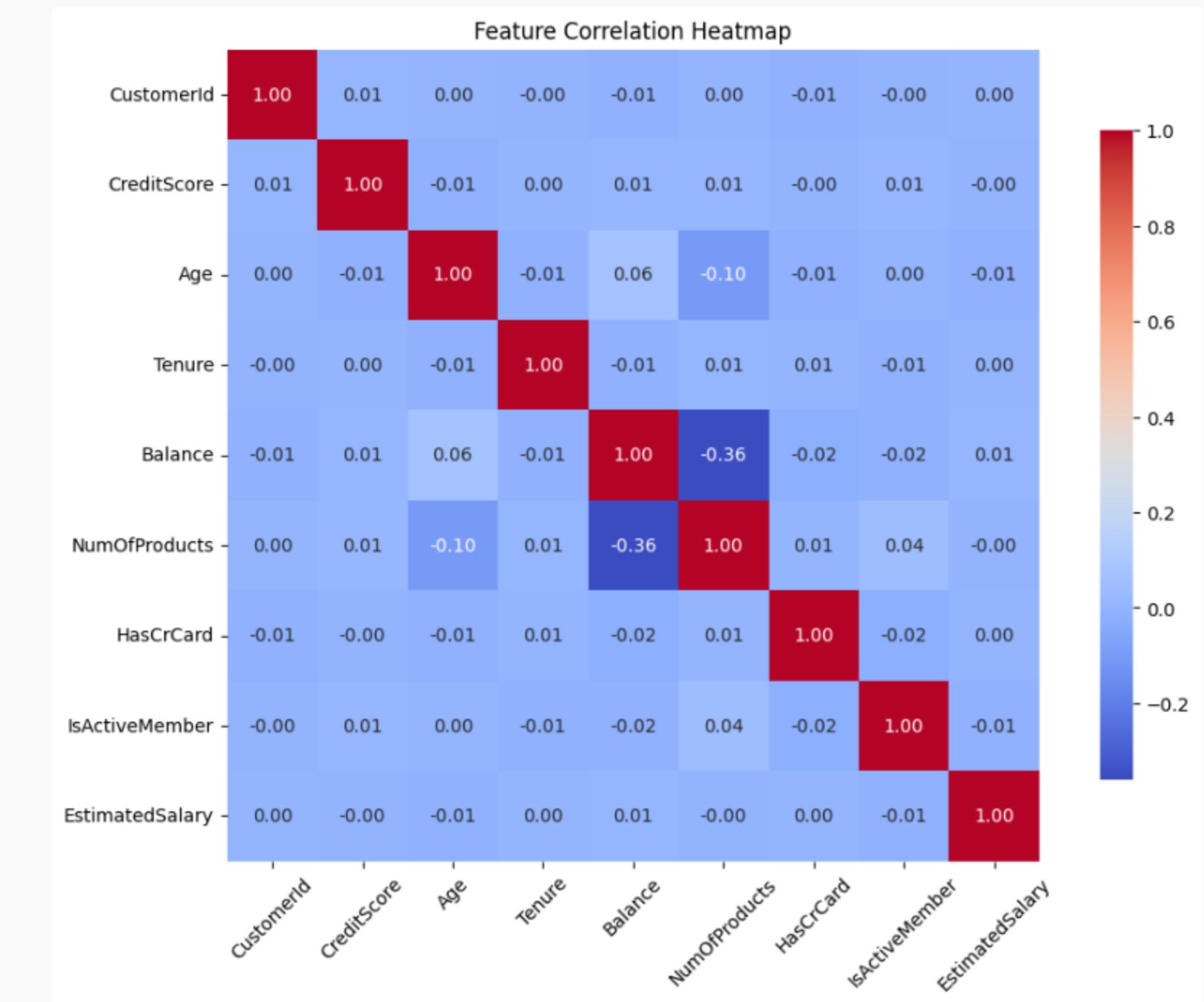
결측치는 관측되지 않으며, 범주형 데이터의 Surname에서 높은 Cardinality가 관측된다.

EDA

전체적 분포 및 상관관계



- 전체 분포를 보니 이상적인 정규분포보다는 **비대칭적인 데이터들이 많이 보인다.**

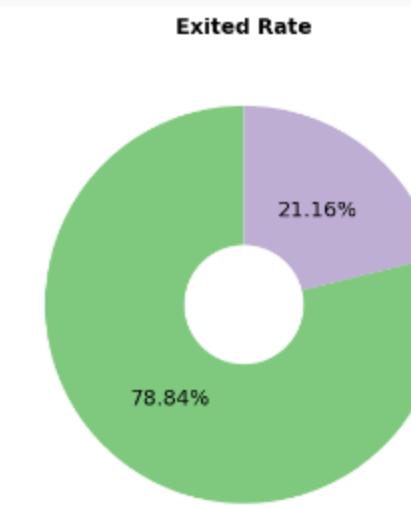
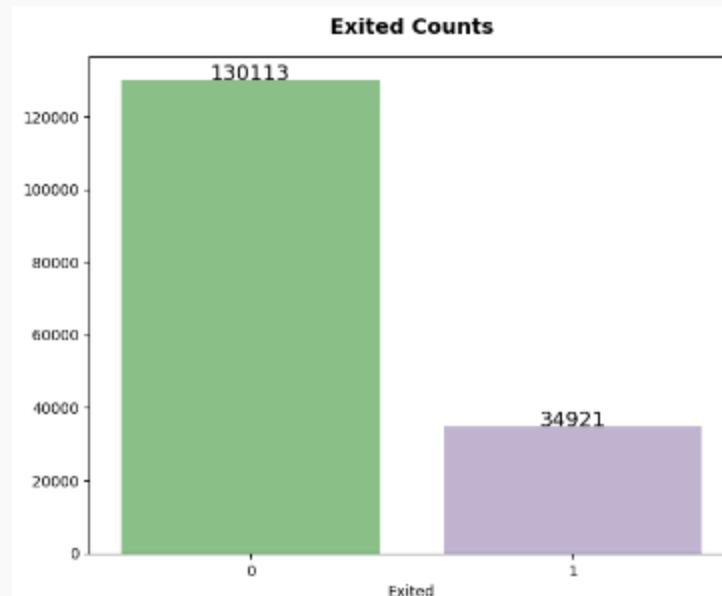


- 상관관계표를 보니 **지나치게 상관관계가 높거나 낮은 관계는 관측되지 않는다.**

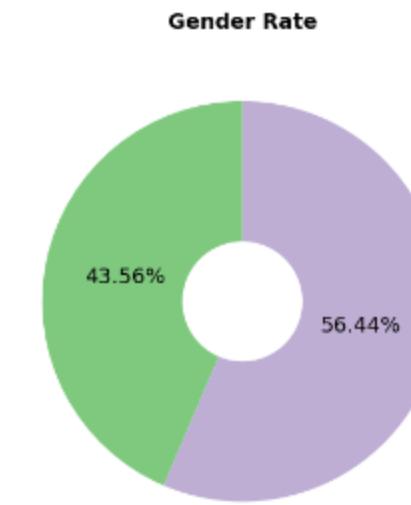
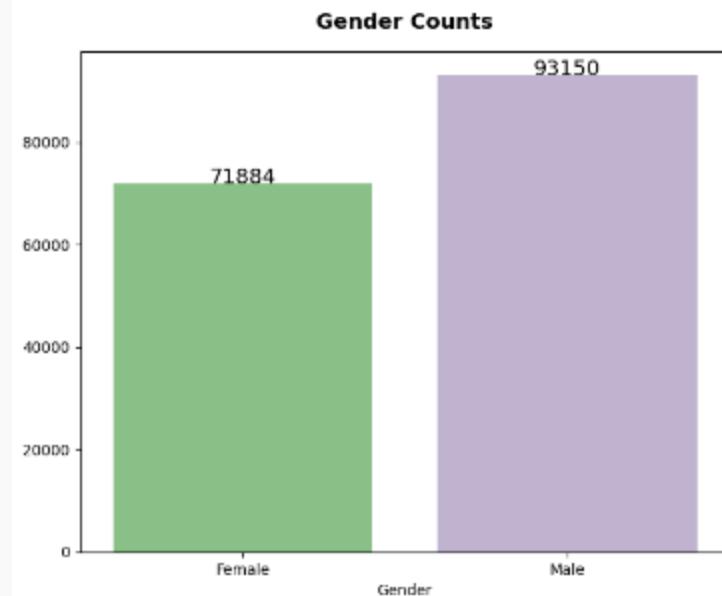
활성

