Team Project

Kaggle Competition

[머신러닝 2분반 5조] 황인성, 오서영, 이지영

CONTENTS

00 참여도 평가

AI빅데이터융합경영학과
👑 황인성

팀장
데이터 전처리
데이터 시각화
모델링
코드 정리
평가

AI빅데이터융합경영학과
오서영

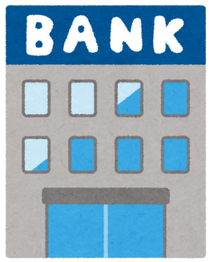
데이터 시각화
모델링
PPT 작업
평가
발표

AI빅데이터융합경영학과
이지영

데이터 탐색
모델링
PPT 작업
평가

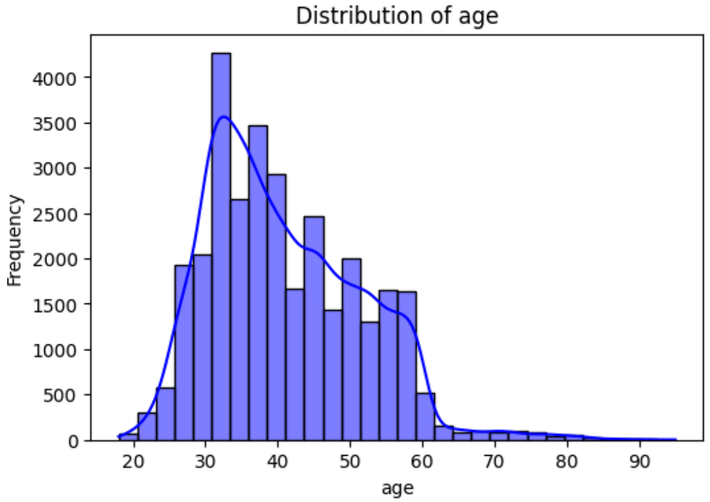
01 컴피티션 목적

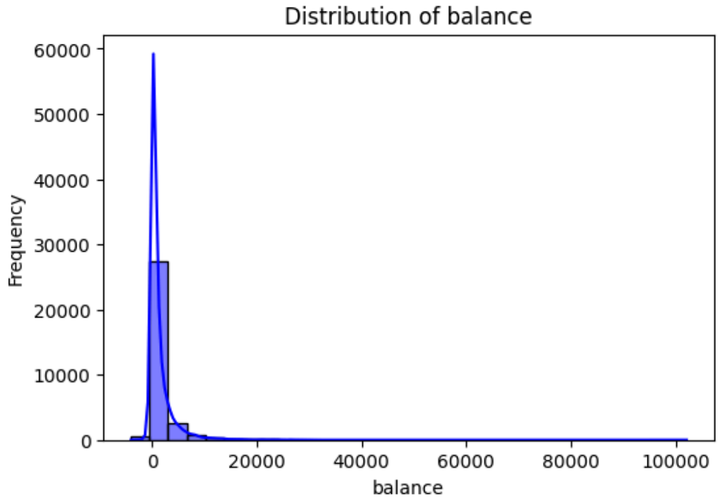
Bank marketing 참여 여부 (정기 예금 가입 여부)를 예측

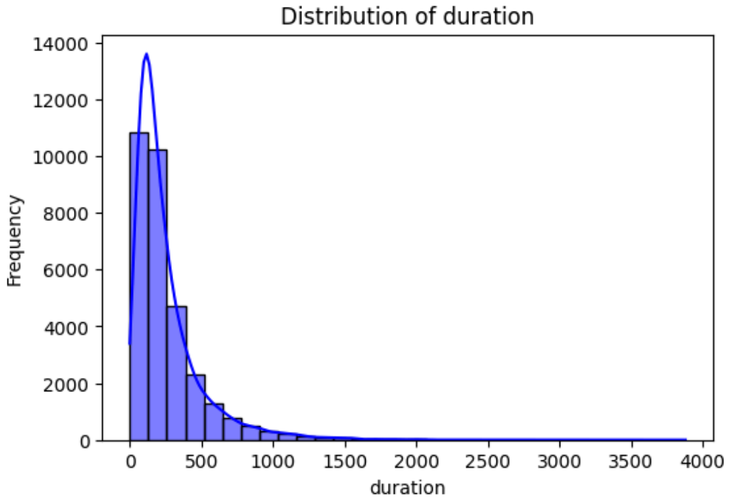


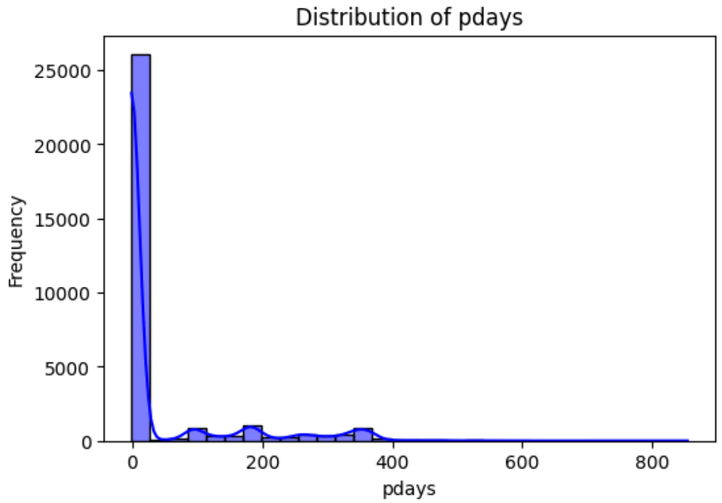
고객별로 정기 예금 가입 가능성을 예측하고 특정 고객을 타겟팅하여
마케팅 비용을 절감하고 효과를 극대화

