



Lending Club Data Analysis

2조

통계데이터사이언스

강수정, 배기태, 심준선, 이강산, 이선유, 전상언, 황정현

CONTENTS

Lending Club Data Analysis

1 배경

2 변수 선정 및 데이터 전처리

3 모델링

4 결론

1. 배경

...

01

배경

Lending Club 데이터

미국의 P2P 대출 중개 사업

- 2007년~2020년까지 대부분 관련 자료를 공개
- 2020년 영업 종료

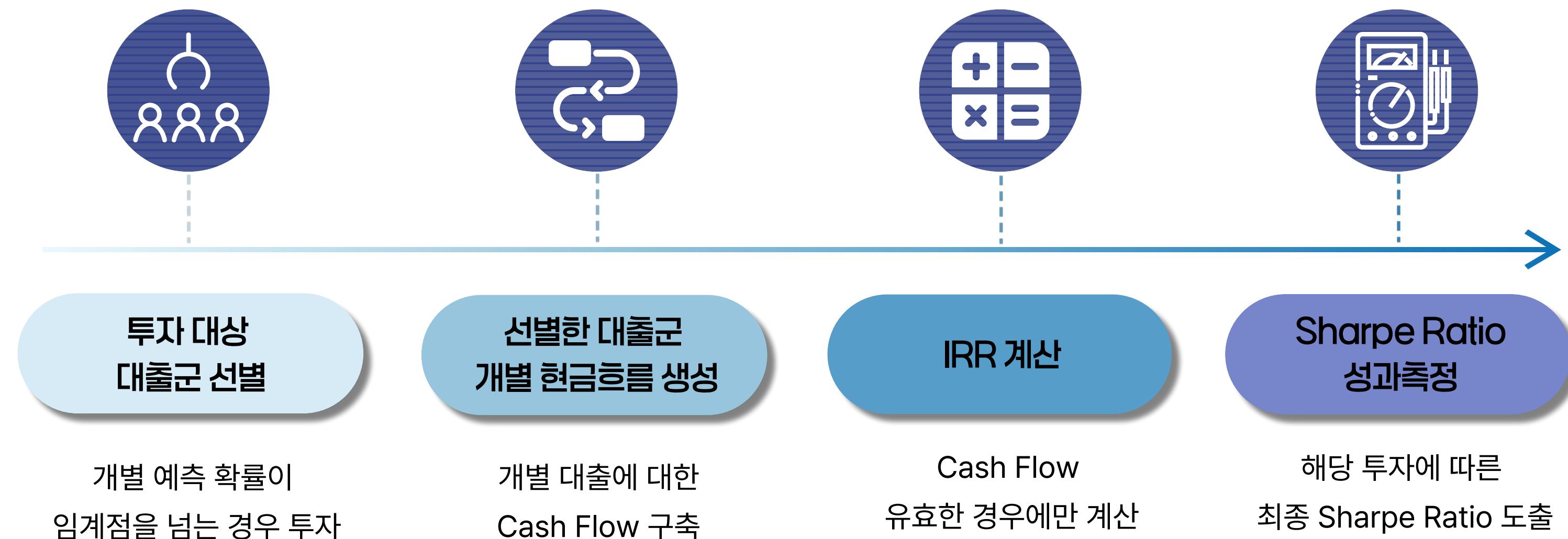


리스크 감안한 초과 수익률 (Sharpe Ratio) 극대화

01

배경

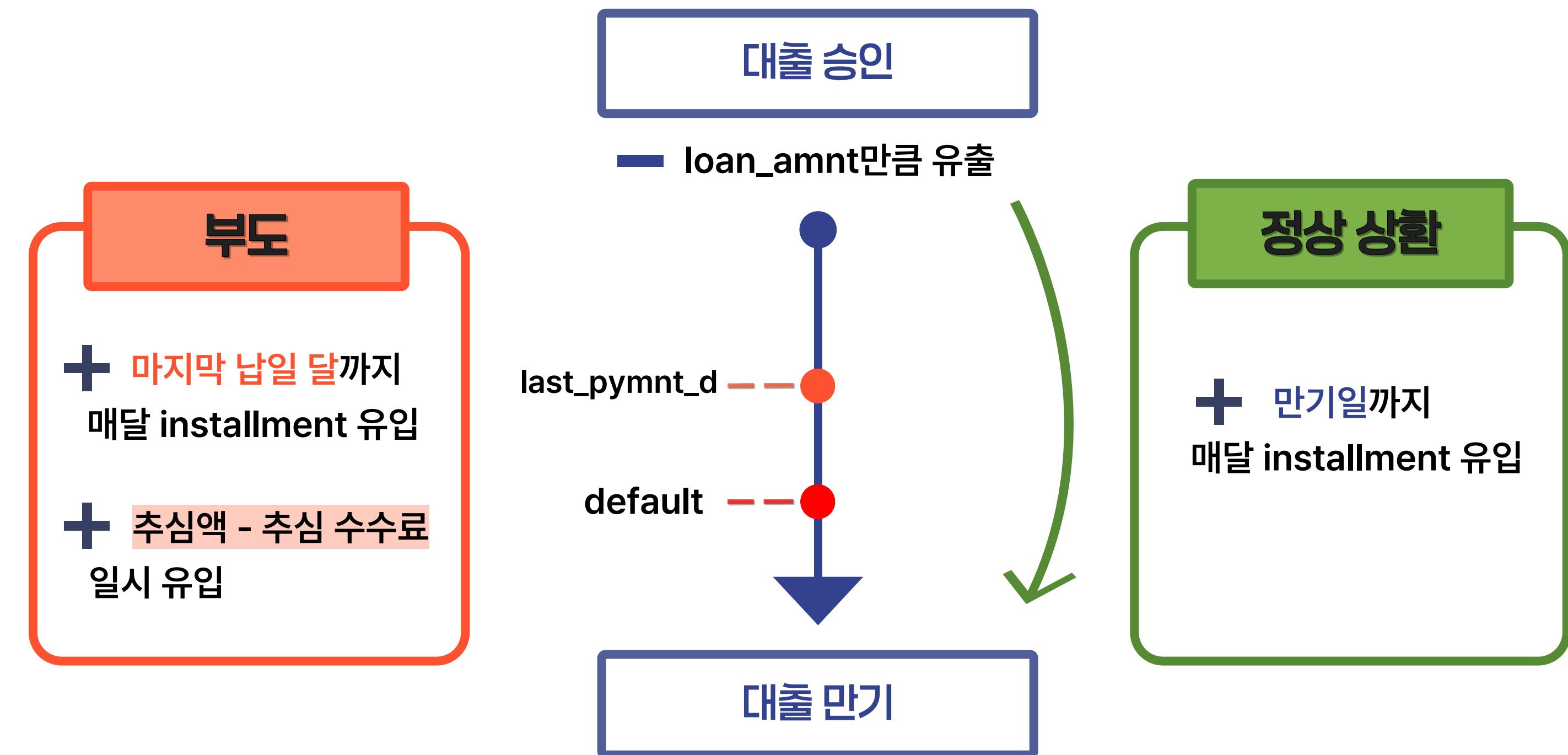
Project Flow



01

배경

대출별 Cash Flow 구축



...

01

배경

Sharpe Ratio Function

$$S_a = \frac{E[R_a - R_b]}{\sigma_a} = \frac{E[R_a - R_b]}{\sqrt{\text{var}[R_a - R_b]}}$$

```
# Sharpe 계산 함수
def calculate_sharpe(returns, risk_free_rates):
    excess = returns - risk_free_rates
    if excess.std(ddof=1) == 0:
        return -np.inf
    return excess.mean() / excess.std(ddof=1)
```

- Risk Free Rate

채권 발행일과 연동된
국채 수익률 (Fred API 활용)

2. 변수선정 및 데이터 전처리



...
02
데이터
전처리

전처리 loan_status

Current

현재 정상적인 상환 중
아직 결과 확인 X → 모델링 시 제외

Fully Paid

대출 원금, 이자 대부분 상환
성공 (비부도: 0으로 처리)

Charged Off

상환을 포기하고 손실처리
실질적 부도 (부도: 1로 처리)

Default

Charged Off와 통합
부도 (부도: 1로 처리)

02

데이터
전처리

전처리 변수 삭제

```
# 2. 삭제할 변수 리스트 (84개)
drop_cols = [ # 이전에 지정한 84개
    'application_type', 'grade', 'sub_grade', 'verification_status_joint', 'hardship_loan_status',
    'hardship_type', 'hardship_reason', 'hardship_status',
    'deferral_term', 'hardship_start_date', 'hardship_end_date', 'payment_plan_start_date',
    'hardship_length', 'hardship_dpd', 'orig_projected_additional_accrued_interest',
    'hardship_amount', 'hardship_payoff_balance_amount', 'hardship_last_payment_amount',
    'sec_app_revol_util', 'revol_bal_joint', 'sec_app_fico_range_low', 'sec_app_fico_range_high',
    'sec_app_earliest_cr_line', 'sec_app_inq_last_6mths', 'sec_app_mort_acc', 'sec_app_open_acc',
    'sec_app_open_act_il', 'sec_app_num_rev_accts', 'sec_app_chargeoff_within_12_mths',
    'sec_app_cc_late_12_mths_ex_med', 'sec_app_cc_joint', 'sec_app_cc_since_last_revoked',
    'mths_since_rcnt_cr', 'mths_since_rcnt_as', 'mths_since_derog', 'last_pymnt_d', 'inq_f',
    'total_cr_l', 'emp_title', 'num_accts_by_type', 'hardship_f', 'earliest_cr_line',
    'earliest_cr_line', 'tunded_amnt_inv', 'id', 'loan_amnt',
    'initial_list_status', 'int_rate', 'last_credit_pull_d',
    'last_pymnt_amnt', 'out_prncp', 'out_prncp_inv', 'policy_code',
    'pymnt_plan', 'total_pymnt', 'total_pymnt_inv', 'total_rec_int',
    'total_rec_late_fee', 'total_rec_prncp', 'url', 'zip_code', 'debt_settlement_flag',
    'desc', 'member_id', 'verified_status_joint', 'sec_app_mths_since_last_major_derog',
    'disbursement_method', 'debt_settlement_flag_date', 'settlement_status', 'settlement_date',
    'settlement_amount', 'settlement_percentage', 'settlement_term', 'application type'
    # 추가 삭제(LC 대출 절차상 LC 대출 심사 후 적용되는 걸로 생각할 수 있음
    'last_fico_range_high', 'last_fico_range_low', 'verification_status'
]

df.drop(columns=[col for col in drop_cols if col in df.columns], inplace=True)
```