EDA

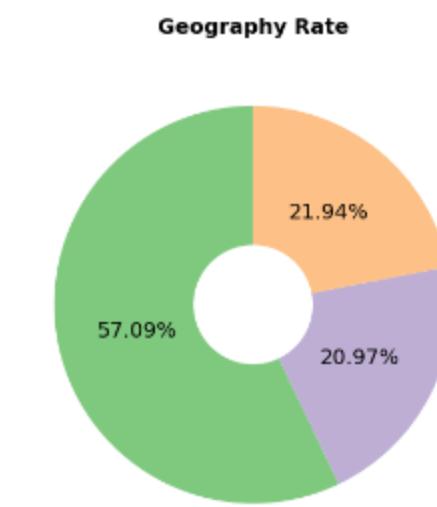
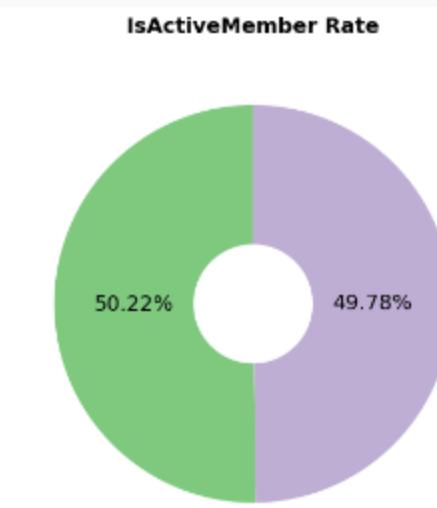
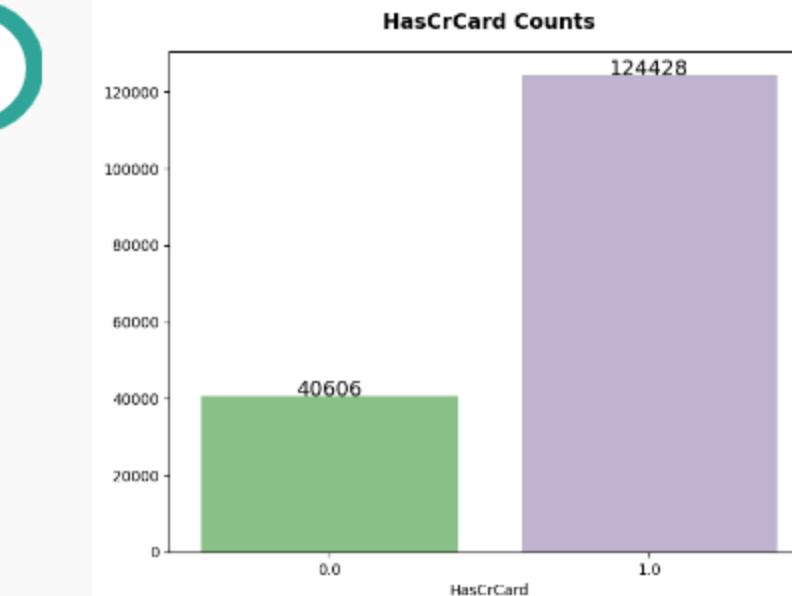
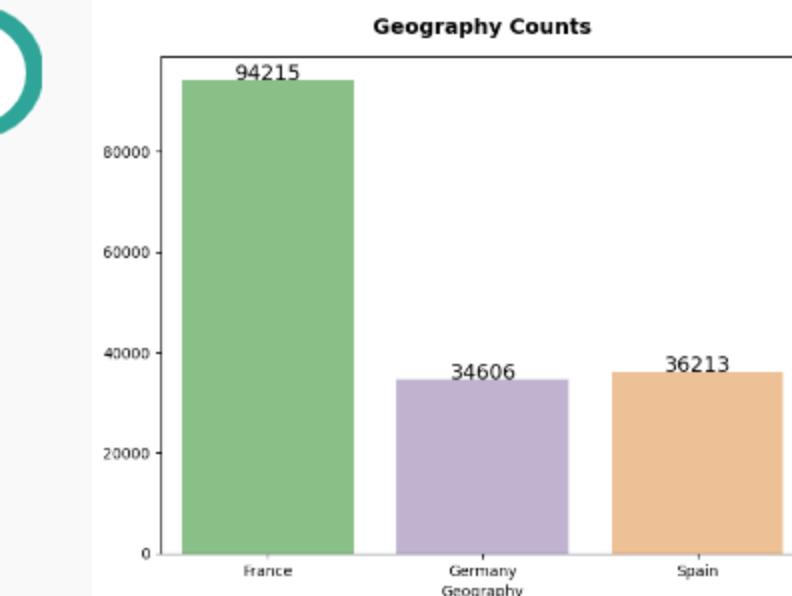
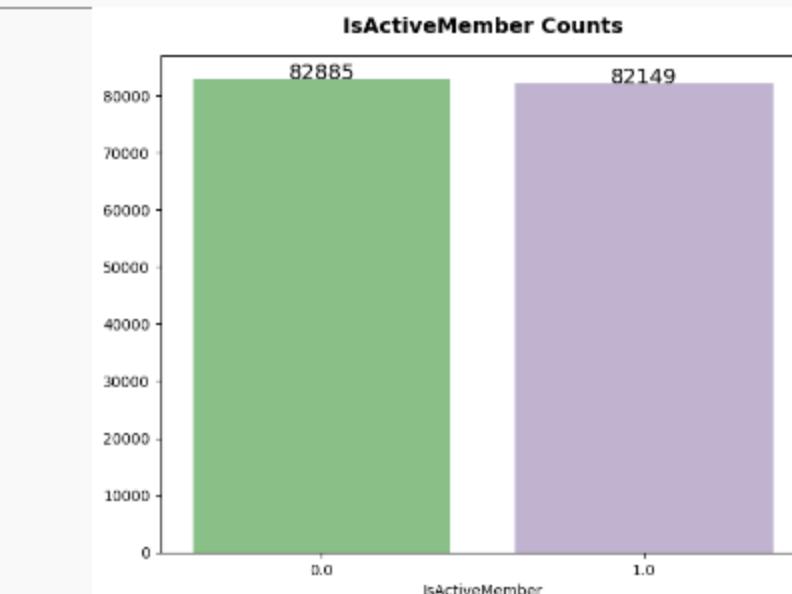
범주형 변수 시각화