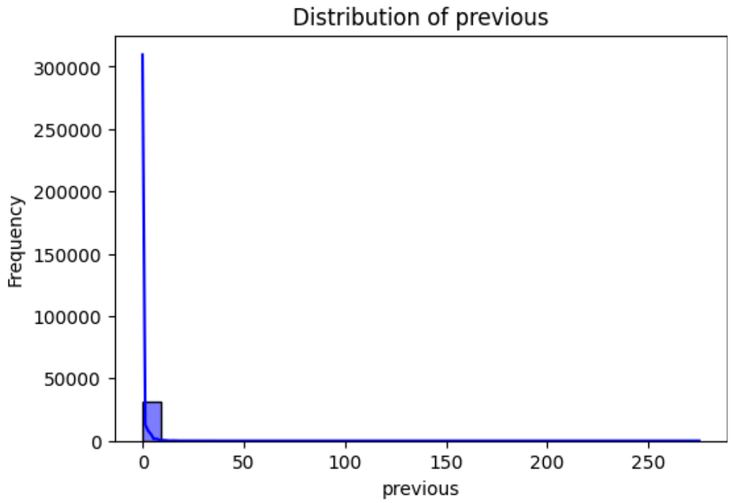
02 데이터 탐색 - 수치형

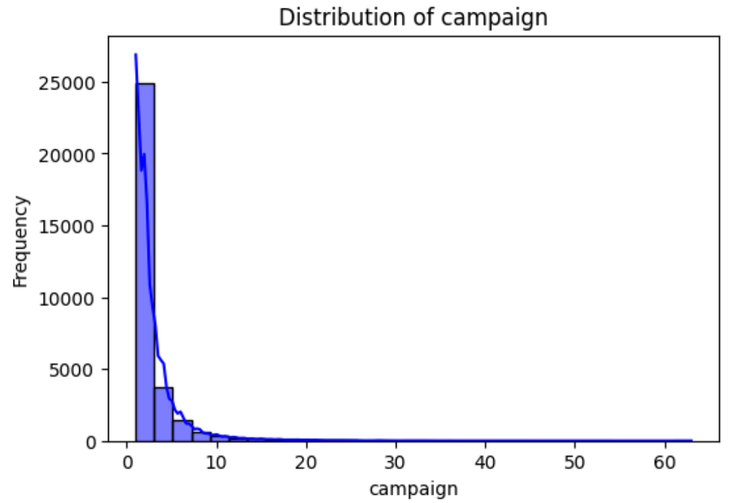

Distribution of age
Frequency
age


Distribution of balance
Frequency
balance

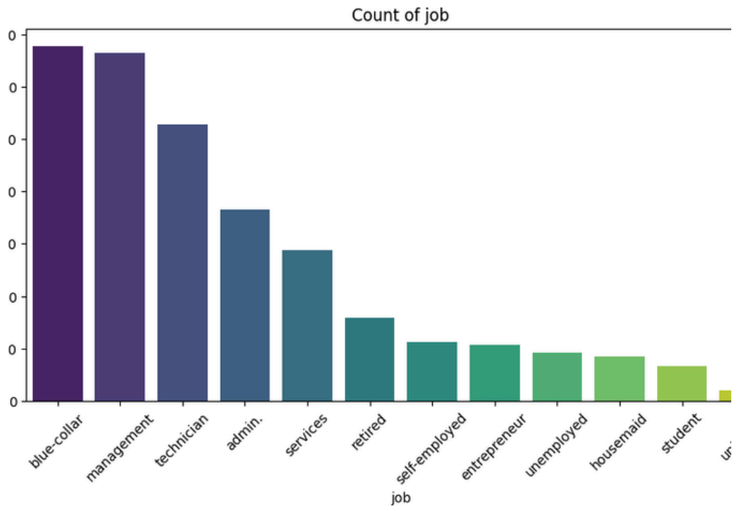

Distribution of duration
Frequency
duration

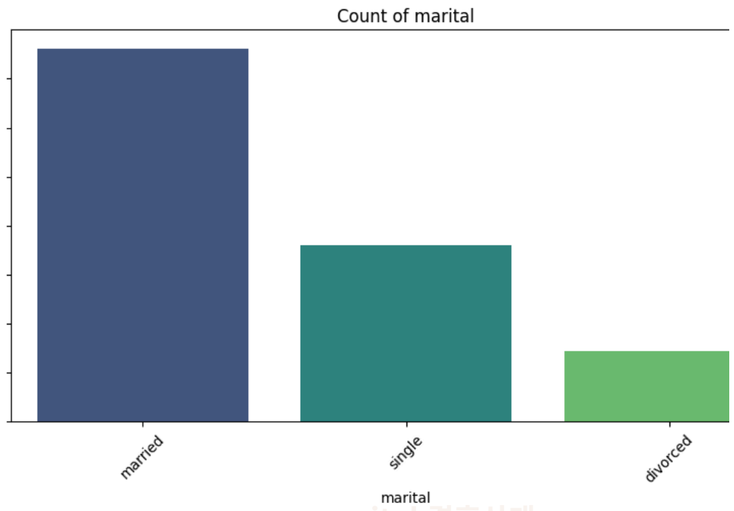

Distribution of pdays
Frequency
pdays


Distribution of previous
Frequency
previous

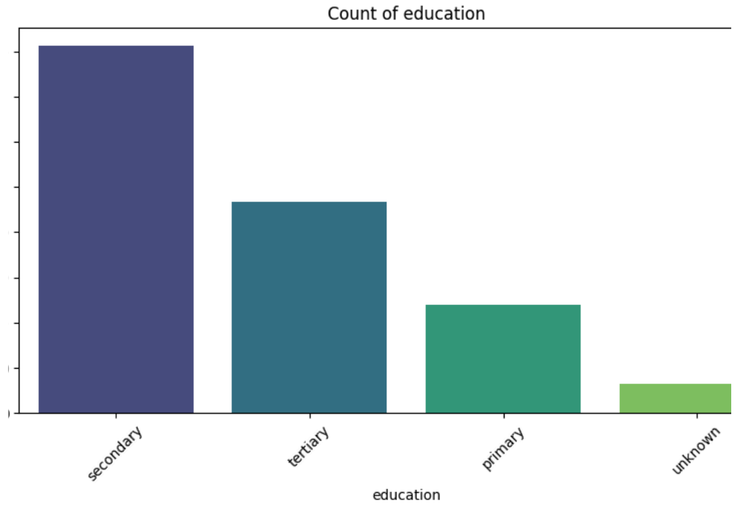

Distribution of campaign
Frequency
campaign

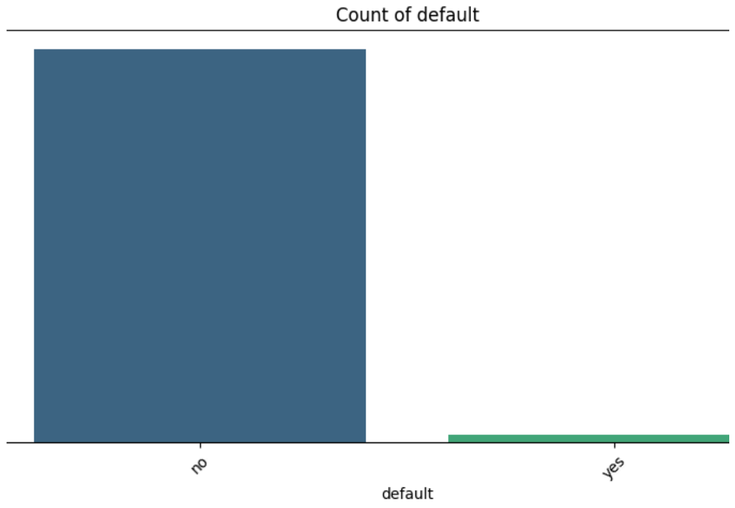
02 데이터 탐색 - 범주형



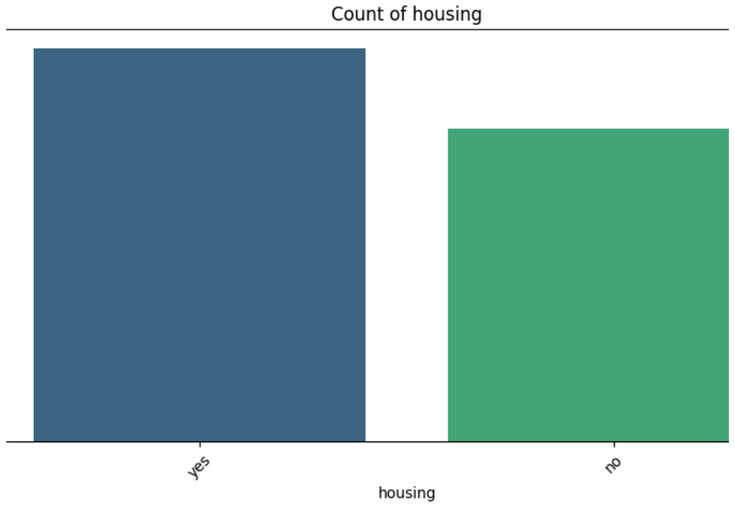


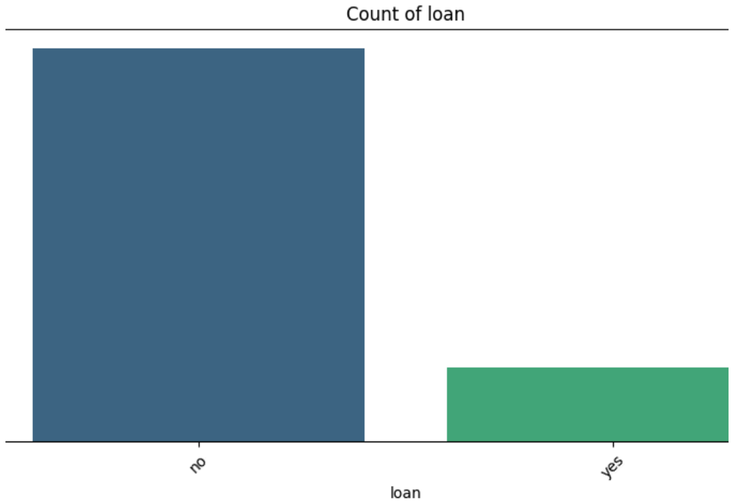
marital 결혼상태

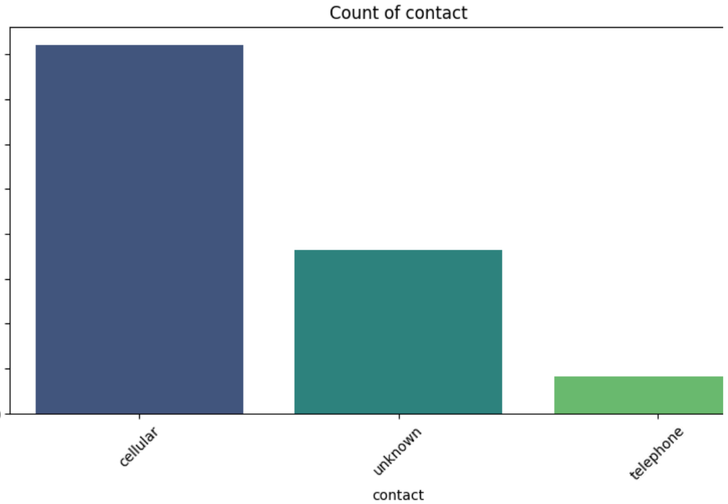