내생변수삭제 결족치 삭제

*75%이상 **공동차입자

02

데이터
전처리

전처리 결측치 보완

10% 기준

미만

```
# 결측률 < 10%인 변수들 전처리

# emp_length: 결측치 = 0 + missing label
df['emp_length_missing'] = df['emp_length'].isnull().astype(int)
df['emp_l']

(1) 0 값 채우기 + Missing 라벨링

# percent_bc_gt_75: 결측치 = 0 + missing label
df['percent_bc_gt_75_missing'] = df['percent_bc_gt_75'].isnull().astype(int)
df['percent_bc_gt_75'] = df['percent_bc_gt_75'].fillna(0)

(2) 0 값 채우기

# 0으로 채운 변수들
fillna0_cols = df[fillna0_cols].fillna(0)

(3) Missing 라벨링

median_replacement_cols = df[median_replacement_cols].fillna(df[median_replacement_cols].median())

df[median_replacement_cols] = df[median_replacement_cols].fillna(df[median_replacement_cols].median())
```

이상

```
# 결측률 10% 이상인 변수들은 일괄적으로 결측값 0 대체+missing indicator로 처리
for col in non_dummy_numeric_cols:
    # 결측 여부 표시하는 변수 생성 (1: 결측, 0: 결측 아님)
    df[f'{col}_missing'] = df[col].isnull().astype(int)

(1) 0 값 채우기

# 결측값을 0으로 채운 변수들
df[col] = df[col].fillna(0)

(2) Missing 라벨링
```

02

데이터 전처리

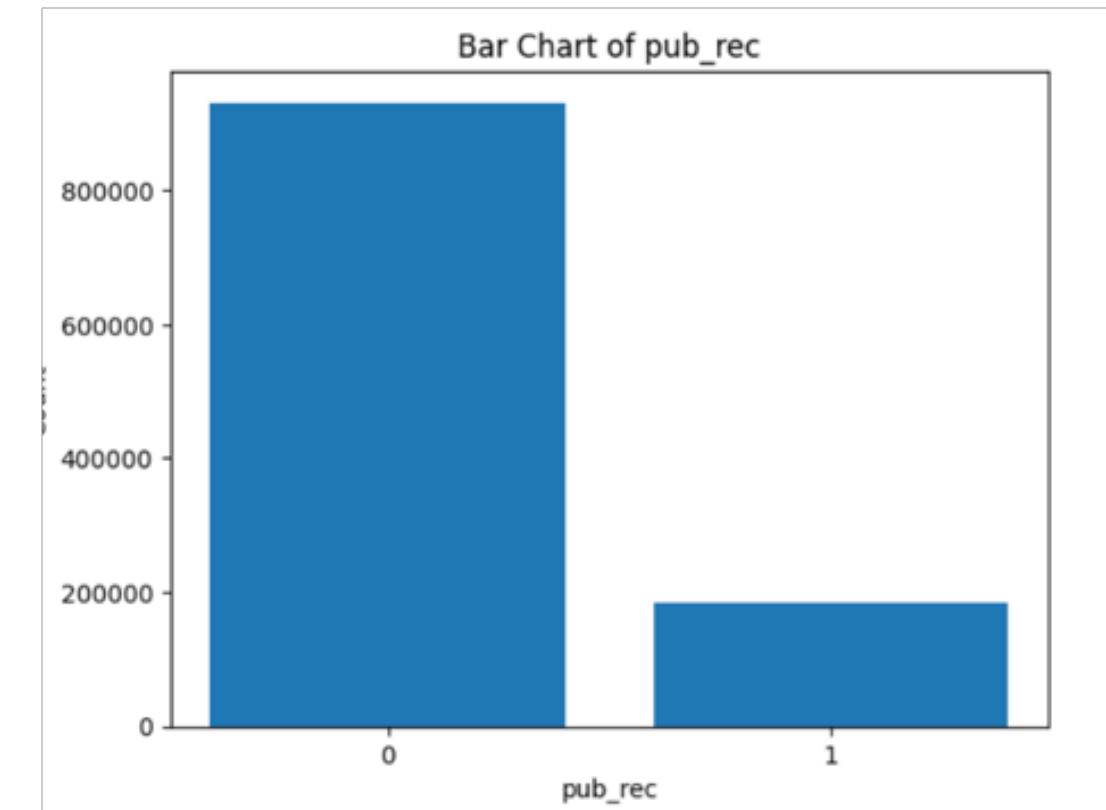
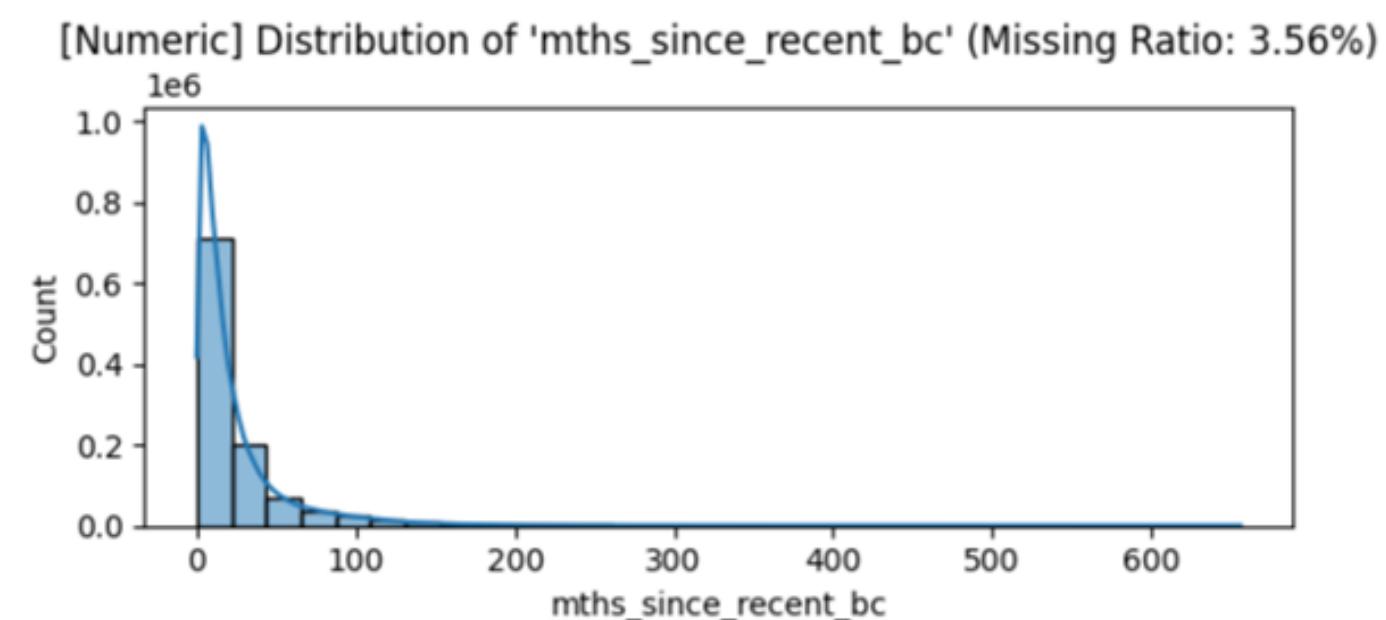
전처리 로그 및 바이너리 변환

로그 변환

바이너리 변환

이상치 조정

유/무 판단 중요



3. 모델링



...
03
모델링

공통 모델 학습 Process

RandomForest
XGBoost
LightGBM

해당 과정을 100회 반복!

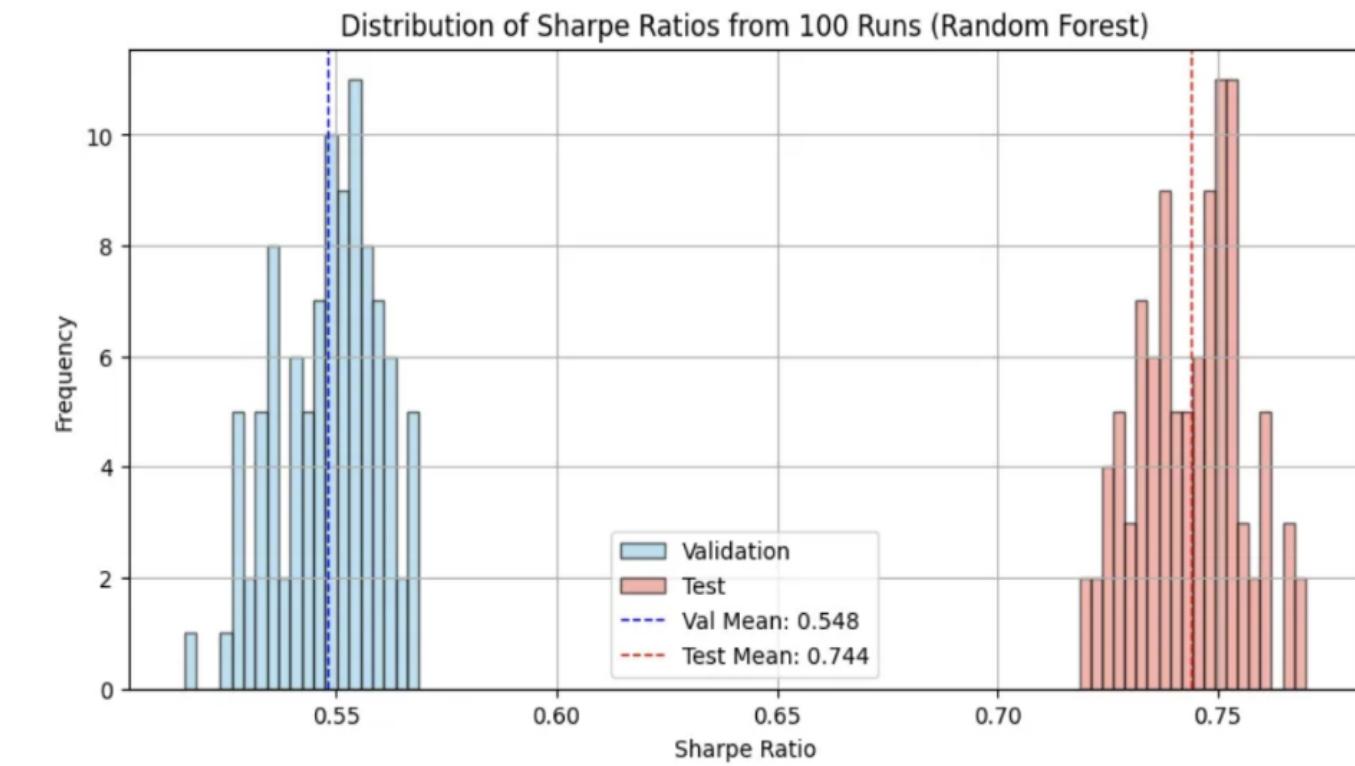
- ① 데이터 분할
- ② 최적화된 Threshold를 기준으로 대출 승인
- ③ 승인된 대출의 IRR로 조정 수익률 생성
- ④ 해당 수익률로 Sharpe Ratio 계산
- ⑤ Sharpe Ratio가 가장 높은 Threshold 선정