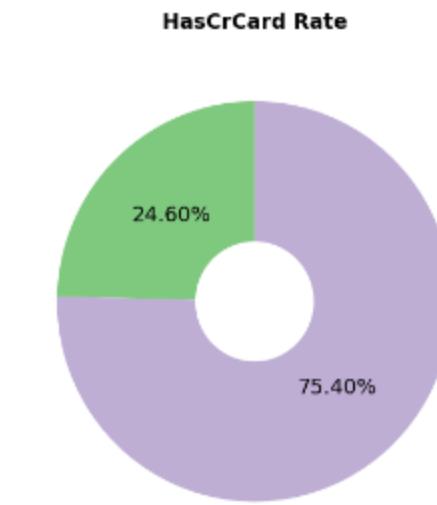
이탈



성별



지역

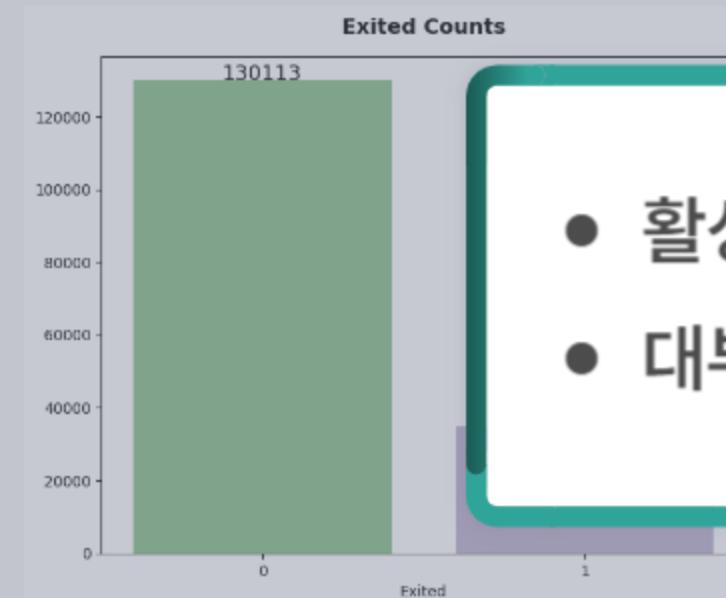


신용카드

활성

EDA

범주형 변수 시각화

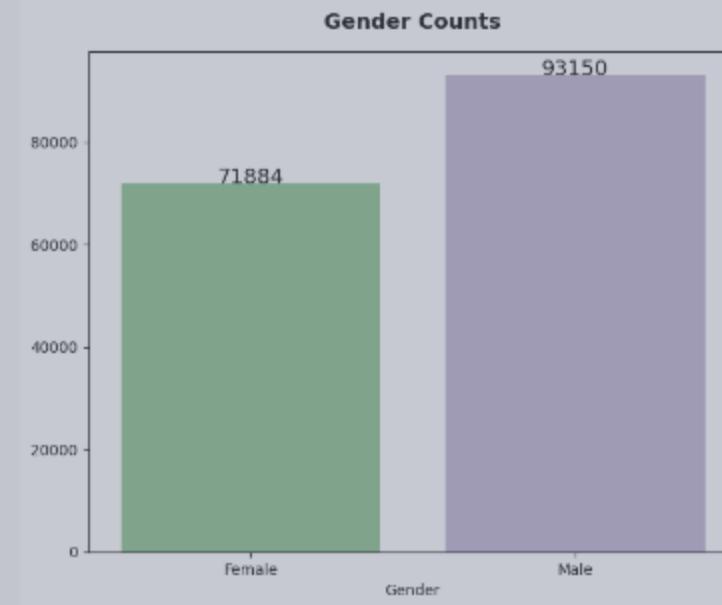
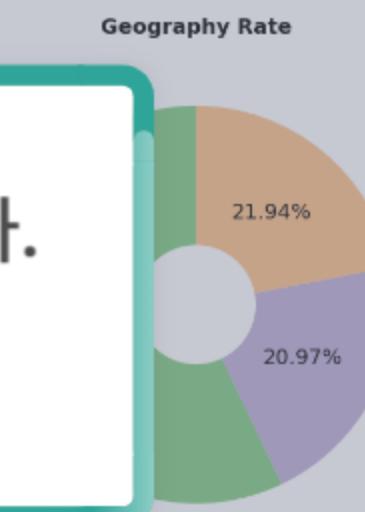
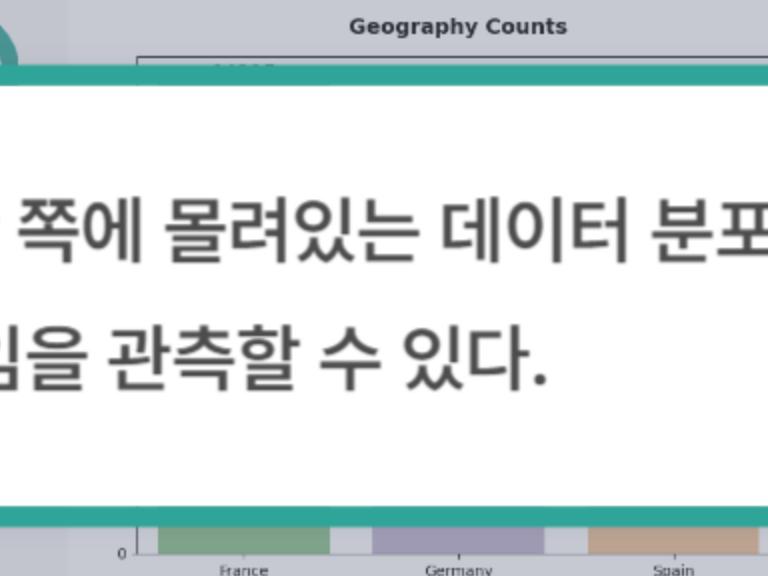
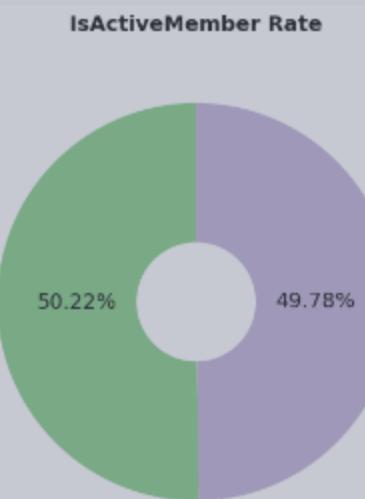
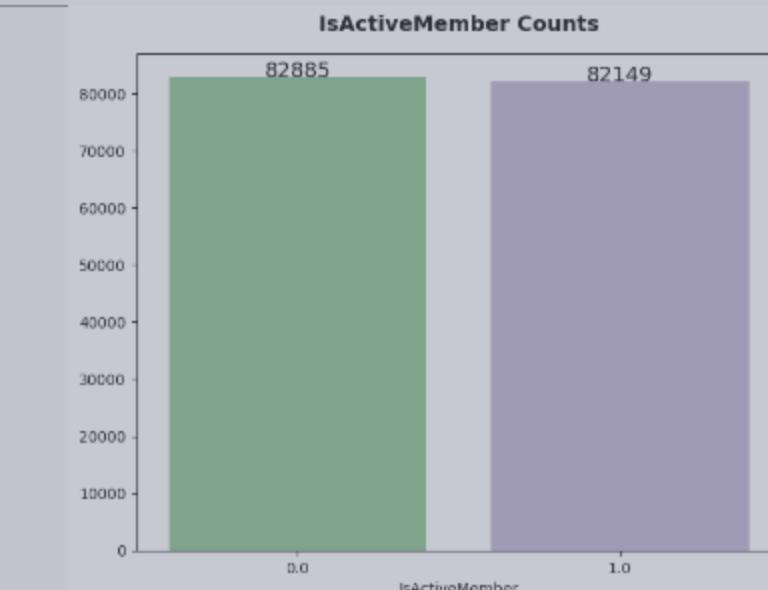


Exited Rate

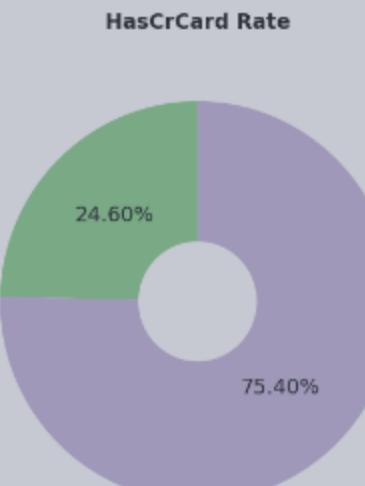
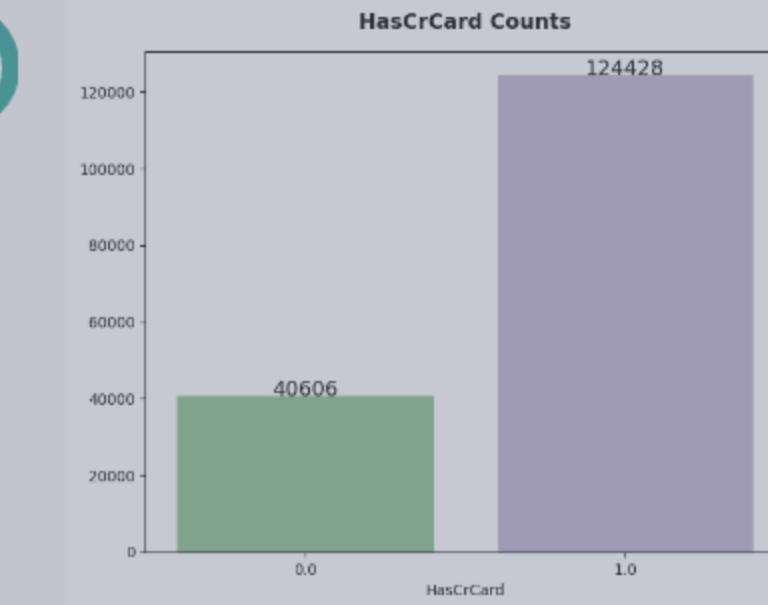
이탈



- 활성 고객 여부, 성별을 제외하곤 한 쪽에 몰려있는 데이터 분포를 보인다.
- 대부분의 범주형 변수가 이진 변수임을 관측할 수 있다.



성별

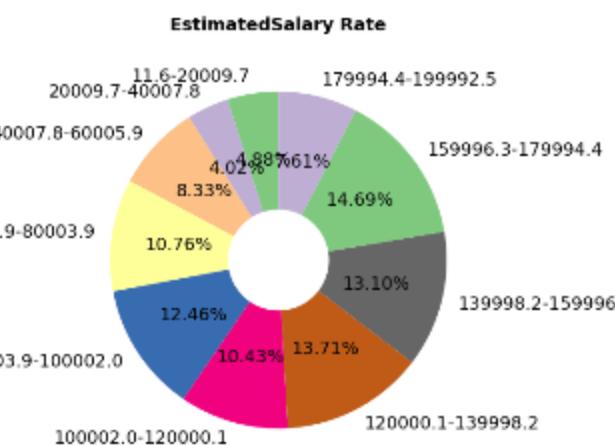
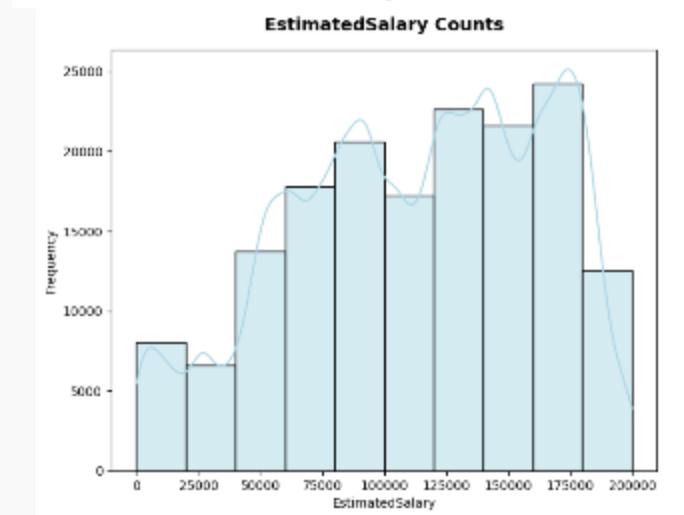
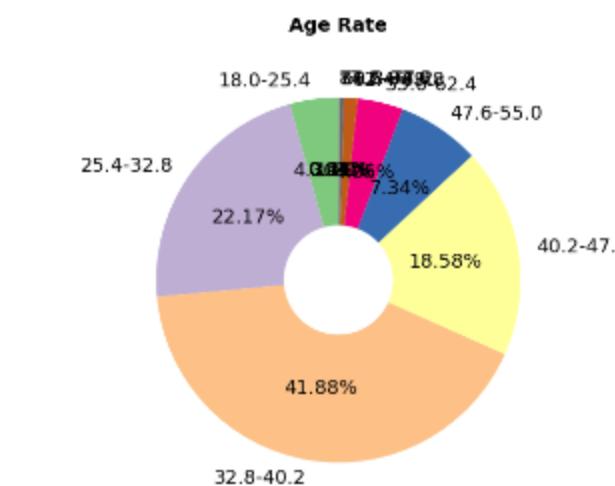
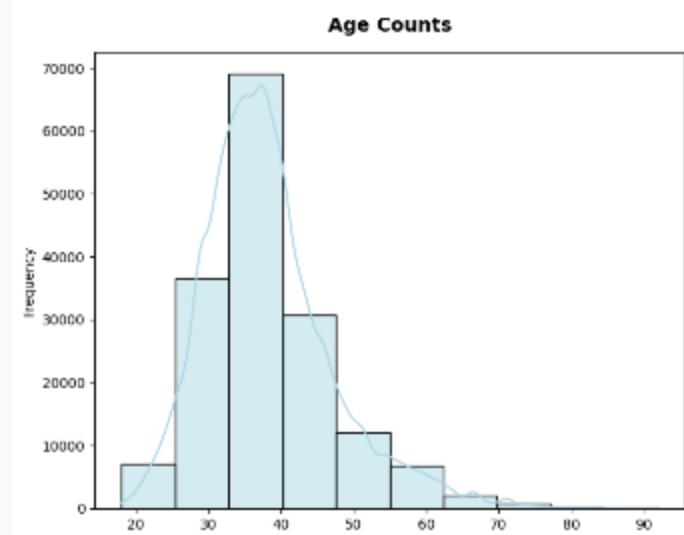
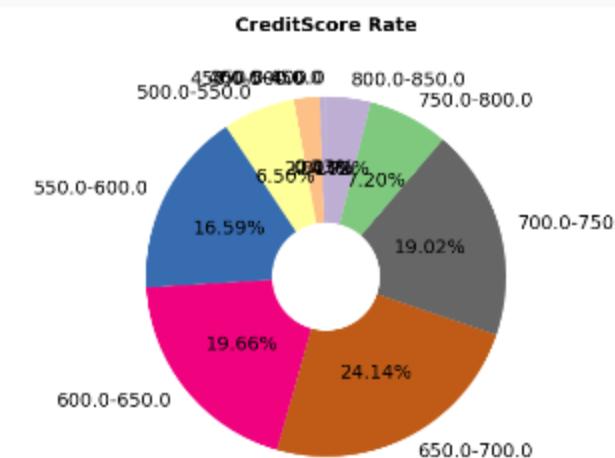
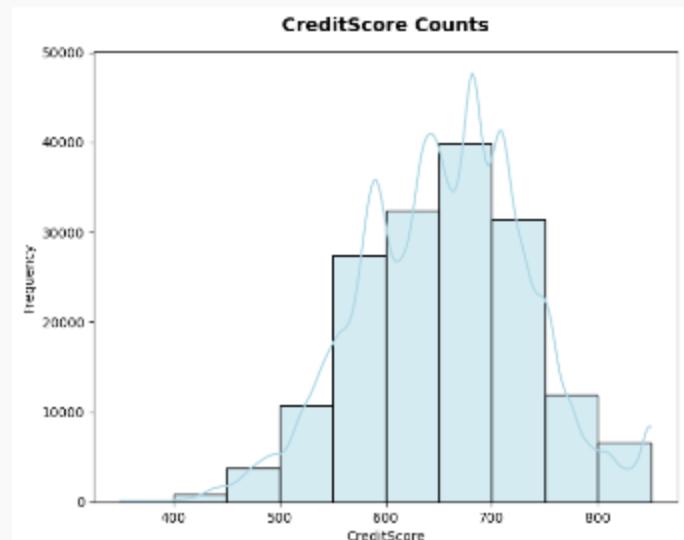