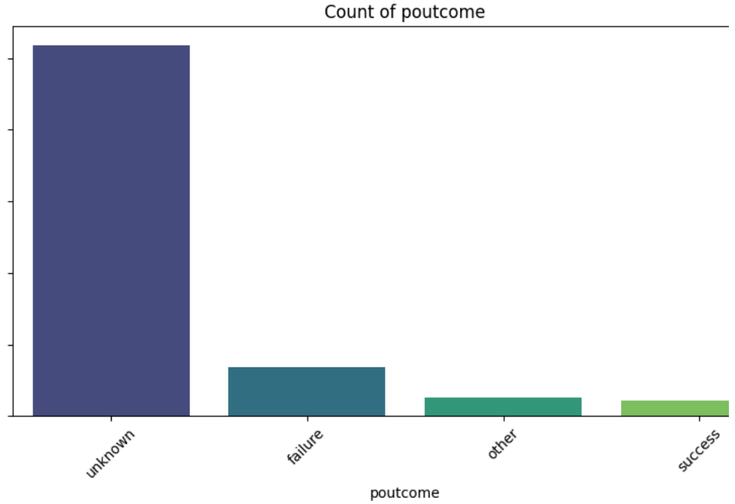
02 데이터 탐색 - 범주형

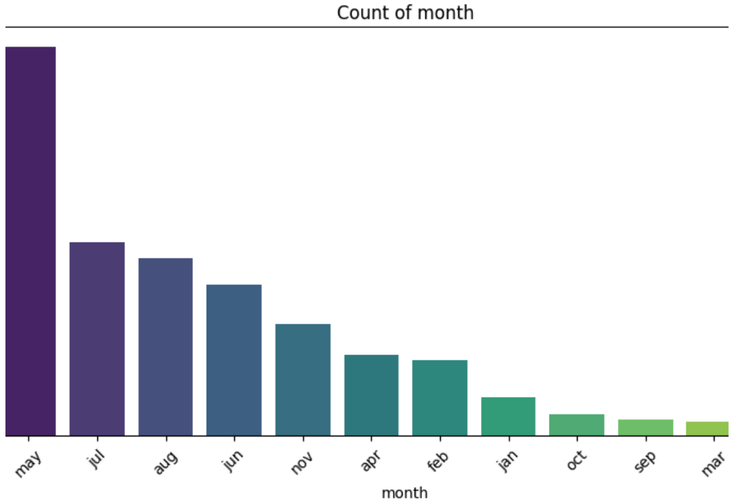






loan 개인대출 여부





03 Data Prepocessing - 피처 추가

balance 정도만 언급하고 넘어가기

balance_log

'balance' 컬럼에 로그 변환 적용

balance 값이 극단적으로 왜곡된 분포를 가지고 있기 때문에, 로그 변환을 통해 데이터 분포를 정규화하고 극단값의 영향을 줄이기 위해 추가

is_first_contact

첫 번째 연락 여부

pdays == -1은 이전 캠페인과 접촉한 적이 없음을 의미하므로 연락 여부를 명시적으로 나타내는 변수를 추가

is_success_previous

이전 캠페인 성공 여부

과거 성공 여부가 현재 캠페인 성과에 중요한 영향을 줄 수 있기 때문에 poutcome을 이진 변수로 변환하여 추가

balance_to_loan_ratio

잔고 대비 대출 비율

고객의 금융 상태를 보다 명확히 파악하기 위해 잔고의 크기를 대출에 따라 조정한 비율을 추가

03 Data Prepocessing - 피처 추가

high_balance_flag 잔고가 높은지 여부 플래그 (balance > 5000)