03 모델링

Tree Model RandomForest

```
for i in tqdm(range(100)):
    # 매 반복마다 다른 seed 사용
    X_temp, X_test, y_temp, y_test = train_test_split(
        X, y, test_size=0.2, random_state=i, stratify=y
    )
    X_train, X_val, y_train, y_val = train_test_split(
        X_temp, y_temp, test_size=0.25, random_state=i, stratify=y_temp
    )

    # 모델 학습
    model = RandomForestClassifier(**best_params, random_state=i)
    model.fit(X_train, y_train)
```



하이퍼파라미터 튜닝

'n_estimators': [50, 100, 200, 300], 'max_depth': [3, 5, 7, 10],
'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4],
'max_features': ['sqrt', 'log2']

03 모델링

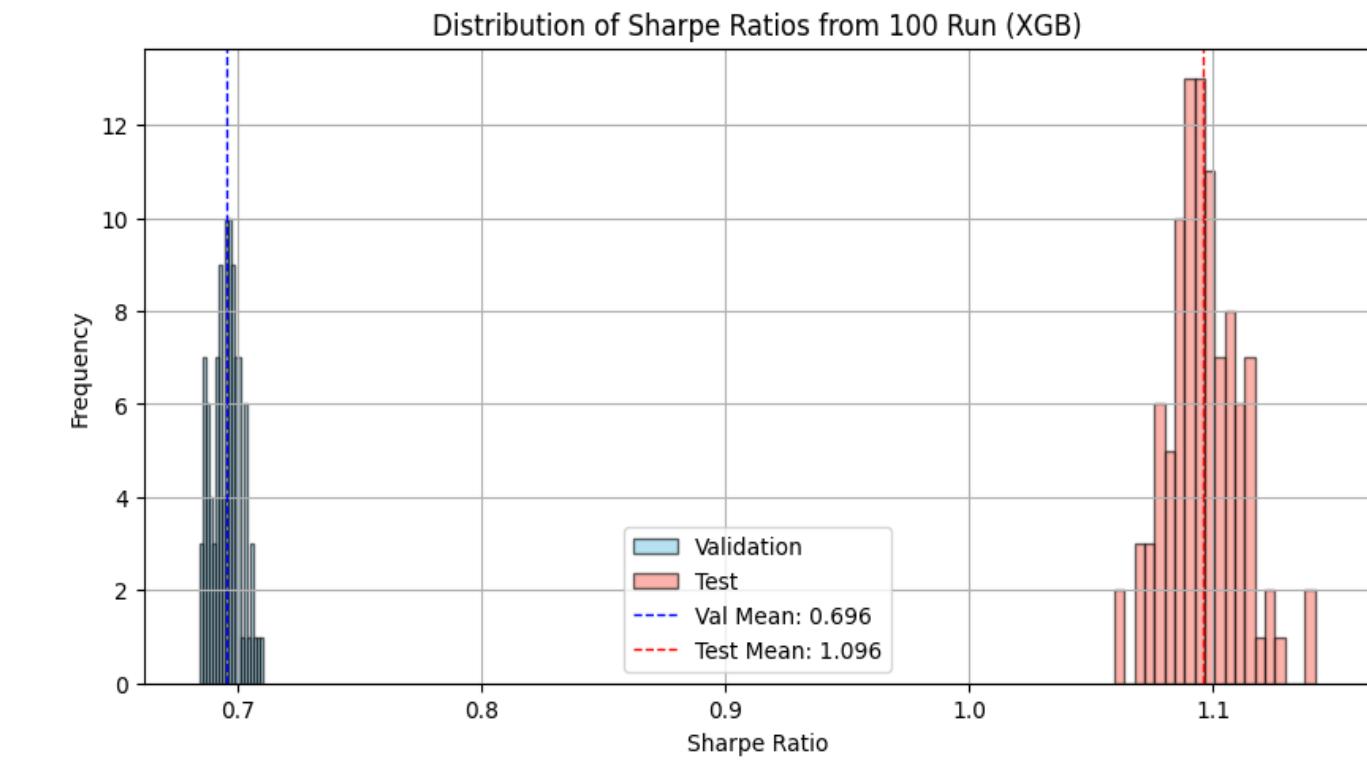
Tree Model XGBoost

```
# 해당 best 파라미터를 기반으로 100번 반복 학습 및 평가
for i in tqdm(range(100)):
    # Train-test split
    X_temp, X_test, y_temp, y_test = train_test_split(
        X, y, test_size=0.2, random_state=i, stratify=y
    )
    X_train, X_val, y_train, y_val = train_test_split(
        X_temp, y_temp, test_size=0.25, random_state=i, stratify=y_temp
    )

    # best 하이퍼파라미터로 모델 생성 및 학습
    model = XGBClassifier(**best_params, eval_metric='logloss')
    model.fit(X_train, y_train)
```

하이퍼파라미터 튜닝

'n_estimators': [100, 200, 300], 'max_depth': [3, 5, 7, 10],
'learning_rate': [0.01, 0.05, 0.1], 'subsample': [0.6, 0.8, 1.0],
'colsample_bytree': [0.6, 0.8, 1.0], 'gamma': [0, 1, 5], 'min_child_weight':
[1, 3, 5], 'reg_alpha': [0, 0.1, 1], 'reg_lambda': [1, 5, 10]



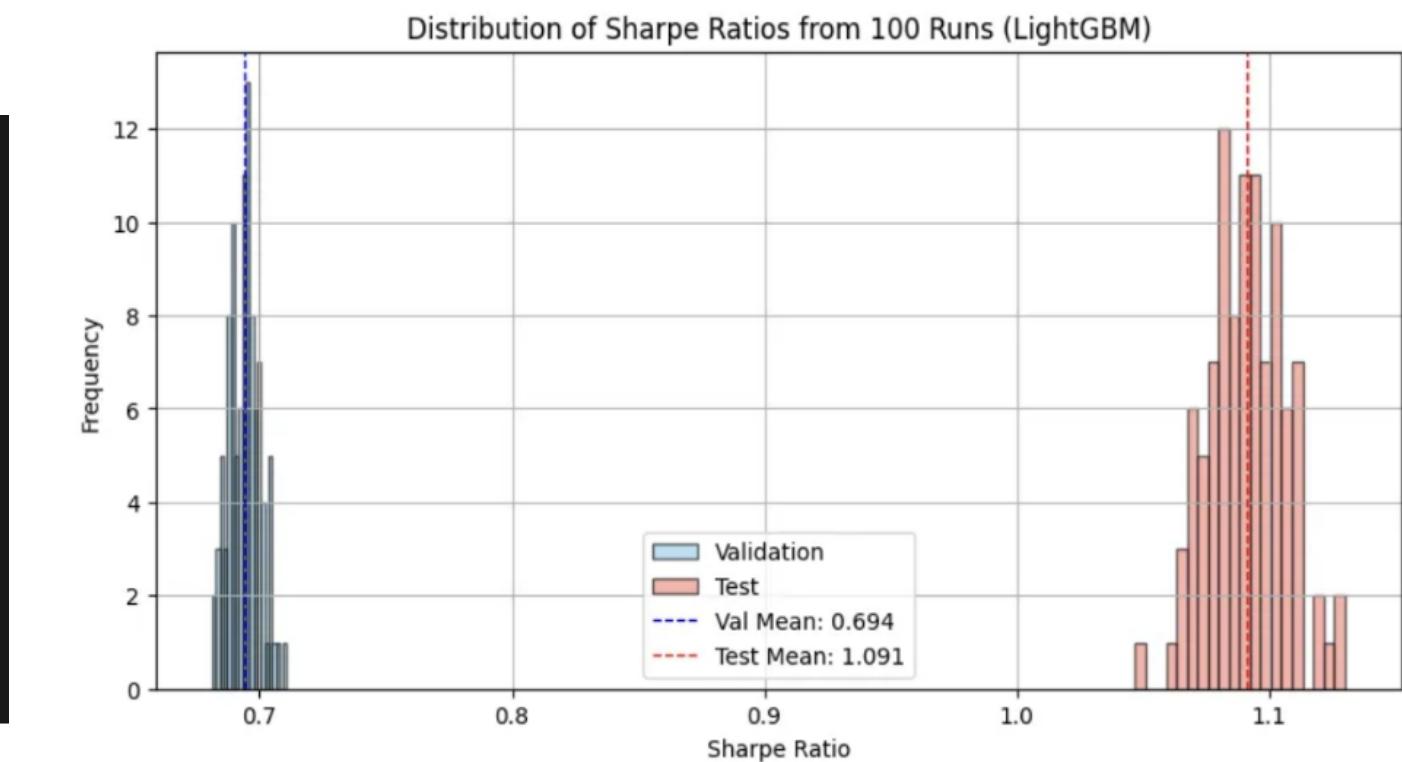
03 모델링

Tree Model **LightGBM**

```
for i in tqdm(range(100)):
    # 데이터 분할
    X_temp, X_test, y_temp, y_test = train_test_split(
        X, y, test_size=0.2, random_state=i, stratify=y
    )
    X_train, X_val, y_train, y_val = train_test_split(
        X_temp, y_temp, test_size=0.25, random_state=i, stratify=y_temp
    )

    # LightGBM 모델 학습
    model = lgb.LGBMClassifier(**best_params, random_state=i, n_jobs=-1)
    model.fit(X_train, y_train)
```