지역

신용카드

연속형 변수 시각화

EDA



신용점수

나이

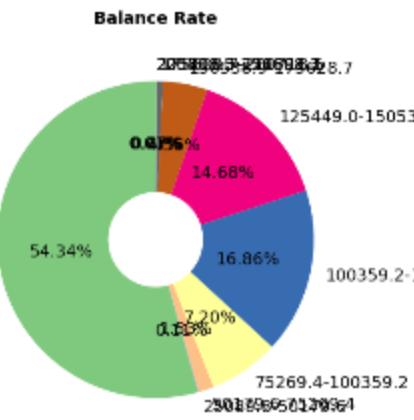
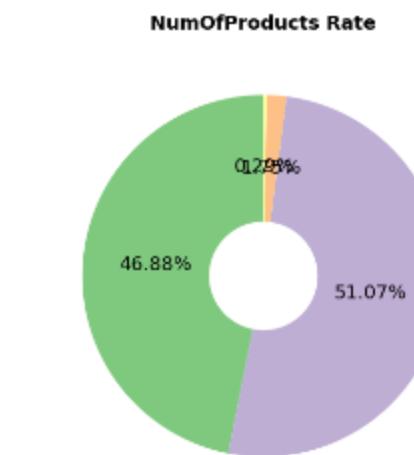
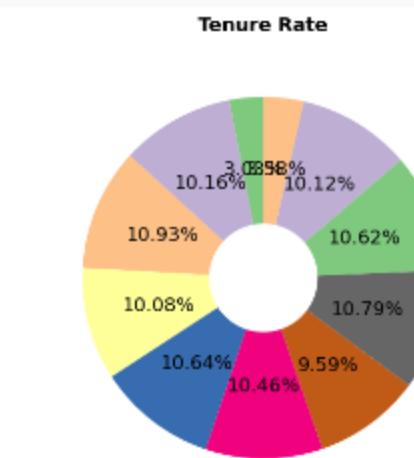
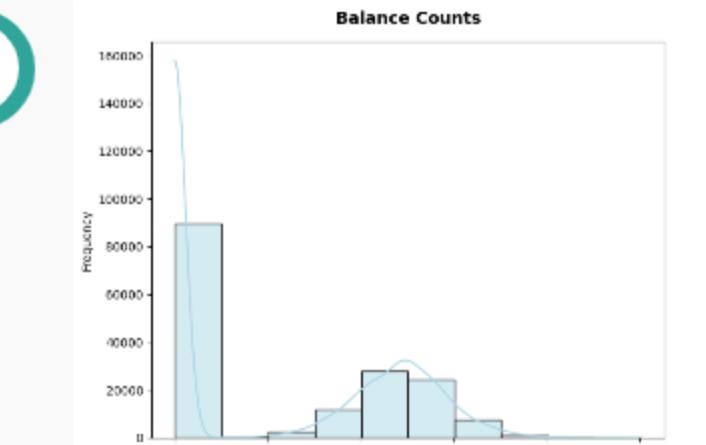
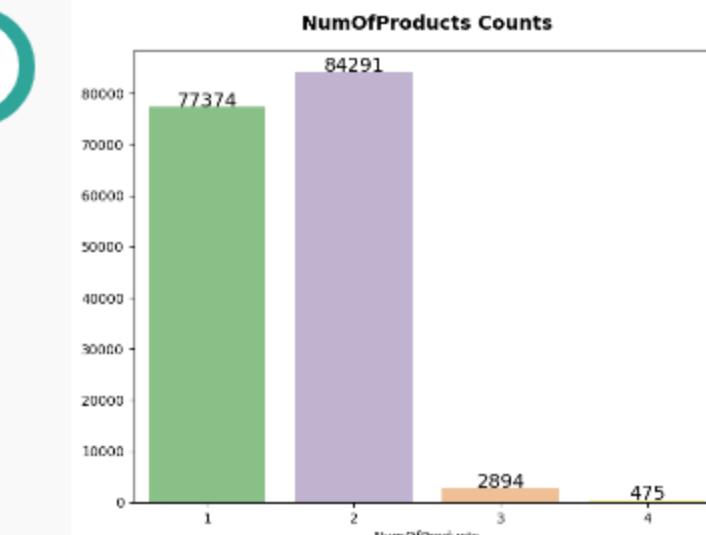
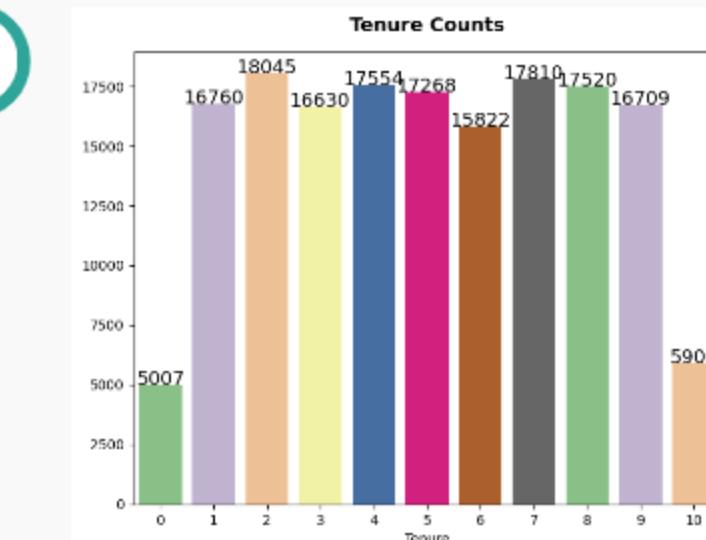
예상소득