잔고가 높은지 여부가 예측에 영향을 줄 수 있으므로 5000 기준으로 잔고가 높은지 나타내는 이진 변수를 추가

is_telephone_contact 전화기로 연락했는지 여부

소수집단인 전화로 연락한 고객의 반응이 다른 연락 방식과 다를 수 있으므로 연락 방식의 효과를 반영하기 위해 추가

contact_duration_per_call 캠페인당 평균 통화 시간

연락 횟수와 통화 시간 간의 관계를 모델이 학습할 수 있도록 하기 위해 캠페인당 평균 통화 시간을 계산하여 추가

is_contact_in_peak_months 성수기(5월, 8월)에 연락했는지 여부

성수기(5월, 8월)에 연락했는지가 캠페인 성과에 영향을 미칠 가능성을 반영하기 위해 추가

03 Data Prepocessing - 피처 추가

quarter 분기 데이터

월별 데이터를 분기(1~4분기)로 나누어 분기별로 캠페인의 효과를 구분하기 위해 추가

age_group 연령대 그룹화

나이를 연령대별로 그룹화하여 연령대별 고객의 반응을 학습하도록 하기 위해 추가

campaign_per_duration 캠페인 횟수 대비 통화 시간 비율

연락 횟수와 통화 시간 간의 관계를 나타내어 연락의 효율성을 반영하기 위해 추가

is_young 나이가 30 이하인지 여부

나이가 30 이하인 고객을 구분하여 젊은 고객의 반응을 학습하도록 하기 위해 추가

03 Data Prepocessing

- balance 로그변환시 발생 무한값 처리

무한값 → 결측값

why?

로그변환 시 음수 or 0 계산 -> 무한값 발생

머신러닝 모델은 무한값 직접 처리 불가 -> 오류 발생

=> 결측값 처리 후 평균 및 최빈값을 사용해 처리

03 Data Prepocessing

- 결측값 처리

수치형변수 → 평균값

why?

평균값 사용
-> 데이터 분포 왜곡 X
+ 성능 저하 최소화

TrainData

ID	0
age	0
job	0
marital	0
education	0
default	0
balance	0
housing	0
loan	0
contact	0
day	0
month	0
duration	0
campaign	0
pdays	0
previous	0
poutcome	0
label	0
balance_log	0
is_first_contact	0
is_success_previous	0
balance_to_loan_ratio	0
high_balance_flag	0
is_telephone_contact	0
contact_duration_per_call	0
is_contact_in_peak_months	0
quarter	0
age_group	0
campaign_per_duration	0
is_young	0

dtype: int64

03 Data Prepocessing

- 결측값 처리

TestData

ID	0
age	0
job	0
marital	0
education	0
default	0
balance	0
housing	0
loan	0
contact	0
day	0
month	0
duration	0
campaign	0
pdays	0
previous	0
poutcome	0
label	13564
balance_log	0
is_first_contact	0
is_success_previous	0
balance_to_loan_ratio	0
high_balance_flag	0
is_telephone_contact	0
contact_duration_per_call	0
is_contact_in_peak_months	0
quarter	0
age_group	0
campaign_per_duration	0
is_young	0

dtype: int64

범주형변수 → 최빈값

why?

범주형 데이터는 평균값 X
=> 최빈값 사용

-> 데이터의 분포 최대한 유지

03 Data Prepocessing

- 최종 피처

TrainData

ID	age	job	marital	education	default	balance	housing	loan	contact	...	is_success_previous	balance_to_loan_ratio	high_balance_flag
train00001	34	blue-collar	married	primary	no	358	yes	no	unknown	...	0	358.0	0
train00002	33	blue-collar	married	secondary	no	-53	yes	no	unknown	...	0	-53.0	0
train00003	32	management	single	tertiary	no	207	yes	no	cellular	...	0	207.0	0
train00004	37	blue-collar	divorced	secondary	no	638	yes	no	cellular	...	1	638.0	0
train00005	33	housemaid	married	secondary	no	826	yes	no	cellular	...	0	826.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
train31643	60	retired	married	secondary	no	3932	yes	no	unknown	...	0	3932.0	0
train31644	44	blue-collar	married	unknown	no	15578	yes	no	unknown	...	0	15578.0	1
train31645	54	unemployed	single	secondary	no	3611	yes	no	cellular	...	0	3611.0	0
train31646	33	services	married	secondary	yes	2	no	no	cellular	...	0	2.0	0
train31647	44	technician	married	tertiary	no	851	yes						

TestData

ID	age	job	marital	education	default	balance	housing	loan	contact	...	is_success_previous	balance_to_loan_ratio	high_balance_flag
test00001	54	management	divorced	tertiary	no	6410	no	no	cellular	...	0	6410.0	1
test00002	56	unemployed	divorced	primary	no	282	no	no	cellular	...	0	282.0	0
test00003	34	management	married	secondary	no	355	no	no	cellular	...	0	355.0	0
test00004	55	management	married	tertiary	no	568	no	no	cellular	...	0	568.0	0
test00005	38	technician	married	secondary	no	6728	no	no	cellular	...	0	6728.0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
test13560	32	management	single	tertiary	no	19985	no	no	cellular	...	0	19985.0	1
test13561	31	student	single	tertiary	no	307	no	no	cellular	...	0	307.0	0
test13562	38	blue-collar	married	secondary	no	20	yes	no	unknown	...	0	20.0	0
test13563	26	management	single	tertiary	no	3	yes	no	cellular	...	0	3.0	0
test13564	33	blue-collar	married	primary	no	-132	yes	yes	cellular	...	0	-66.0	0

04 Encoding

< Label Encoding >

저희가 사용한 트리 기반 모델들은 범주형 데이터를
직접 처리할 수 있기 때문에 굳이 원핫 인코딩을 쓰지 않고
레이블 인코딩을 사용했습니다

원핫은 개별 차원으로 표현하기 때문에 게산량이
늘어날 수 있습니다.

cat boost - 범주형 데이터 처리 가능 but 내부적으로 숫자로 변환된 값 사용

-> 범주형 데이터 정수로 변환하는 Label Encoding 사용

-> 고유값 간의 관계 자동 학습

-> 효율적 범주형 데이터 처리

TestData - TrainData 학습 매핑 규칙 그대로 적용

-> 데이터 간 불일치 방지

05 Modeling

CatBoost

다양한 카테고리형 변수

데이터의 복잡성 처리

과적합 방지

Cross-validation

데이터 분포의 불균형 처리

일반화 성능 확보

05 Modeling

< 파라미터 설명 >

f1 스코어는 0과 1 중에서 1인 걸 예측하는 것이 중요하다

근데!
정밀도는 맞다고 예측 -> 실제로 맞음
재현율은 실제로 맞음 -> 맞다고 예측

'iterations': 10000

'learning_rate': 0.04

'depth': 4

'verbose': 1000

'random_seed': 42

'eval_metric': 'F1'

-> F1 스코어는 정밀도와 재현율의 조화 평균으로, 클래스 불균형 문제에서 유용

05 Modeling

< OOF -> 확률로 변환 후 임계값 변환 >

OOF(Out-of-Fold) 예측

predict_proba는 확률이 저장되기 때문에
우리가 수정할 수 있는 개꿀 장점

Cross-Validation을 사용하여 모델을 평가할 때, OOF 예측을 통해 각 fold에서 모델이 예측한 확률을 얻음.