하이퍼파라미터 튜닝



'n_estimators': [100, 200, 300], 'max_depth': [3, 5, 7, 10], 'num_leaves': [15, 31, 63], 'learning_rate': [0.01, 0.05, 0.1], 'min_child_samples': [10, 20, 30], 'subsample': [0.6, 0.8, 1.0], 'colsample_bytree': [0.6, 0.8, 1.0]

4. 결론

04

결과

모델 비교

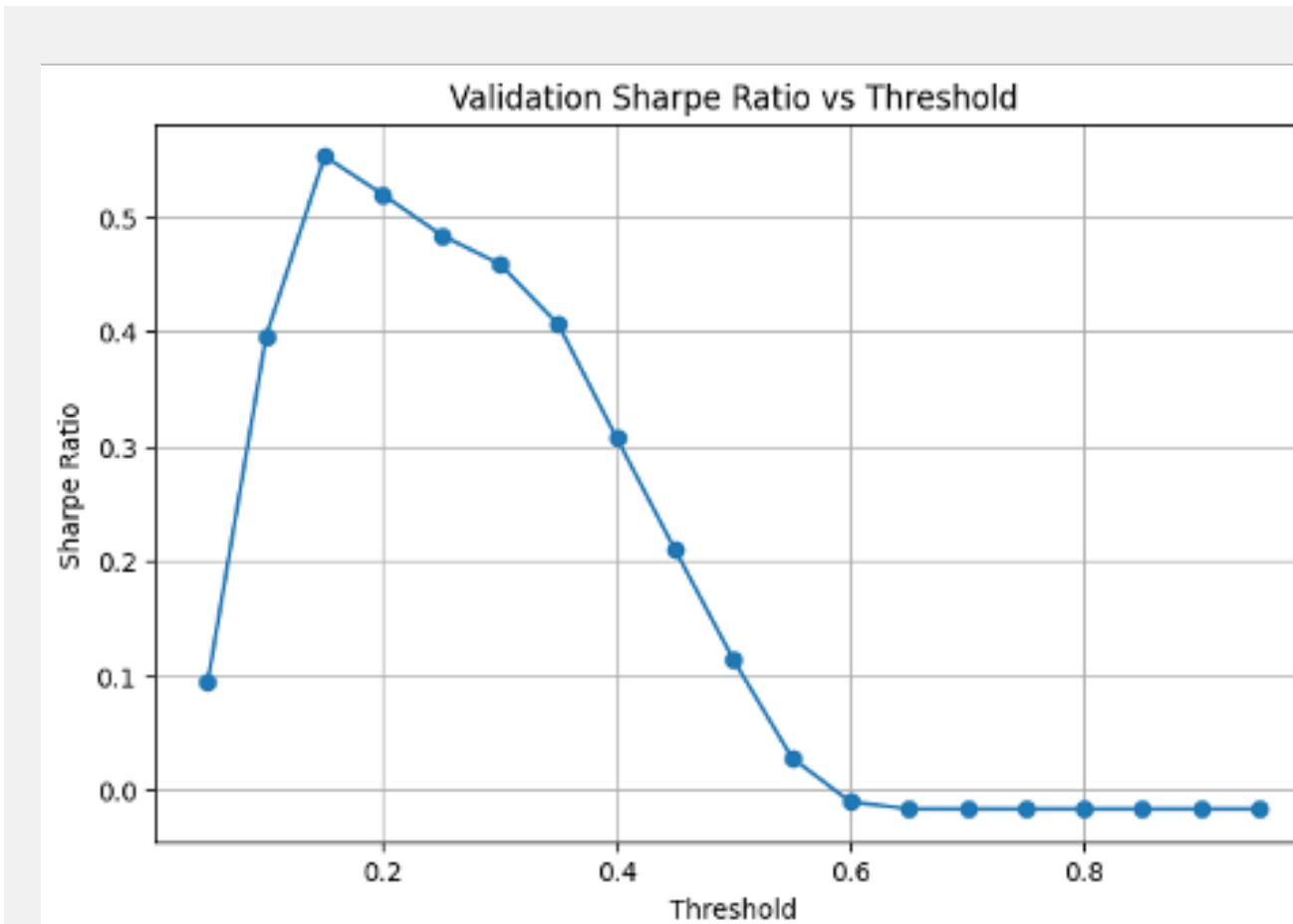
	Random Forest	XGBoost	LightGBM
✓ Best model index	23	5	11
✓ Best validation Sharpe ratio	0.56	0.70	0.69
✓ Best test Sharpe ratio	0.77	1.14	1.13
✓ Best approval rate	64.7%	59.7%	59.7%
✓ Mean IRR	11.1%	12.1%	12.1%
✓ Positive IRR ratio	97.8%	99.1%	99.0%
✓ Best threshold	0.15	0.05	0.05
✓ Sharpe Ratio	0.55	0.68	0.68
✓ Approval Rate	65.4%	60.3%	60.0%
✓ Mean IRR	7.7%	7.7%	7.7%
✓ Positive IRR Ratio	98.5%	99.3%	99.4%

...

04
결과

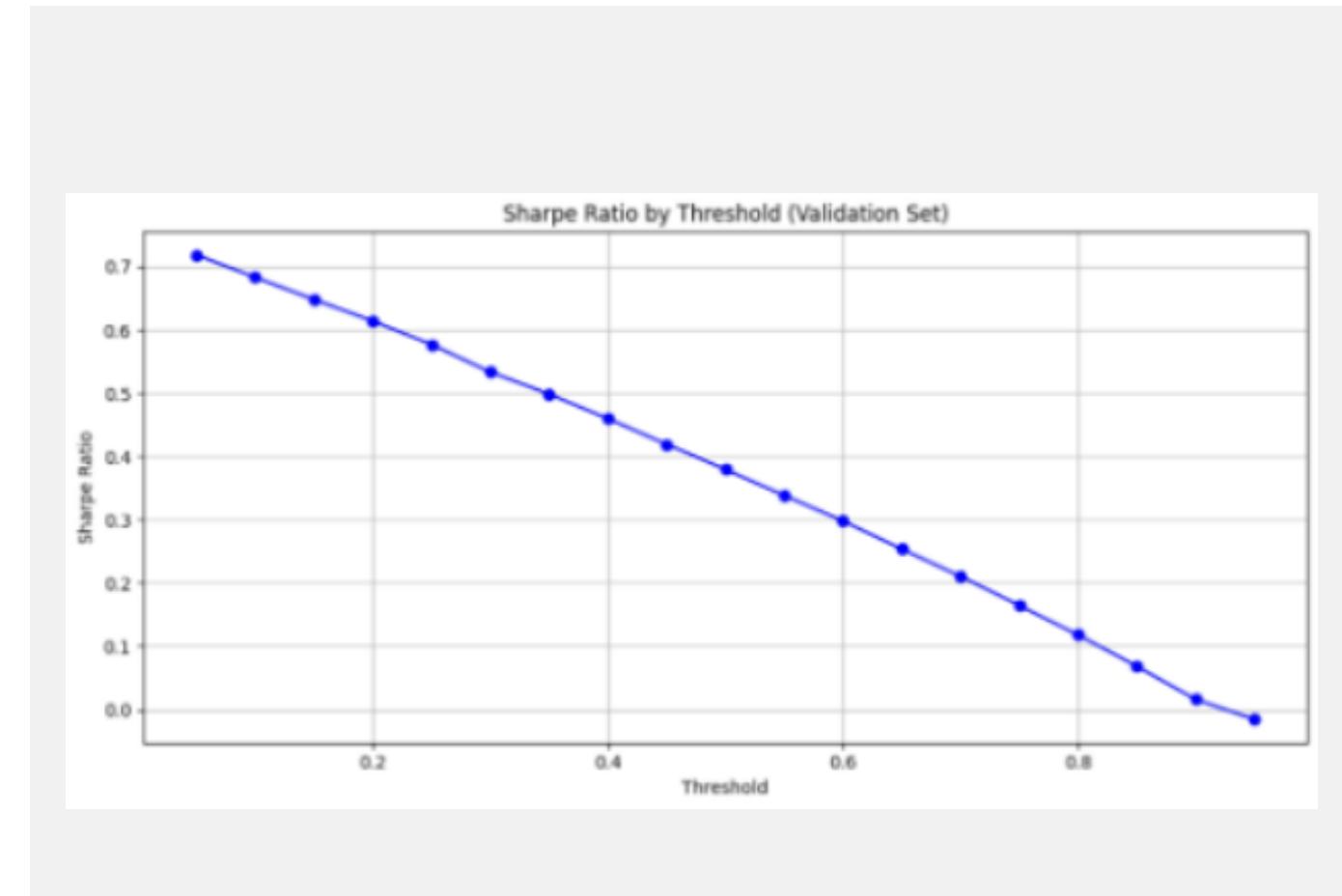
모델 비교 Sharpe Ratio by Threshold

RandomForest



Optimal Threshold: 0.15

XGBoost



Optimal Threshold: 0.05

...