EDA 및 전처리

거래기간

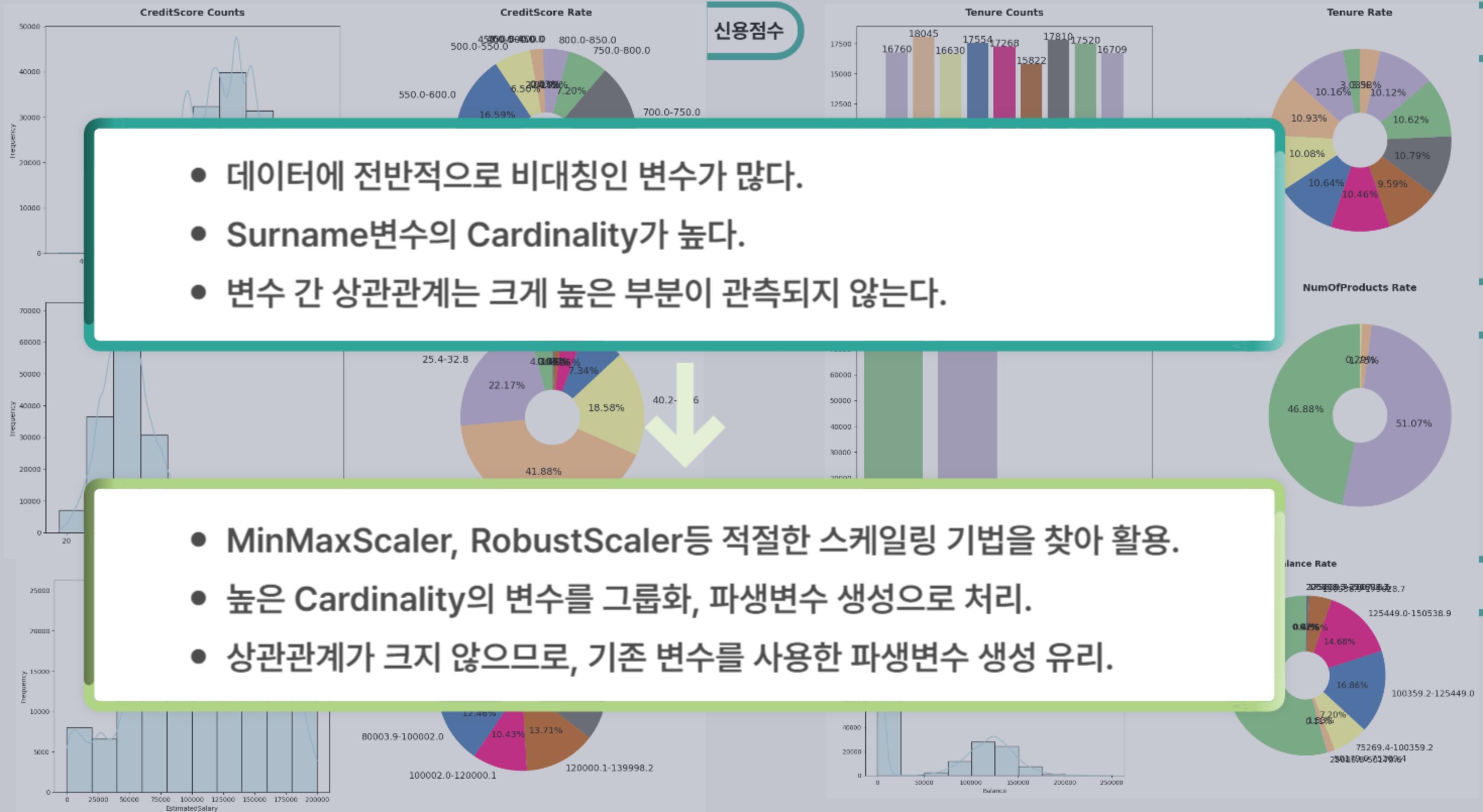
보유상품

계좌잔액



연속형 변수 시각화

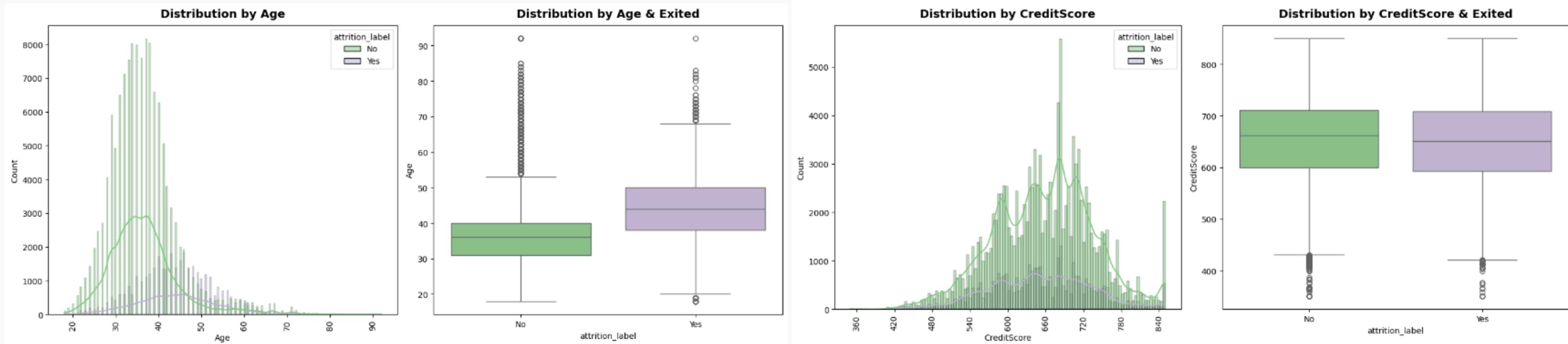
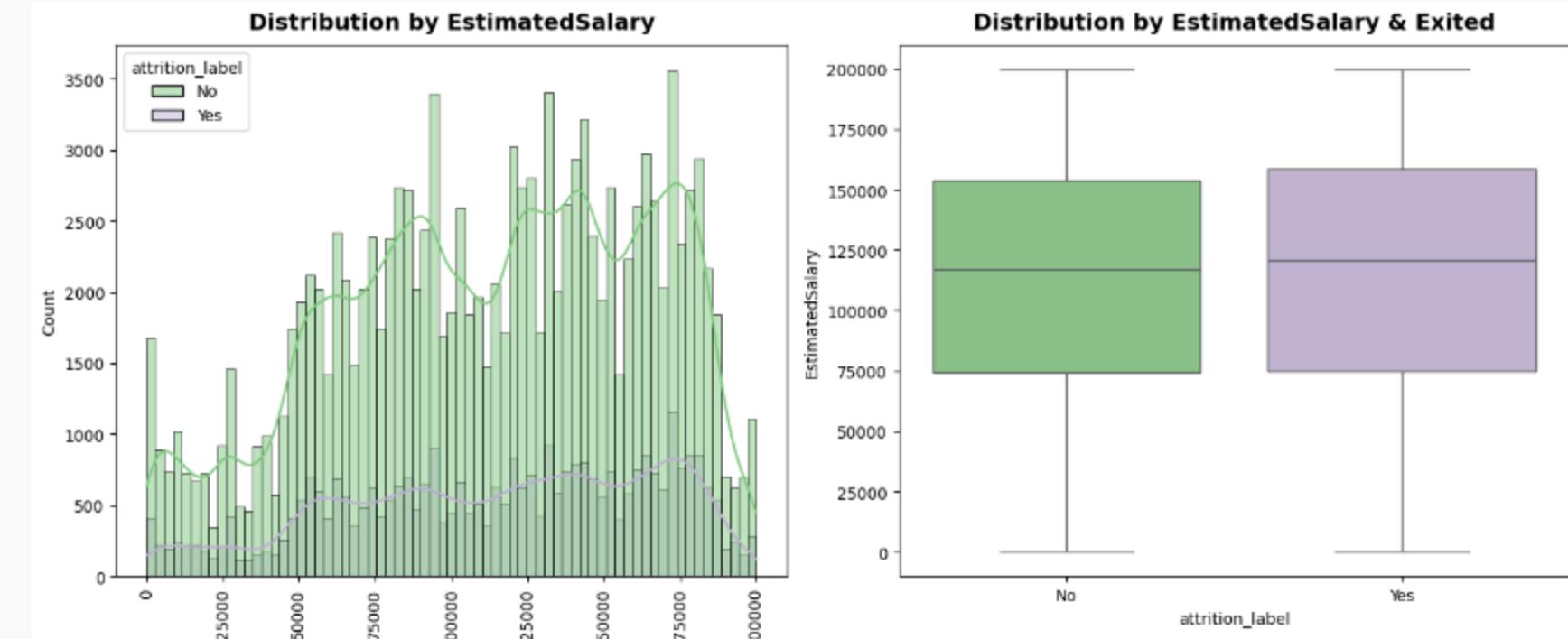
EDA



EDA

이탈 변수 시각화

- 나이가 많은 층에서 이탈이 더 높은 경향을 띤다.
- 예상 소득과는 큰 차이가 없는 것으로 보인다.
- 신용 점수와도 큰 차이가 없는 것으로 보인다.

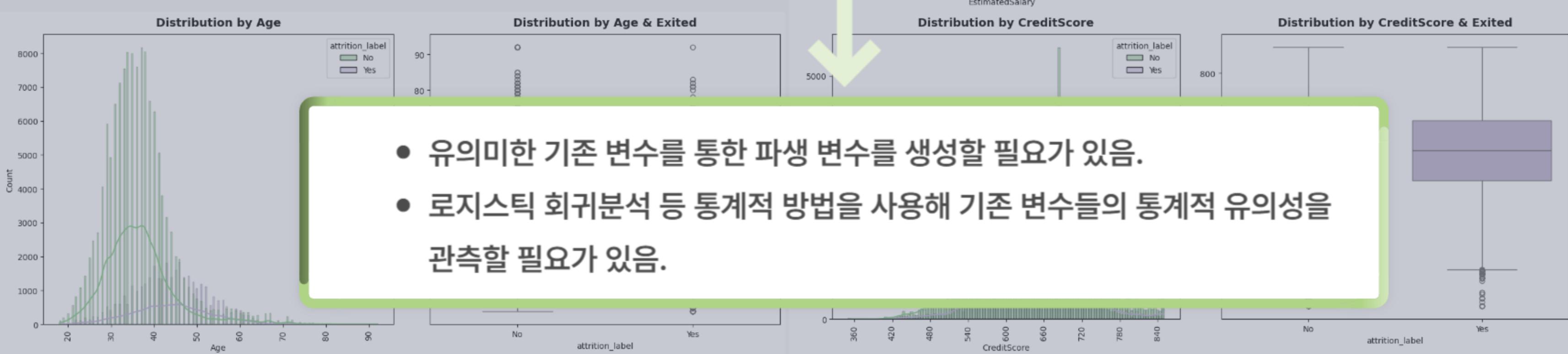


EDA

이탈 변수 시각화

- 나이가 많은 층에서
- 예상 소득과는 큰 차
- 신용 점수와도 큰 차

- 기존의 변수가 이탈 - 비이탈 여부를 명확하게 설명하지 못함
- 나이 변수는 어느 정도 이타 비이탈간의 차이가 관측됨
- 모든 변수가 유의한지 살펴볼 필요가 있음



통계 분석

로지스틱 회귀분석

이진변수(0,1)인 종속변수 Y에 대해

- 귀무가설(H_0): 독립변수가 종속변수에 영향을 미치지 않는다.
- 대립가설(H_1): 독립변수가 종속변수에 영향을 끼친다.

종속변수(Y): Exited / 독립변수(X): 그 외 변수

$$Y \text{ (확률)} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$$

독립변수의 P값이 0.05 이하라면;
귀무가설은 기각되고, 해당 독립변수가
종속변수에 끼치는 영향이 통계적으로 유의하다.