이를 통해 모델이 새로운 데이터를 어떻게 일반화할지 알 수 있음.

확률을 임계값으로 변환

OOF 예측에서 나온 확률값을 사용하여 임계값(0.3)을 적용해 최종 클래스 결정을 함. 이렇게 함으로써 모델이

각 인스턴스에 대해 예측한 확률을 기반으로 최적의 임계값을 설정하고, 이에 따라 F1 스코어를 최적화할 수 있음.

06 Evaluation

<평가지표>

F1-score

정밀도(Precision)와 재현율(Recall)의 조화 평균으로, 불균형 데이터에서도 모델 성능을 공정하게 평가

본 컴피티션의 주요 평가지표가 F1-Score였으므로, F1-Score 최적화를 중심으로 모델링을 진행

AUC-PR

XGBoost는 F1-Score를 eval_metric으로 지원하지 않으므로 AUC-PR(Average Precision Score)를 사용하여

학습 과정에서 모델의 성능을 평가

불균형 데이터에서 Positive 클래스(가입)를 효과적으로 평가할 수 있는 지표

06 Evaluation
성능확인

CatBoost

```
Cross-Validation(5-Fold Stratified K-Fold) 결과:
- 각 Fold의 F1-Score: [0.6213821618428824, 0.6073805202661826, 0.6150091519219036, 0.6361386138613861, 0.6178073894609327]
- 평균 F1-Score: 0.6195435674706575

테스트 데이터 결과:
- F1-Score: 0.619
```

▶▶▶▶

훈련 데이터와 검증 데이터의 성능 차이를 비교하여 과적합 여부를 평가
Cross-Validation에서 평균 F1-Score는 0.619, 테스트 데이터에서도 F1-Score 0.619를 기록하여 일반화 성능을 확인

Hard voting

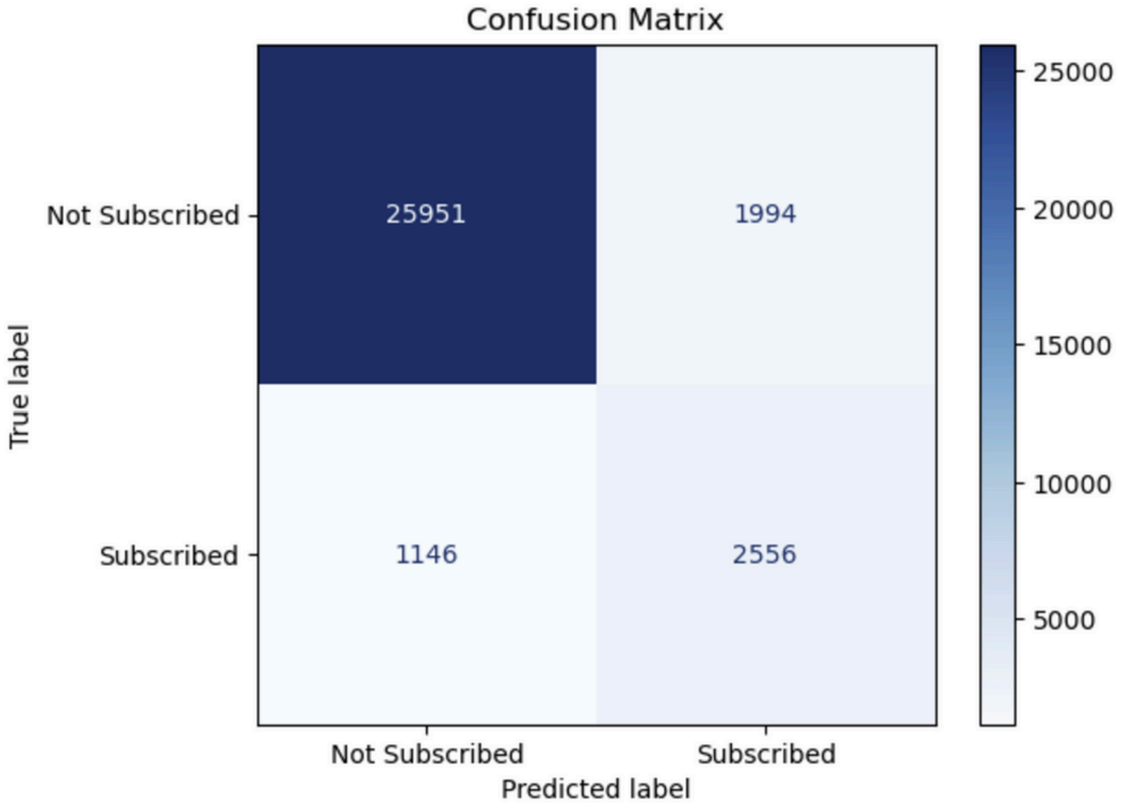
```
Cross-Validation(5-Fold Stratified K-Fold) 결과:
- 각 Fold의 F1-Score: [0.6309255079006771, 0.6126436781609195, 0.6208476517754868, 0.6424809830310123, 0.6385681293302541]
- 평균 F1-Score: 0.629

테스트 데이터 결과:
- F1-Score: 0.639
```

▶▶▶▶

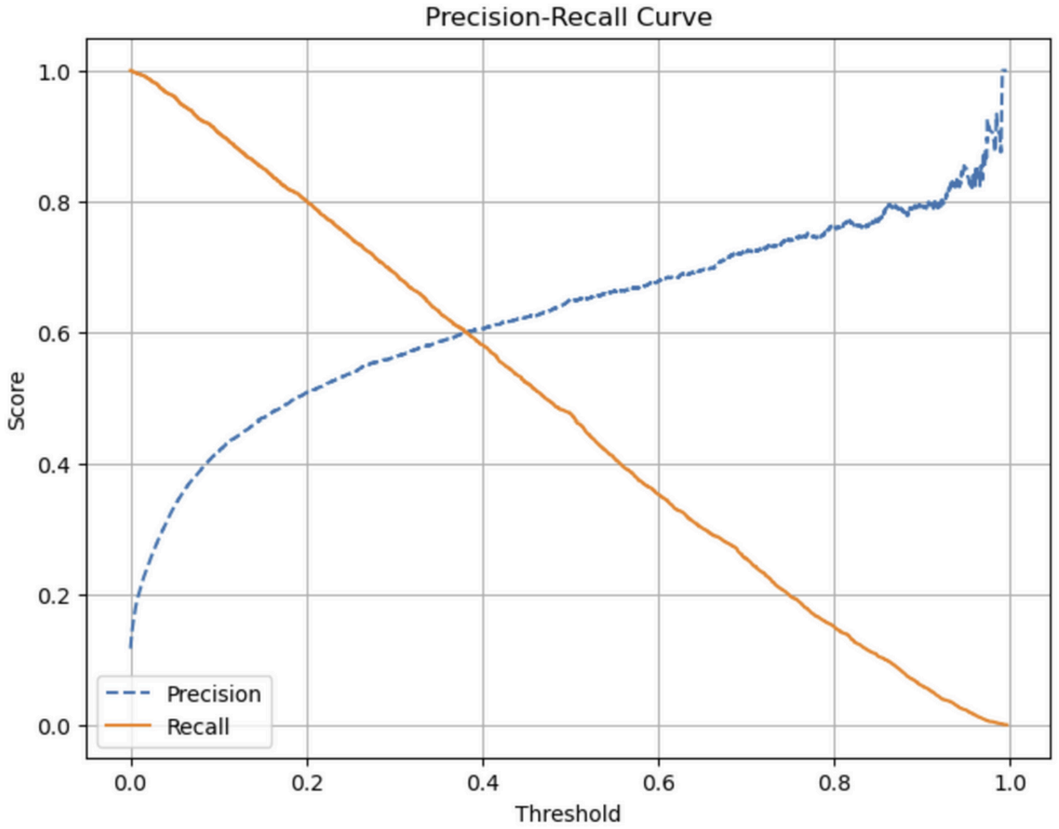
Cross-Validation에서 평균 F1-Score는 0.629, 테스트 데이터에서는 F1-Score 0.639를 기록하여 일반화 성능을 확인