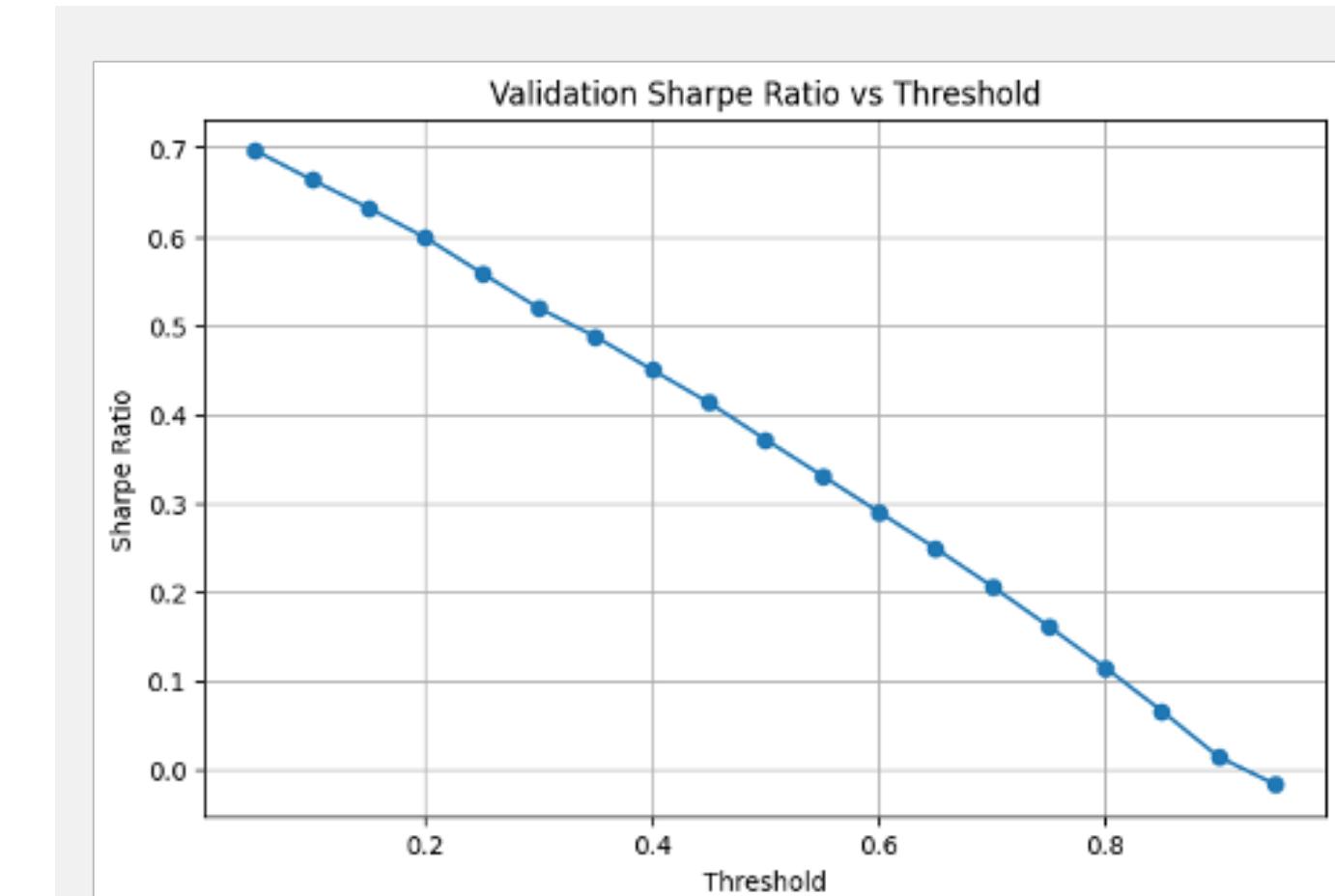
04

결과

모델 비교

Sharpe Ratio by Threshold

LightGBM



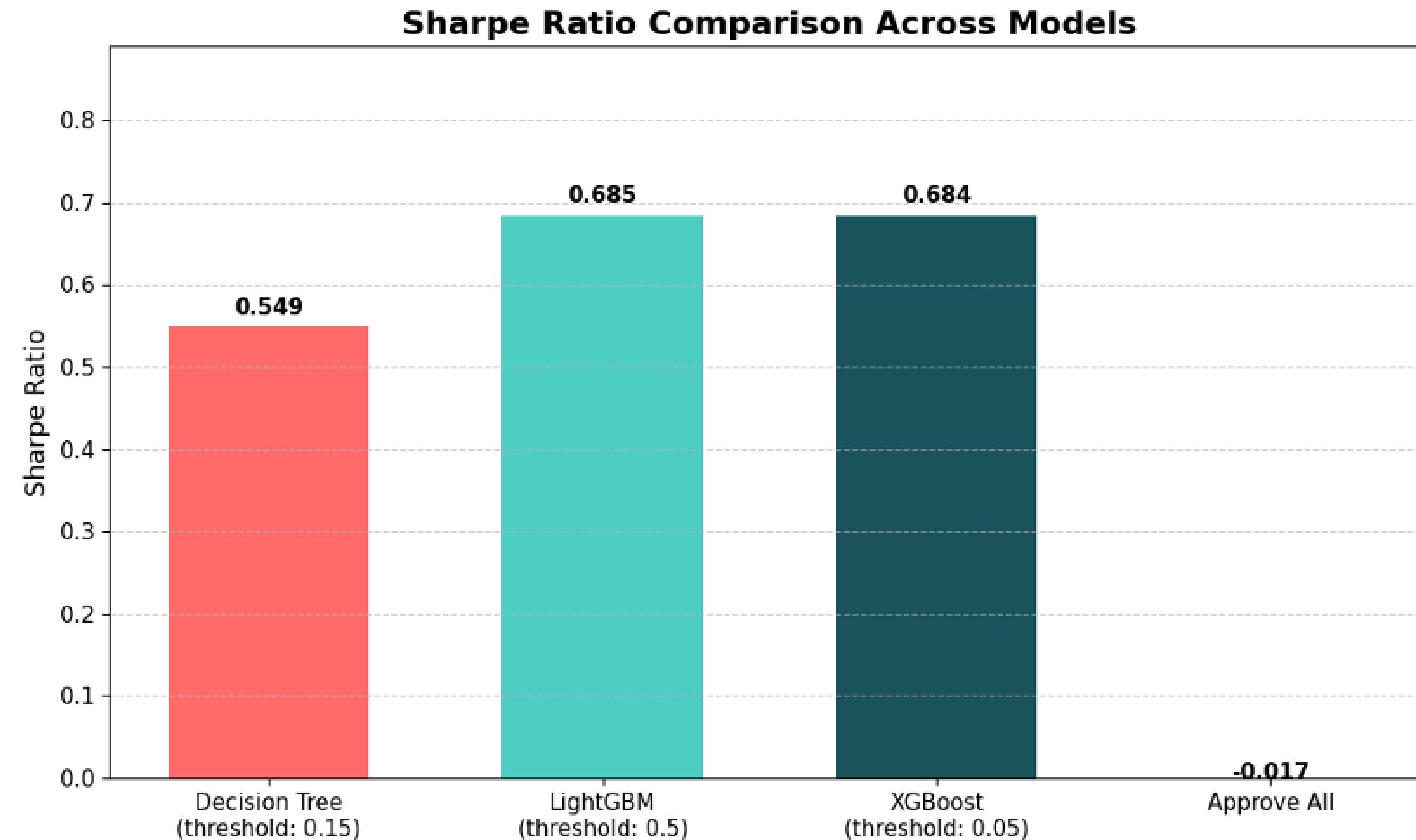
Optimal Threshold: 0.05

...

04

결과

모델 비교 Sharpe Ratio



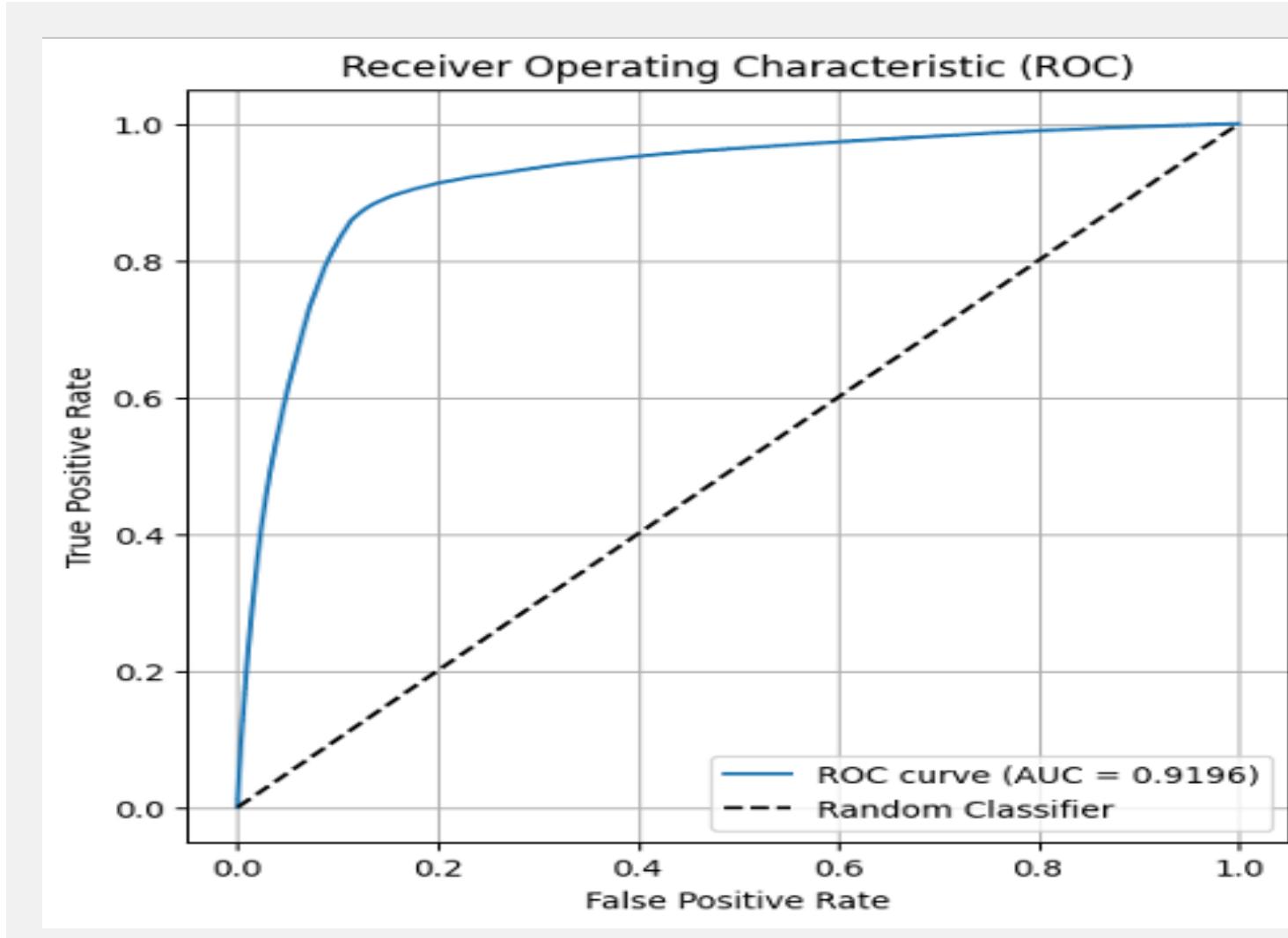
...