통계 분석

로지스틱 회귀분석

결과 해석

- 분석 결과 모든 변수에서 P값이 0.05 미만임을 알 수 있다.
- 이는 해당 데이터의 모든 변수가 다 종속 변수에 통계적으로 유의미한 영향을 끼치고 있어, 전처리 시 딱히 제거할 변수가 없다는 것을 의미한다.
- 특히 성별, 보유 상품 갯수, 활성 고객의 여부를 나타내는 변수들이 종속변수에 큰 영향을 끼치는 것이 확인되었다.

| Logit Regression Results | | | | | | | | |
|--------------------------|------------|-------------------|---------|------------|------------------|----------------|---------|--|
| Dep. Variable: | Exited | No. Observations: | 132027 | Model: | Logit | Df Residuals: | 132014 | |
| Method: | MLE | Df Model: | 12 | Date: | Mon, 09 Dec 2024 | Pseudo R-squ.: | 0.2129 | |
| Time: | 17:13:01 | Log-Likelihood: | -53659. | converged: | True | LL-Null: | -68173. | |
| Covariance Type: | nonrobust | LLR p-value: | 0.000 | | | | | |
| | coef | std err | z | p> z | [0.025 | 0.975] | | |
| const | 6.0922 | 1.688 | 3.610 | 0.000 | 2.784 | 9.400 | | |
| CustomerId | -5.576e-07 | 1.07e-07 | -5.189 | 0.000 | -7.68e-07 | -3.47e-07 | | |
| Surname | -2.476e-05 | 9.98e-06 | -2.481 | 0.013 | -4.43e-05 | -5.2e-06 | | |
| CreditScore | -0.0007 | 9.56e-05 | -7.480 | 0.000 | -0.001 | -0.001 | | |
| Geography | 0.1091 | 0.010 | 11.438 | 0.000 | 0.090 | 0.128 | | |
| Gender | -0.6730 | 0.015 | -43.733 | 0.000 | -0.703 | -0.643 | | |
| Age | 0.0953 | 0.001 | 109.509 | 0.000 | 0.094 | 0.097 | | |
| Tenure | -0.0161 | 0.003 | -5.886 | 0.000 | -0.021 | -0.011 | | |
| Balance | 2.291e-06 | 1.29e-07 | 17.694 | 0.000 | 2.04e-06 | 2.54e-06 | | |
| NumOfProducts | -0.8263 | 0.015 | -54.329 | 0.000 | -0.856 | -0.797 | | |
| HasCrCard | -0.1506 | 0.018 | -8.565 | 0.000 | -0.185 | -0.116 | | |
| IsActiveMember | -1.2883 | 0.017 | -77.940 | 0.000 | -1.321 | -1.256 | | |
| EstimatedSalary | 9.833e-07 | 1.53e-07 | 6.418 | 0.000 | 6.83e-07 | 1.28e-06 | | |

전처리

- 현재 사용중인 데이터를 만드는데 쓰인 원본 데이터가 존재한다.



- 원본 데이터를 기존 학습 데이터와 합쳐 학습 샘플을 늘리고, 이후 전처리까지 같이 진행.

- 연속형 변수들 대부분이 정규분포를 따르지 않고 한 쪽에 몰려있는 데이터 불균형 현상을 보임.
- 또한, 연속형 변수들끼리 값의 스케일(단위)이 달라 조정이 필요함.



- **MinMaxScaler** 방식 ($((각 요소 - 최솟값) / (최댓값 - 최솟값))$)을 통해 데이터를 스케일링 해 스케일 차이를 줄이고 기존 자료의 왜곡을 최소화하면서도 학습 속도를 높이도록 한다.

- 연속형 변수와 범주형 변수가 혼재되어 사용중임.
- 또한, 기존 자료에 원본 데이터를 더했기 때문에 ID, 이름 등등이 중복되어 나타나는 경우가 있음.



- 고객의 개인 정보와 관련된 피쳐들 기준으로 그룹화하여 신용점수, 예상 급여 등 연속 변수들을 최솟값, 최대값을 비롯한 통계량으로 나타내어 **그룹에 대한 통계량으로 나타낸다.**

파생변수 생성

- 장년층 여부를 나타내는 파생변수 생성 → • 60세 이상이 넘으면 장년층으로 나타냄 (1 = yes)
 - 카드도 가지고 있고, 활성 고객인 충성 고객 여부 → • 카드 변수와 활성 고객 변수를 곱해서 나타냄 (1 = yes)
 - 거래 기간당 보유 상품 개수를 나타내는 파생변수 생성 → • 거래 기간 / 보유 상품 갯수로 계산하여 파생변수 생성
 - 나이대를 범주화 시키는 파생변수 생성 → • 나이를 20으로 나눈 후 카테고리화 하여 파생변수 생성
- 이후 범주화, 그룹화 되지 않은 기존 연속형 변수는 제거한다.
• 이렇게 완성된 최종 학습 데이터 셋의 크기는 (165034, 352)

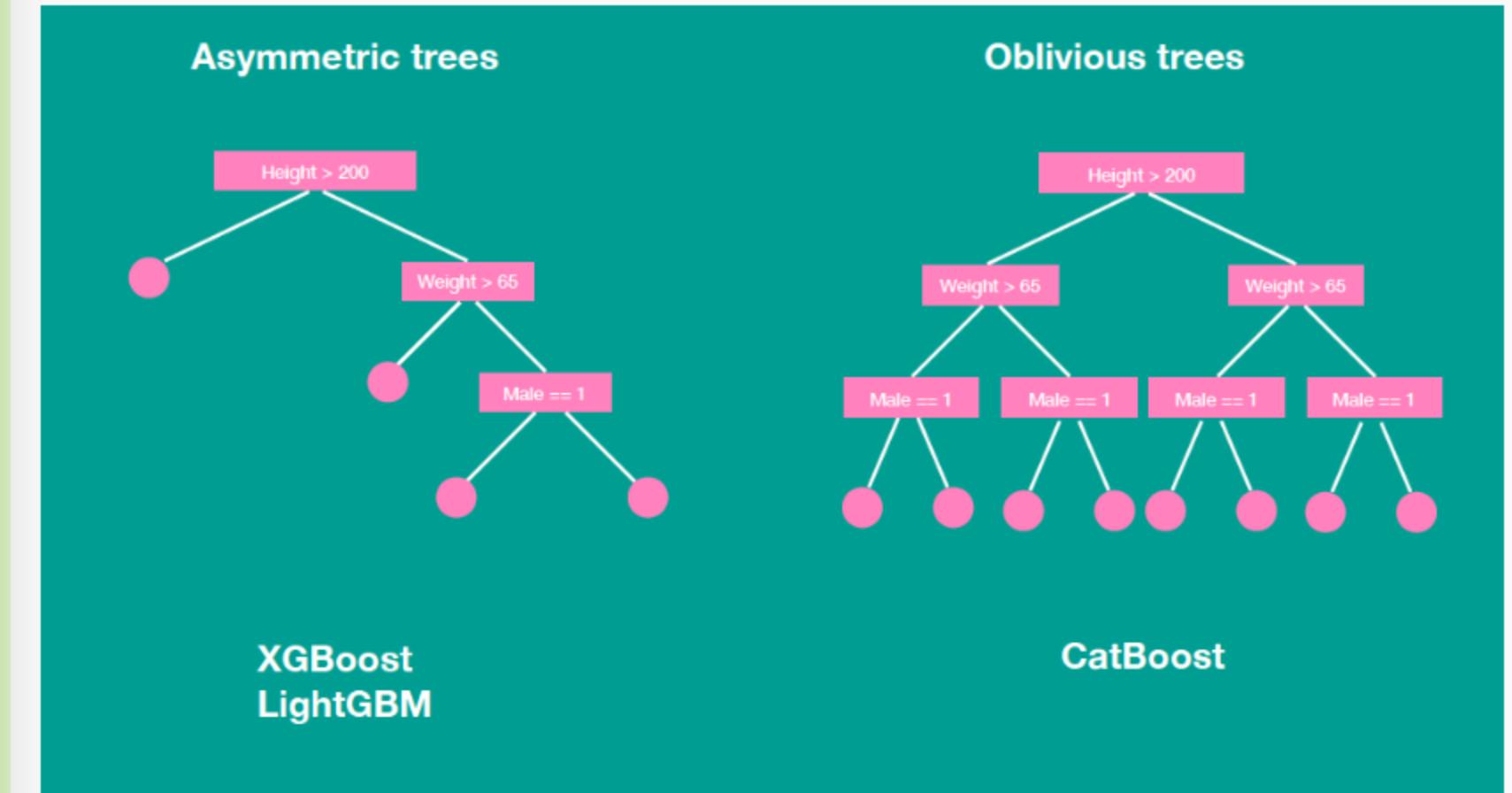
머신러닝 기법

CatBoost



CatBoost

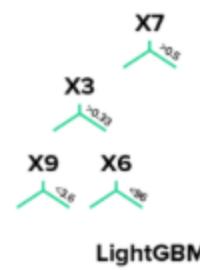
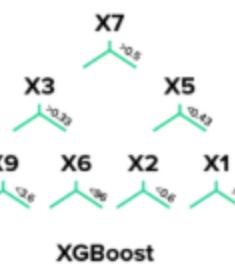
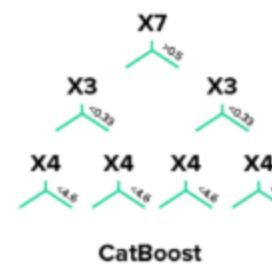
- CatBoost는 Categorical Boosting의 약자로, 결정 트리 모델의 그레디언트 부스팅(GBM)기반의 알고리즘이다.
- 이름에서 알 수 있듯이, 특히 범주형 데이터를 처리하는 데 뛰어난 성능을 보여준다.
- 범주형 데이터를 처리하는데 뛰어난 만큼, 모델 자체적으로 범주형 데이터 전처리를 진행하므로 따로 범주형 자료의 전처리를 진행할 필요가 없다. 또한 그러면서도 과적합을 방지하는 데에도 뛰어나다.
- GPU를 사용하여 연산을 더욱 빠르게 할 수 있다.
- 하이퍼 파라미터 튜닝을 하지 않아도 뛰어난 성능을 보여준다.