06 Evaluation
-Catboost

혼동행렬
정밀도 재현율 그래프
를 통해 모델의 결과를 시각적으로 확인



Confusion Matrix



Precision-Recall Curve

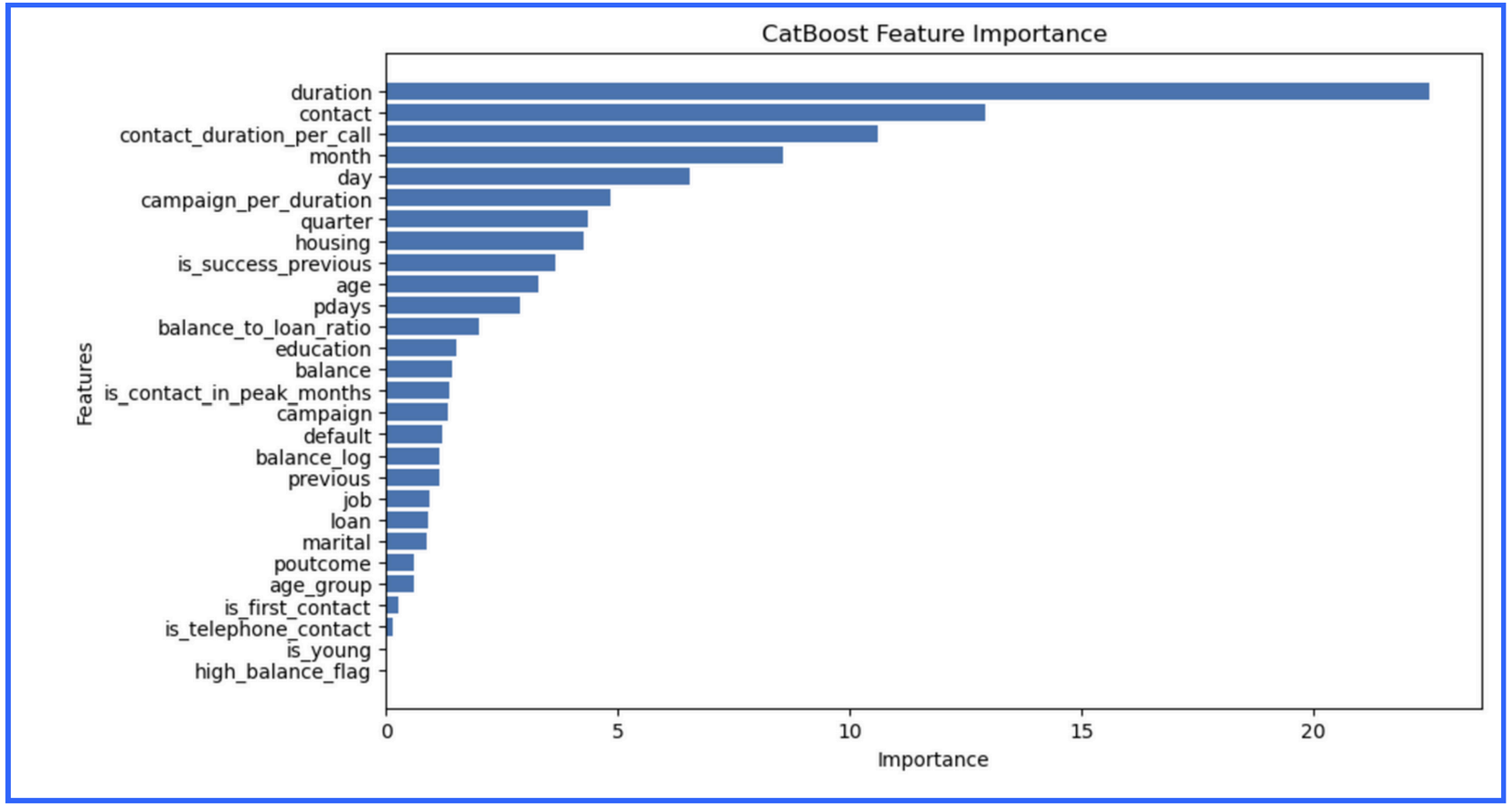
06 Evaluation

-Hard Voting

캣부스트임

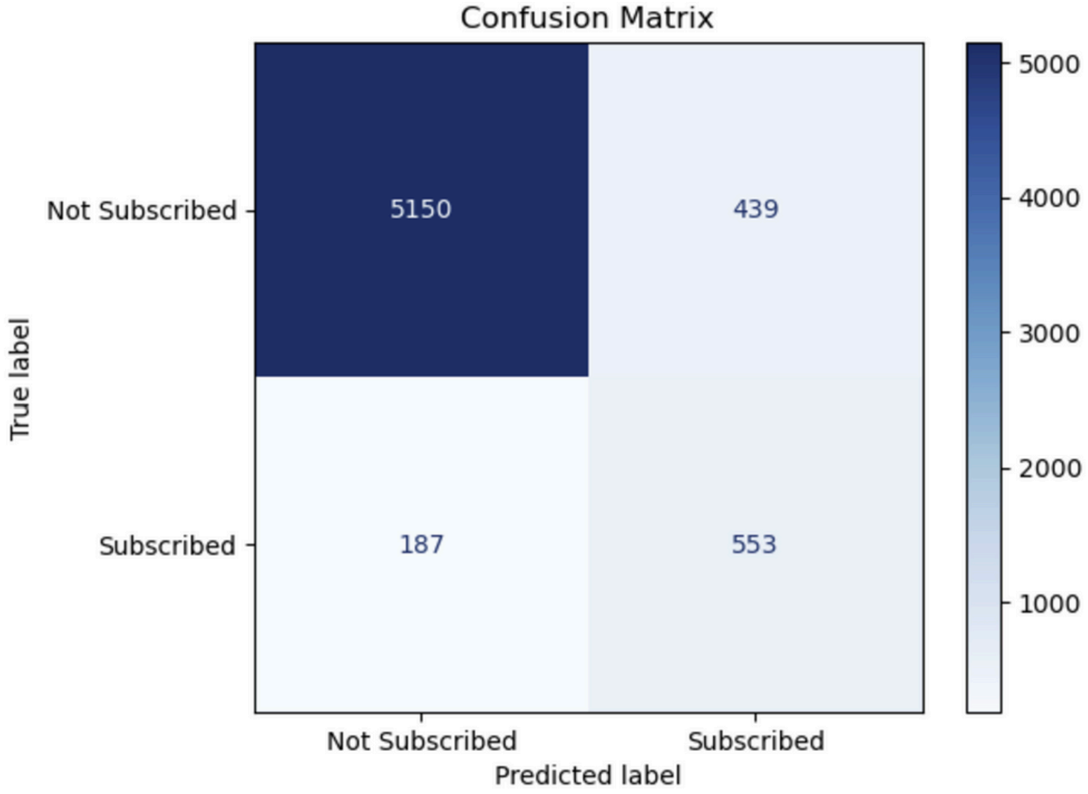
오타임 ㅎㅎ