04

결과

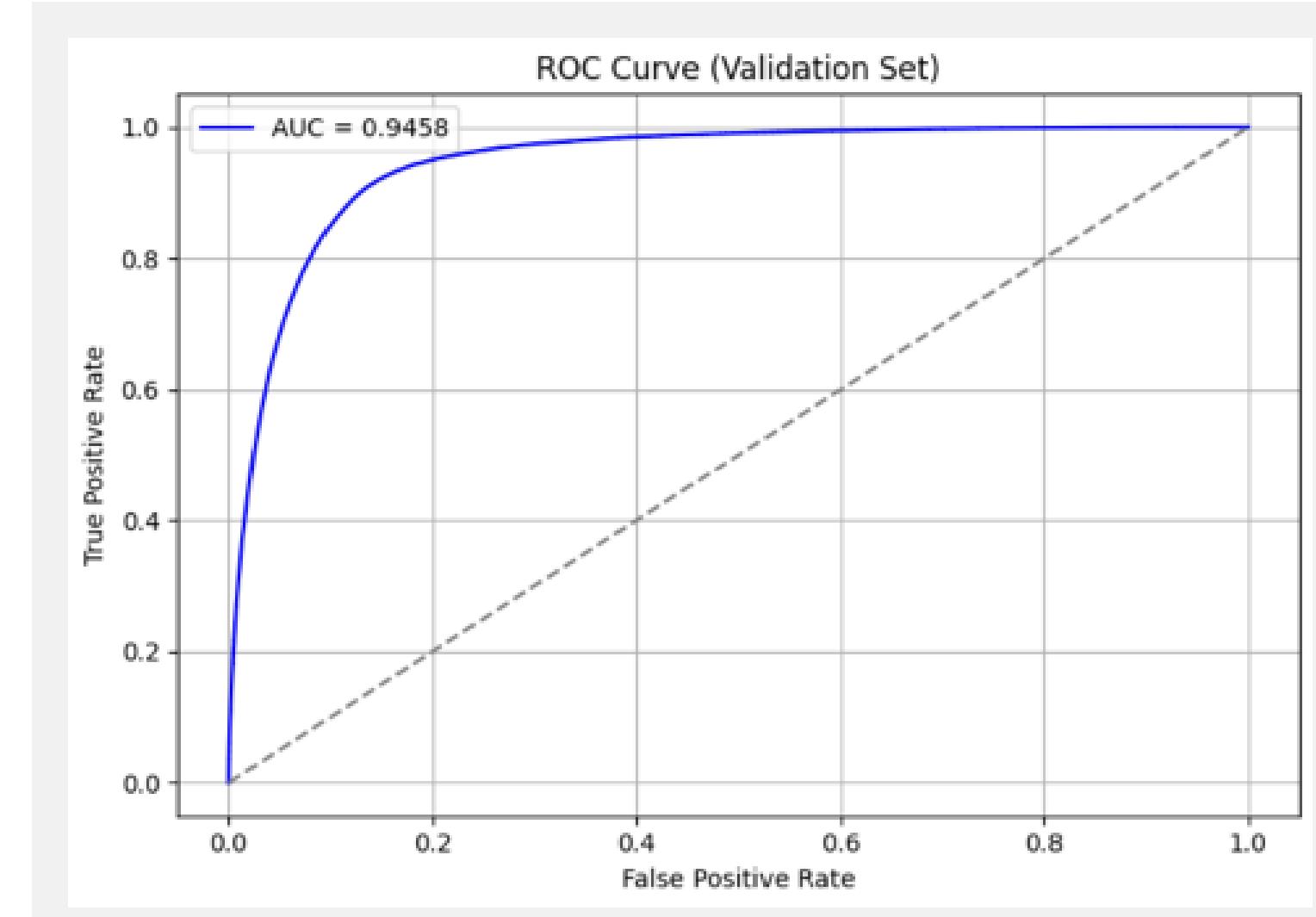
모델 비교 ROC

RandomForest



AUC: 0.9196

XGBoost



AUC: 0.9450

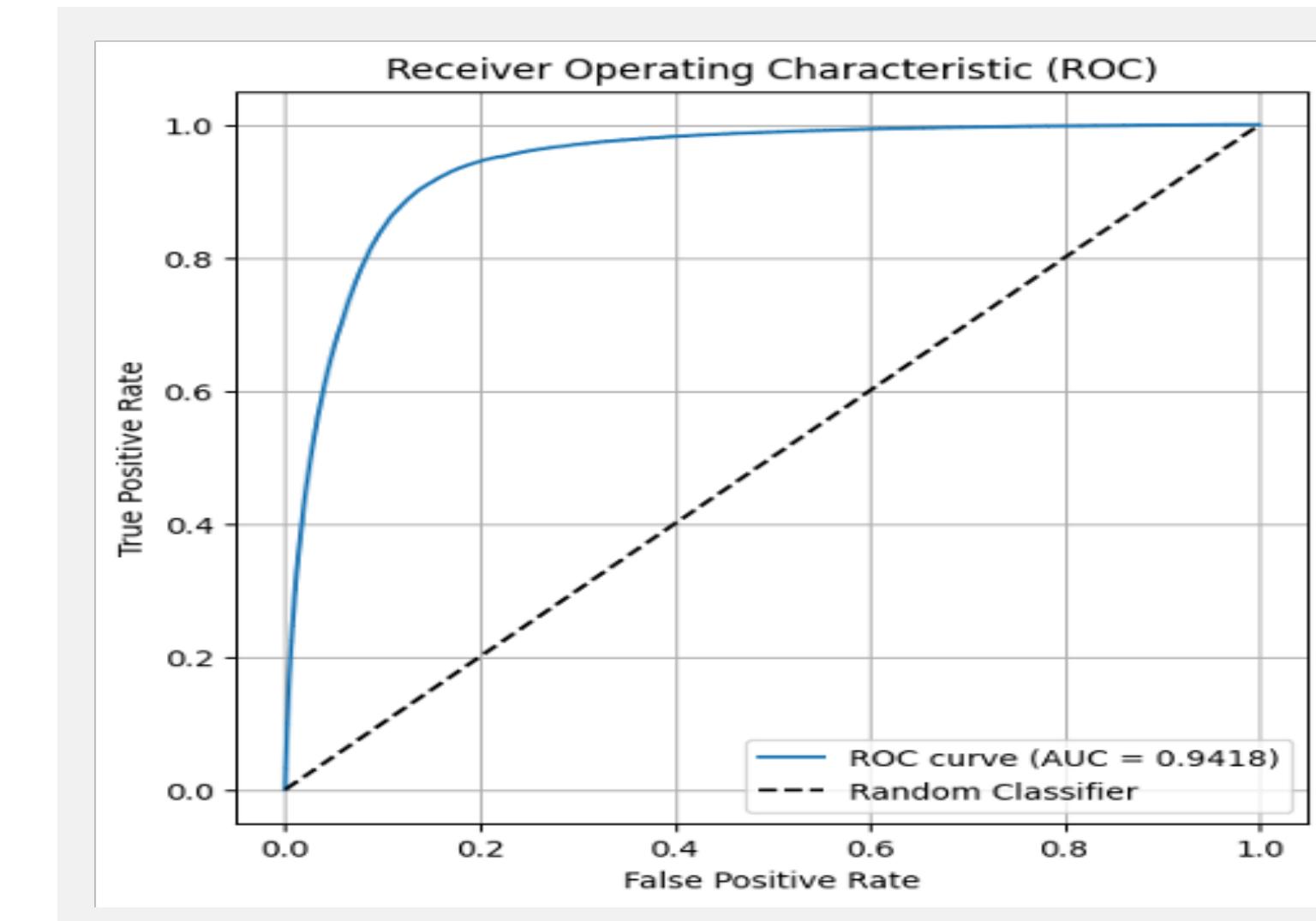
...