CatBoost의 특징

Level-wise Tree

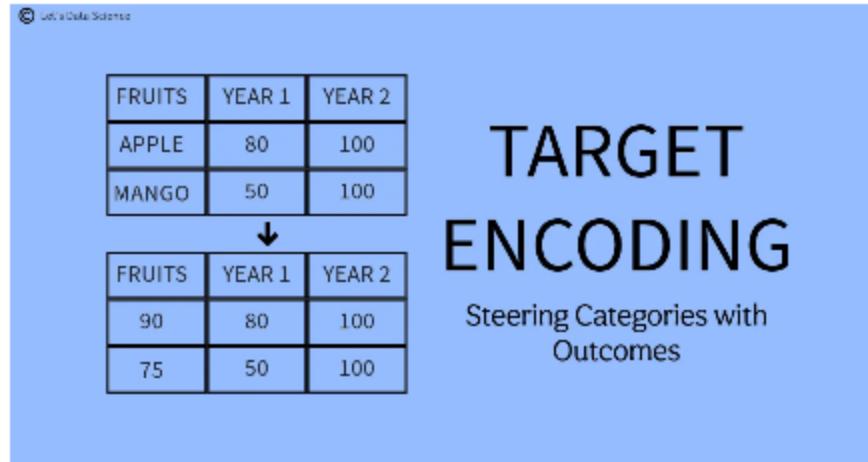
Tree growth examples:



CatBoost는 XGBoost처럼 BFS 방식 즉, level-wise 방식으로 트리의 깊이보다는 너비를 우선으로 트리를 형성하나 Feature를 모두 동일하게 대칭적인 트리 구조로 형성한다.

이런 방식을 통해 예측 시간을 감소시킨다.

Ordered Target Encoding



문자형 자료를 인코딩 할 때 흔히 쓰이는 Mean Encoding은 예측해야 할 target 값을 통해 계산하는 Target leakage 문제를 야기할 수 있다.

CatBoost는 이러한 문제를 해결하기 위해 현재 데이터의 인코딩을 위해 이전 데이터들의 Target 값을 사용하여 과적합을 막으면서도 수치값의 다양성을 챙길 수 있다.

Optimized Parameter tuning



보통 모델을 하이퍼 파라미터 튜닝을 하는 이유는 트리의 다형성과 과적합을 막고자 하는 것인데, CatBoost는 이를 모두 내부 알고리즘으로 처리 한다.

이러한 이유로 CatBoost는 이미 기본적으로 하이퍼 파라미터 최적화가 잘 되어 있으며, 설사 튜닝을 한다 해도 그 튜닝에 민감하게 반응하지 않는다.

AutoGluon



AutoGluon

- AutoGluon은 AWS에서 만든 AutoML 프레임워크이다.
- AutoGluon은 'Quick Prototyping', 즉 빠르게 가장 적절한 모델을 산출 해 내는데 특화되어 있다.
- 고전적인 행렬을 다루는 Tabular모델링부터, 시계열 모델링과 이미지와 텍스트 등을 기반으로 하는 멀티모달 모델링까지 지원한다.
- 행렬 데이터를 다루는 AutoGluon-Tabular 알고리즘은 다양한 데 이터 유형, 관계 및 배포를 처리하여 분류, 회귀, 순위 문제에 사용할 수 있다.

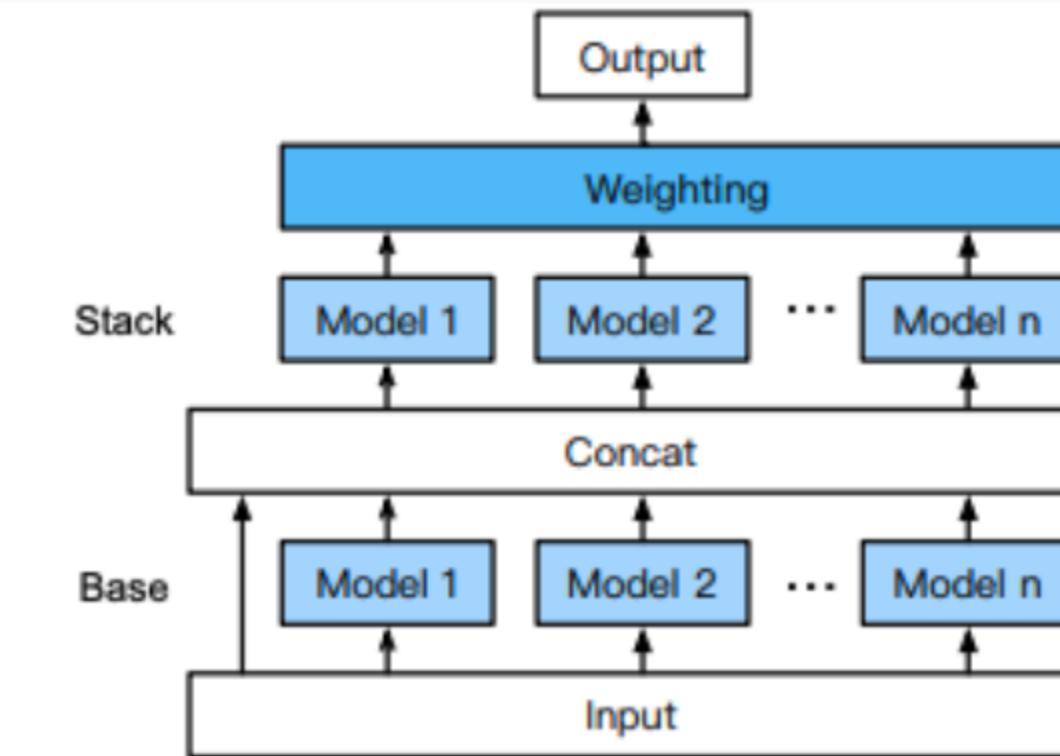
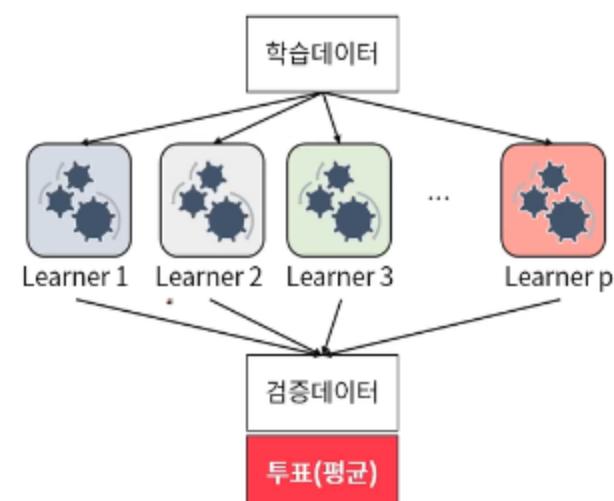


Figure 2. AutoGluon's multi-layer stacking strategy, shown here using two stacking layers and n types of base learners.

AutoGluon의 특징

자동화된 모델 선택과 앙상블 학습



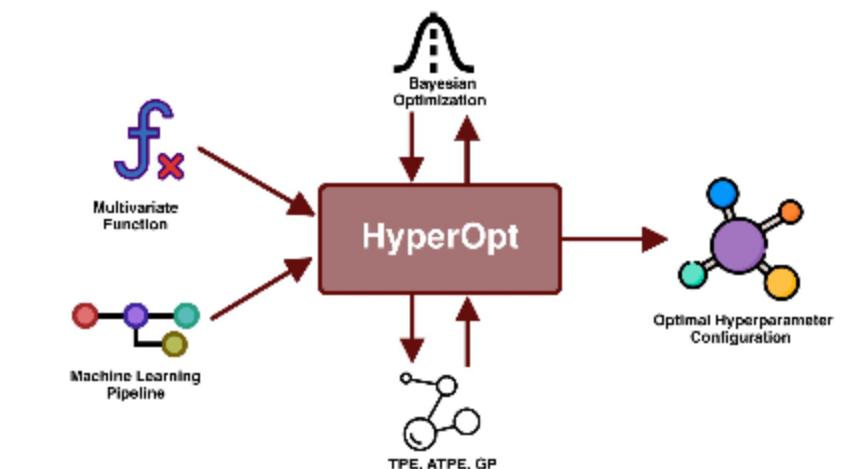
AutoGluon은 다양한 알고리즘을 기반으로 최적의 모델을 자동으로 선택하고, 앙상블 및 스택 앙상블 방식을 활용하여 높은 예측 성능을 제공한다.

다양한 데이터 유형 지원



표 형식(tabular), 텍스트와 이미지 등 다양한 데이터 유형을 처리하는 멀티모달(Multimodal), 시계열 데이터 등 다양한 데이터 유형들을 다룰 수 있으며, 데이터 유형에 따라 최적화된 모델을 적용한다.

쉬운 사용성과 자동 하이퍼파라미터 튜닝



사용자가 최소한의 코딩만으로 작업할 수 있도록 설계되었으며, 실제로 전처리도 진행하지 않고 단 세 줄로 머신 러닝을 할 수 있다. 또한 하이퍼파라미터 최적화를 자동으로 수행해 시간과 리소스를 절약한다.

왜 이 모델들인가?



AutoGluon

- 빠르게 어떤 모델이 데이터에 적합한지 알 수 있다.
- 앙상블 모델 제작을 통해 최적의 모델을 산출한다.
- 번거롭고 오래 걸리는 하이퍼 파라미터 튜닝 등도 자동으로 해 준다.