CatBoost

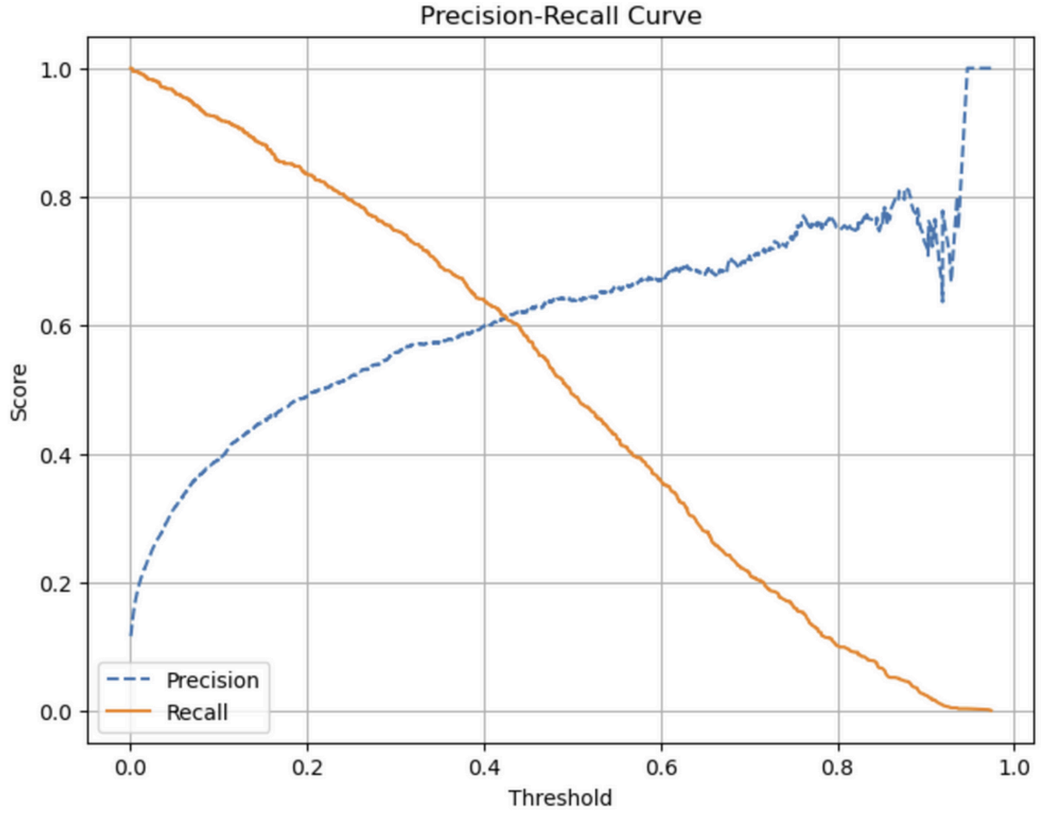


06 Evaluation
-Hard Voting

CatBoost 모델에서는 전체 데이터를 기준으로 혼동 행렬과 PR-Curve를 생성했는데, Hard Voting Ensemble에서는 실수로 마지막 K-Fold 검증 데이터만 사용해 혼동 행렬과 PR-Curve를 생성했습니다. 그래서 두 혼동 행렬의 표본 수가 다르게 나타나게 되었음