04

결과

모델 비교 ROC

LightGBM



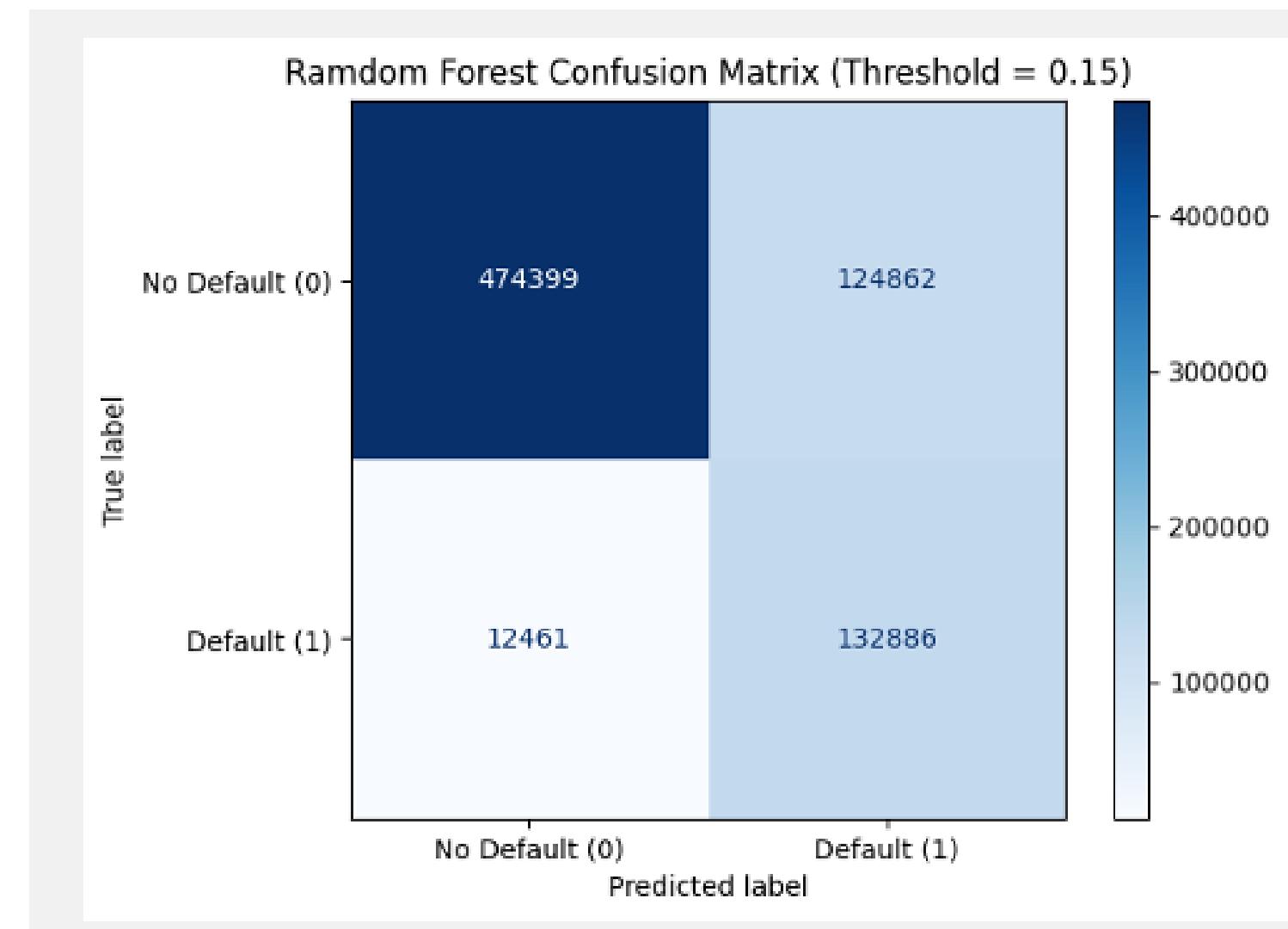
AUC: 0.9418

...

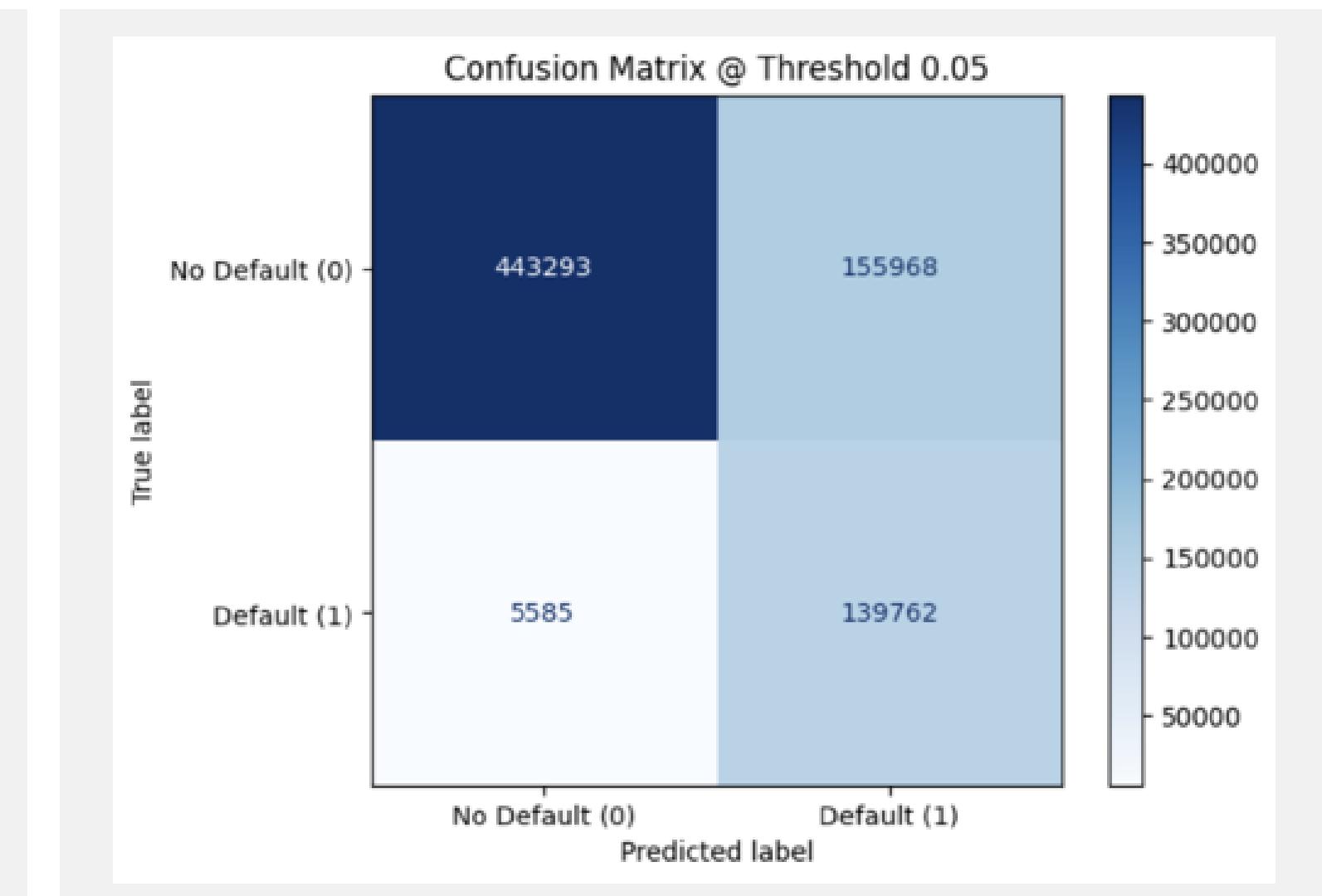
04
결과

모델 비교 Confusion Matrix

RandomForest



XGBoost



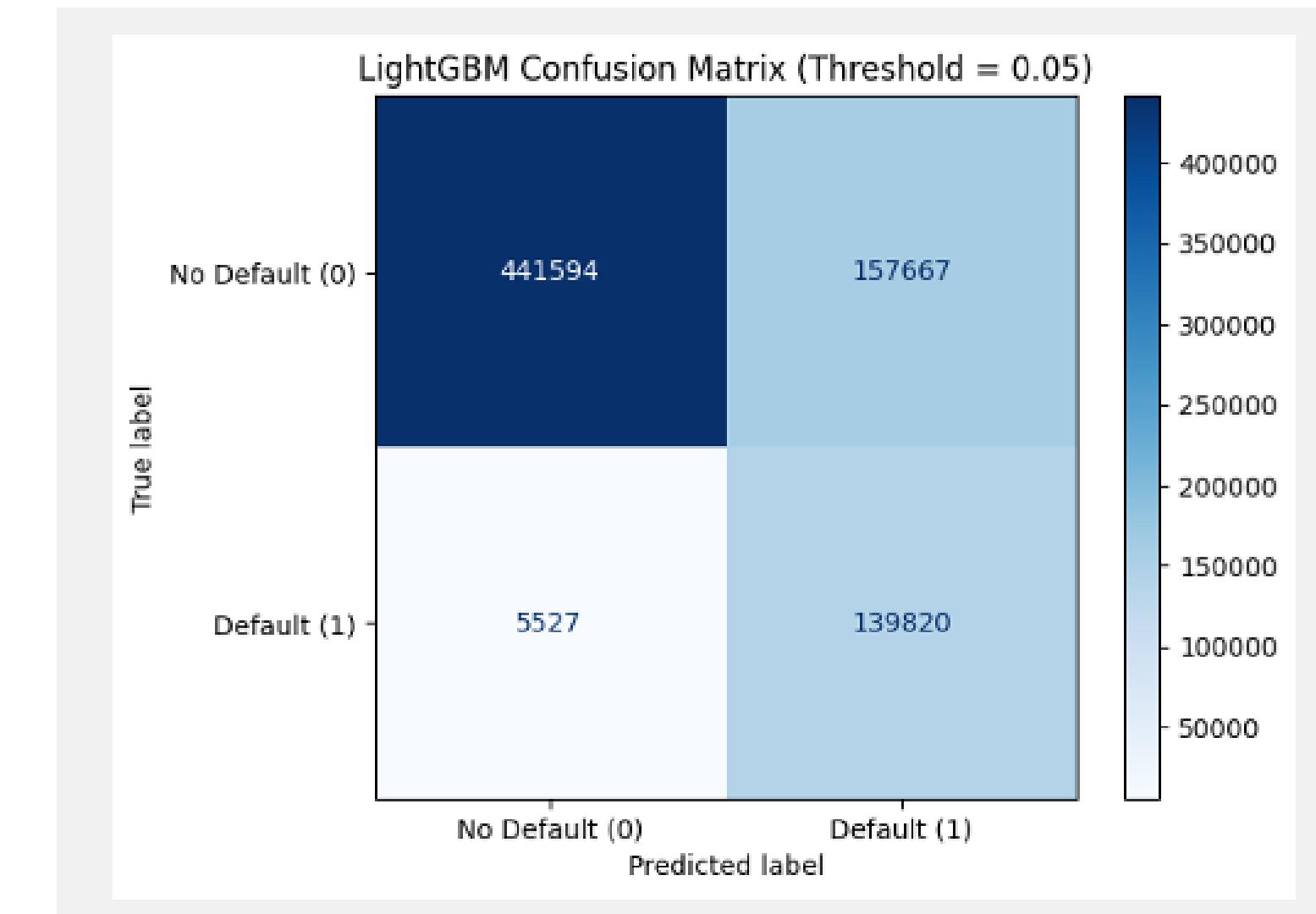
...