CatBoost

- AutoGluon에서 앙상블 외 가장 성능이 좋은 모델이 CatBoost였으며, 앙상블 모델에서도 가장 비중이 컸다.
- 범주형 데이터가 많고 중요한 이번 데이터에 유리함.
- 하이퍼 파라미터 튜닝에 매달리지 않아도 된다.

조정 및 결과

하이퍼 파라미터 조정



- 조정하지 않음
- `presets='best_quality'`
 - 모델 학습 시 **최고의 성능을 우선으로 함.**
 - 즉, 여러 모델들을 모두 검증하며, 각 모델에 대한 하이퍼 파라미터 조정 및 양상블과 스태킹 모델 생성 과정을 더욱 강화 한다.

하이퍼 파라미터 조정



CatBoost • CatBoost는 하이퍼 파라미터 조정을 해도 크게 성능이 변하지 않는다. → 데이터 학습을 늘린다.

| Fold 수 | iterations 수 | depth | 소요 시간 | private 점수 |
|--------|--------------|---------|----------|------------|
| 5 | 6000 | Default | 1h 56min | 0.90557 |
| 7 | 6000 | Default | 1h 28min | 0.90566 |
| 8 | 8000 | Default | 2h 16min | 0.90573 |
| 8 | 9000 | Default | 2h 29min | 0.90575 |
| 10 | 8000 | 10 | 5h 31min | 0.90507 |
| 8 | 8000 | 8 | 2h 27min | 0.90558 |
| 8 | 8000 | 7 | 2h 23min | 0.90571 |
| 10 | 8000 | Default | 2h 53min | 0.90571 |
| 9 | 10000 | Default | 3h 14min | 0.90578 |

최종 모델 및 결과



CatBoost

- CatBoost는 알고리즘상 하이퍼 파라미터를 건드리지 않아도 최적이라는 것 답게, 그나마 성능과 가장 직결되는 하이퍼 파라미터인 Depth를 건드리면 건드릴수록 오히려 성능 저해가 발생했다.
- 이는 너무 많은 fold수와 겹쳐 과적합이 일어난 것으로 추측된다.
- 따라서, 학습 데이터를 늘리는 방식으로 하이퍼 파라미터를 조정했다.
- 그렇게 찾은 최적의 하이퍼 파라미터는 모두 기본, fold 수는 9, iterations 수는 10000이다.
- 이 때의 private score는 0.90578이었다.

최종 모델 및 결과



AutoGluon

| | model | score_val | eval_metric | pred_time_val | fit_time | pred_time_val_marginal | fit_time_marginal | stack_level |
|----|-------------------------|-----------|-------------|---------------|-------------|------------------------|-------------------|-------------|
| 0 | WeightedEnsemble_L3 | 0.906713 | roc_auc | 248.781471 | 2084.281033 | 0.026547 | 10.265164 | 3 |
| 1 | CatBoost_BAG_L2 | 0.906494 | roc_auc | 202.059704 | 1824.935344 | 0.972099 | 344.704226 | 2 |
| 2 | WeightedEnsemble_L2 | 0.906346 | roc_auc | 18.124900 | 1120.616754 | 0.025915 | 6.501705 | 2 |
| 3 | CatBoost_BAG_L1 | 0.905922 | roc_auc | 1.615865 | 890.617586 | 1.615865 | 890.617586 | 1 |
| 4 | LightGBMXT_BAG_L2 | 0.905211 | roc_auc | 203.698440 | 1539.723399 | 2.610835 | 59.492281 | 2 |
| 5 | LightGBMXT_BAG_L1 | 0.904981 | roc_auc | 12.227741 | 152.336238 | 12.227741 | 152.336238 | 1 |
| 6 | LightGBM_BAG_L2 | 0.902605 | roc_auc | 203.277095 | 1531.804423 | 2.189490 | 51.573305 | 2 |
| 7 | RandomForestEntr_BAG_L2 | 0.902102 | roc_auc | 224.378161 | 1606.611350 | 23.290556 | 126.380232 | 2 |
| 8 | ExtraTreesGini_BAG_L2 | 0.901573 | roc_auc | 223.728326 | 1529.209524 | 22.640721 | 48.978406 | 2 |
| 9 | RandomForestGini_BAG_L2 | 0.900946 | roc_auc | 224.492269 | 1602.931410 | 23.404664 | 122.700292 | 2 |
| 10 | LightGBM_BAG_L1 | 0.899211 | roc_auc | 4.255380 | 71.161226 | 4.255380 | 71.161226 | 1 |
| 11 | ExtraTreesGini_BAG_L1 | 0.894360 | roc_auc | 19.506263 | 48.050297 | 19.506263 | 48.050297 | 1 |
| 12 | ExtraTreesEntr_BAG_L1 | 0.894191 | roc_auc | 19.972792 | 53.344084 | 19.972792 | 53.344084 | 1 |
| 13 | RandomForestEntr_BAG_L1 | 0.891152 | roc_auc | 20.232647 | 109.229315 | 20.232647 | 109.229315 | 1 |
| 14 | RandomForestGini_BAG_L1 | 0.890013 | roc_auc | 23.041610 | 105.732556 | 23.041610 | 105.732556 | 1 |
| 15 | XGBoost_BAG_L1 | 0.888967 | roc_auc | 2.698642 | 47.545084 | 2.698642 | 47.545084 | 1 |
| 16 | KNeighborsUnif_BAG_L1 | 0.539474 | roc_auc | 49.134801 | 1.287982 | 49.134801 | 1.287982 | 1 |
| 17 | KNeighborsDist_BAG_L1 | 0.534620 | roc_auc | 48.401864 | 0.926751 | 48.401864 | 0.926751 | 1 |

- 최고의 모델은 AutoGluon 자체 양상블 모델인 'WeightedEnsemble_L3' 모델.
- 그 뒤로는 단일모델로서 CatBoost가 가장 뛰어난 성능을 보여주고 있다.
- 양상블을 하지 않고 최고 모델인 'WeightedEnsemble_L3'의 predict값만을 제출했을 때는 0.90556의 점수가 산출되었다.

- 모델의 BAG는 부트스트랩 양상블이 사용되었음을 의미하며, L1,L2,L3는 스태킹의 어느 레벨에서 학습되었는가를 의미한다.

최종 모델 및 결과

AutoGluon의 Best Model인 **WeightedEnsemble_L3**, 하이퍼 파라미터를 조정하지 않은 **CatBoost (fold = 9, estimators = 10000)** 의 **단순 앙상블 모델** 사용



| Submission and Description | Private Score ⓘ | Public Score ⓘ | Selected |
|--|---|----------------|--------------------------|
|  engineered_data + AutoGluon - Version 12 Complete (after deadline) · 2m ago |  0.90599 | 0.90238 | <input type="checkbox"/> |

시행착오

XGBoost

- XGBoost는 GBM에 기반한 알고리즘으로, CatBoost와 동일한 level-wise 트리 방식을 사용 한다.
- 여러 내장 평가 지표를 지원하며, 필요 시 직접 평가 함수를 만들 수도 있다.
- 사용자가 직접 하이퍼 파라미터를 최적화 시킬 수 있지만, 항목이 많으며 실험에 시간이 많이 소요된다.



LightGBM

- LightGBM도 XGBoost, CatBoost처럼 GBM에 기반한 알고리즘이다. CatBoost, XGBoost와는 달리 level-wise가 아닌 leaf-wise방식 트리 분할을 쓴다.
- leaf-wise방식 덕분에 XGBoost, CatBoost보다 더 빠른 학습 속도를 제공한다.
- GPU를 지원하여 대규모 데이터셋 처리에도 유용함.

- XGBoost, LightGBM방식도 시도해 보았으나 범주형 자료에 특화된 데이터 특성 때문인지 CatBoost만큼 성능이 좋게 나오지 않았다.
- XGBoost의 경우 private score 기준 **0.89**, LightGBM은 **0.903** 정도의 성능을 보여줬다.

시행착오



- 두 모델 모두 하이퍼 파라미터에 민감하게 반응하는 만큼,
최적의 하이퍼 파라미터를 찾아낸다면 CatBoost보다 좋은
성능을 낼 가능성 있음.
- 그러나 시간 및 자원 관계상, 최적의 파라미터를 찾기 위한 실험
을 여러 번 진행 할 수 없었음.
- 따라서, 최종 모델은 적은 실험으로도 좋은 성능을 보여주는
CatBoost를 사용함.

참고 문헌

참고 문헌

- **AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data**
<https://arxiv.org/pdf/2003.06505>
- **CatBoost**
<https://velog.io/@tjddls321/CatBoost>
- **Parameter tuning**
<https://catboost.ai/docs/en/concepts/parameter-tuning>
- **나는 모델 고민할 시간에 Autogluon을 써**
<https://dacon.io/competitions/official/236075/codeshare/7764>
- **1st Place Solution - CatBoost메인**
<https://www.kaggle.com/competitions/playground-series-s4e1/discussion/472502>
- **2nd place solution - AutoGluon메인**
<https://www.kaggle.com/competitions/playground-series-s4e1/discussion/472496>

참고 코드

- Post Deadline *Top Score* 0.9055 (Late Sub) - Aravind Pillai
<https://www.kaggle.com/code/aspillai/post-deadline-top-score-0-9055-late-sub/notebook>
- Bank_churn | EDA | Catboost+LGBM+XGboost
<https://www.kaggle.com/code/getanmolgupta01/bank-churn-eda-catboost-lgbm-xgboost#-Encoding->
- 11th Place Solution - Single Model CV (CAT)
<https://www.kaggle.com/code/aspillai/11th-place-solution-single-model-cv-cat>
- EDA | Feature engineering | Ensemble model
<https://www.kaggle.com/code/joaquinemsarg/eda-feature-engineering-ensemble-model>

**THANK YOU
FOR WATCHING!**