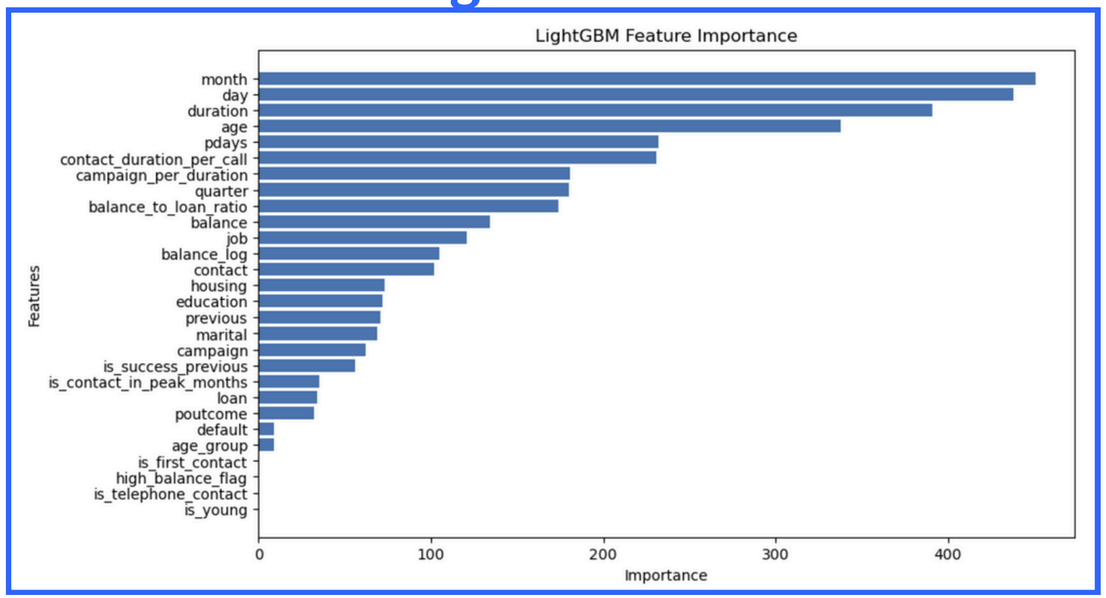
Confusion Matrix



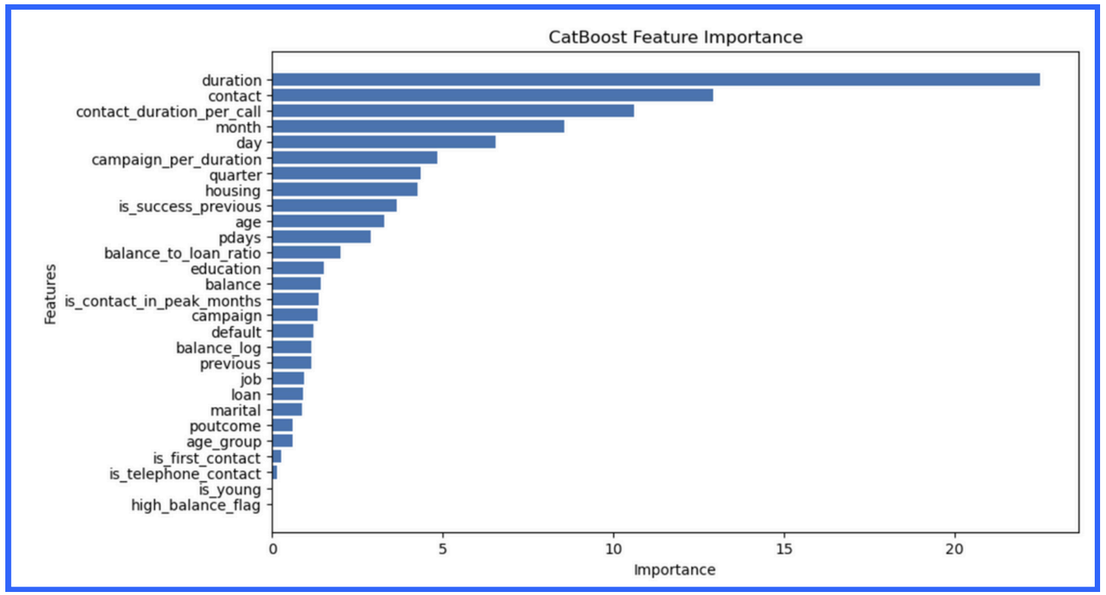
Precision-Recall Curve

06 Evaluation

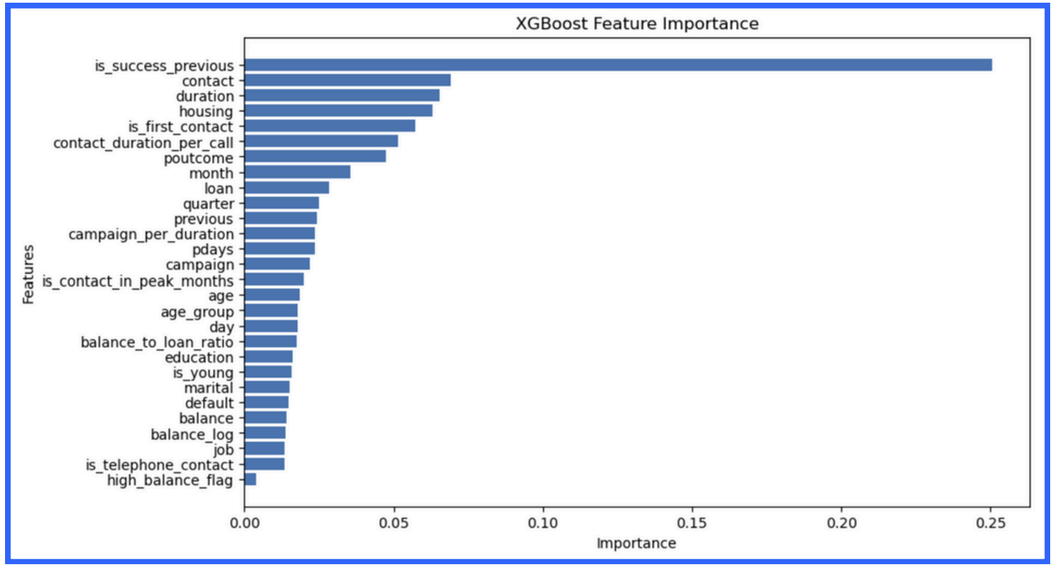
-Hard Voting



LightGBM



CatBoost



XGBoost

07 아쉬운 점

1. 임계값 0.42로 제출된 결과

Public 점수: 임계값 0.42는 Public 데이터에서 높은 점수를 기록
문제점: Public 데이터에 과적합된 결과로, Private 점수가 하락

2. 최적 임계값 0.32

최적화 과정: F1-Score를 기준으로 최적 임계값을 0.32로 계산 후 Private 데이터에서 더 높은 점수를 기록
-> 0.32는 Public 데이터에서는 점수가 낮아졌지만, Private 데이터에서 안정적인 성능을 보임

3. 교훈 및 아쉬움

Public 데이터에 과적합된 임계값(0.42)을 제출한 점이 아쉬움
F1-Score 기반 최적 임계값(0.32)을 적극적으로 활용했다면 Private 점수 향상이 가능했을 것

Submission and Description	Private Score	Public Score
oof_predictions_0.3_feature3.csv Complete · 5d ago · 피처추가한거	0.64846	0.65489
base.csv Complete · 5d ago	0.64798	0.65927
hardvoting_0.32.csv Complete · 3d ago	0.64750	0.65945

-> 제출 못 함
-> 제출 됨
-> 제출 못 함 (앙상블 최고 성능)

07 아쉬운 점

Hard voting Ensemble (Catboost, LightGBM, XGboost)

Gradient Boosting 기반의 알고리즘인 세 모델은 서로 다른 특성
(CatBoost: 범주형 데이터에 강함
LightGBM: 속도가 빠름
XGBoost: 안정적인 성능 제공)
을 가지고 있어 상호보완적임

Hard Voting의 효과
다양성 보장
리스크 분산
결과의 안정성 향상

Cross-validation (fold=5, StratifiedKFold)

데이터 불균형 문제 해결

일관된 성능 평가

Ensemble 최적화

07 아쉬운 점

19페이지로 이동 ㄱㄱㄱㄱ

CatBoost

'iterations': 10000,

'learning_rate': 0.04,

'depth': 4,

'eval_metric': 'F1',

'verbose': 100,

'random_seed': 42

LightGBM

'learning_rate': 0.074,

'num_leaves': 37,

'max_depth': 12,

'min_child_samples': 53,

'subsample': 0.86,

'colsample_bytree': 0.92,

'reg_alpha': 2.5,

'reg_lambda': 0.001

XGBoost

'n_estimators': 1000,

'max_depth': 5,

'learning_rate': 0.02,

'subsample': 0.6,

'colsample_bytree': 0.8,

'gamma': 0.93,

'min_child_weight': 1,

'scale_pos_weight': 1.47,

'objective': 'binary:logistic',

'eval_metric': 'aucpr',

'random_state': 42,

'use_label_encoder': False,

'verbosity': 1

클래스 불균형이 심했기 때문에 XGBoost의 scale_pos_weight 조절에 집중함

감사합니다.