04

모델 비교 결과

모델 비교 Confusion Matrix

LightGBM





04

결과

인사이트

한계점

- 1 Sharpe Ratio 표준편차 계산 시 대출 규모별
가중치 및 각 대출 상품별 상관관계 고려 X
- 2 Lending Club 변수명의 모호함
- 3 RandomForest 하이퍼파라미터 튜닝 시
Sharpe Ratio(튜닝 전) > Sharpe Ratio(튜닝 후)
- 4 Sharpe Ratio 1 미만 나오는 것
0.55 / 0.68 / 0.68

보완 방향

- 1 고려 시 포트폴리오 이론의 확장 가능
- 2 각 논문, 자료, 사례마다 전부 다르기 때문에
Lending Club에 직접 문의 필요
- 3 이유: 메모리 부족으로 튜닝 작업 시 AUC를 적용
*AUC = 부도 예측 가능성 UP
신용등급이 높은 대출 상품에 투자가 몰려 하락할
것을 의미하기 때문에 Sharpe Ratio 감소 예상
- 4 변수 선택을 보수적으로 진행해 신중한 모델 제작

Q & A

THANK YOU!

05

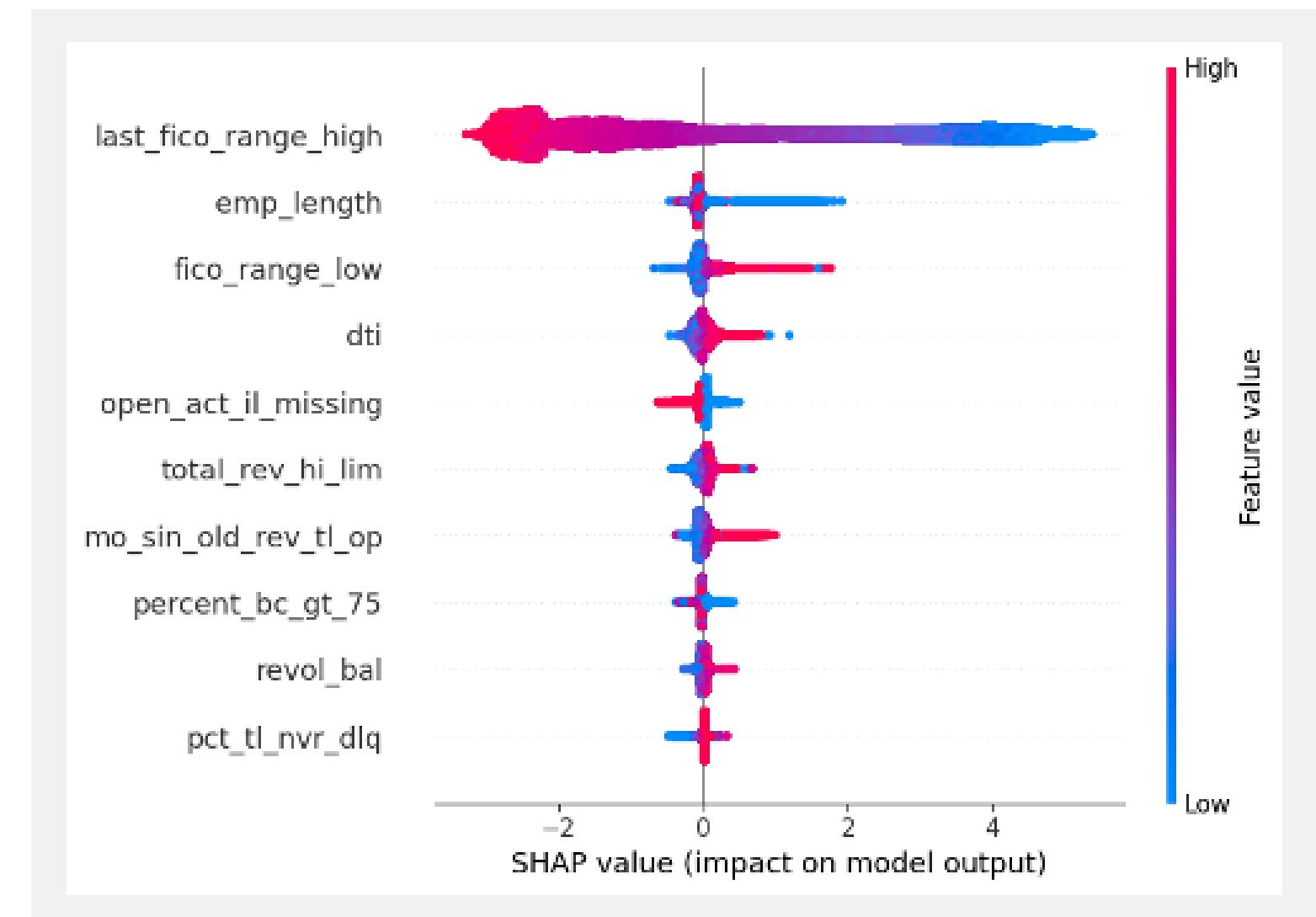
모델 비교 Shapley value

오류 인식

RandomForest



XGBoost



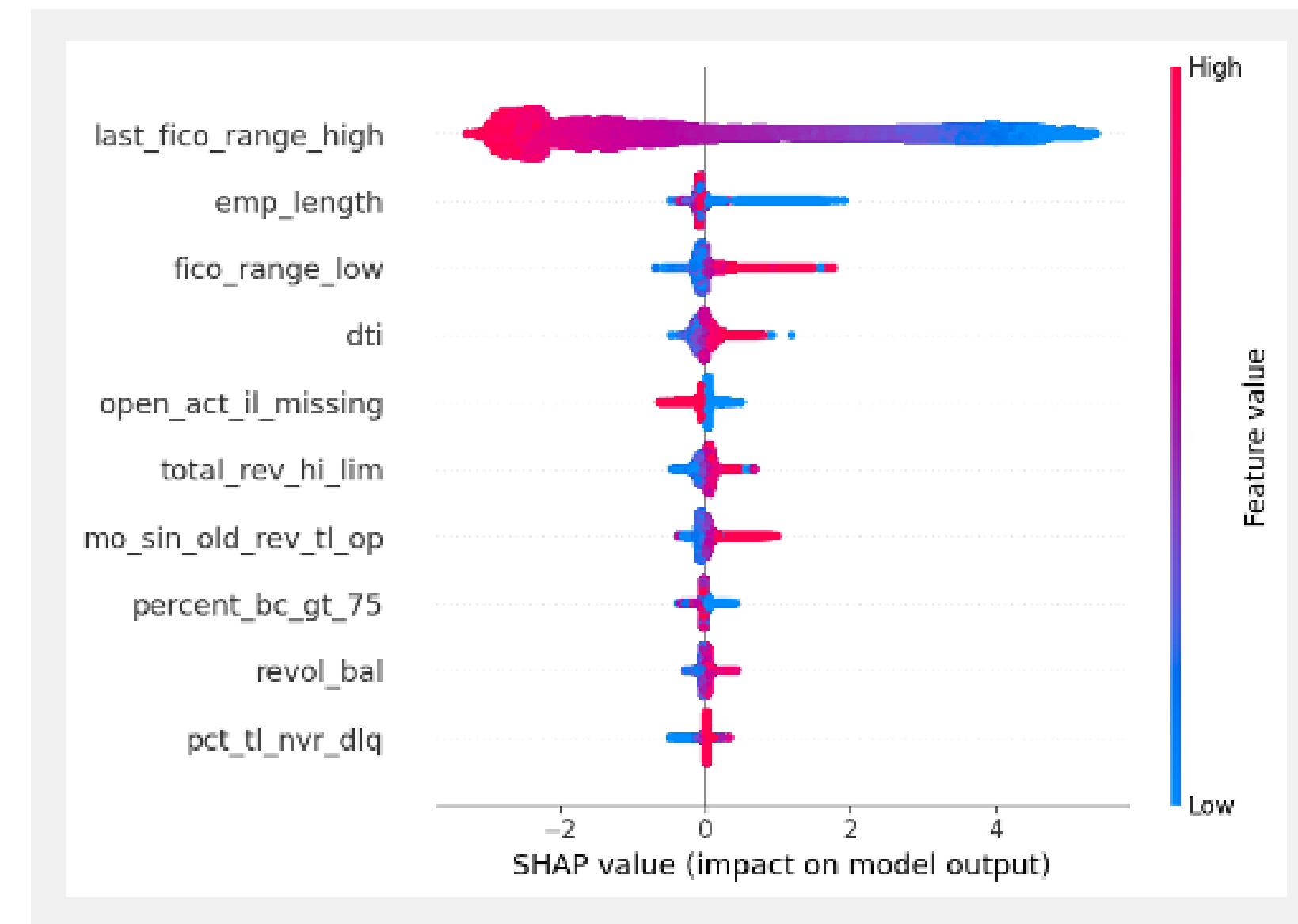
05

모델 비교

Shapley value

오류 인식

LightGBM



05

Re - 모델링

인식 후 개선

RandomForest

