

섹션2 머신러닝 프로젝트

이 정도면 거의 보험사기

Codestates AIB 14

박성희

진행 순서

의료 보험금 청구건의 사기(Fraud)
여부 판별

문제 정의
데이터 소개

모델링

결론 - 보완사항

데이터 전처리
EDA & 시각화

최종 모델



문제 정의 & 데이터 소개

문제 정의

- 의료서비스 제공자의 의료보험 청구 사기 판별

문제 해결의 필요성

- 의료보험 사기 청구 배상금이 의료보험 총 지출의 상당 부분을 차지
- 보험사의 재정적 피해
- 보험료 인상에도 영향

문제 정의 & 데이터 소개

데이터 소개

- 1. 민간 의료보험 가입자 정보
- 2. 입원환자 진료 및 보험금 청구 내역
- 3. 외래환자 진료 및 보험금 청구 내역
- 4. Provider별 잠재적 사기여부 정보

문제 유형

- **분류** (0: non-fraud / 1: fraud) 모델
- 타겟 변수: **PotentialFraud** ['Yes', 'No']

문제 해결에 의미 있는
가설

- 1 가입자의 나이
- 2 가입자의 기저질환
- 3 보험금^{(청구.배상),}
납부한 보험료
- 4 보험금
청구기간

보험 청구의 **사기** 분류 예측에 유의미한
영향을 미칠 것이다.

데이터 전처리 EDA & 시각화



DOB, DOD

DOB: Date of Birth, DOD: Date of Death
pd.to_datetime()

수치형으로 변경 - **Patient_Age** 칼럼 신규 생성
patient_age = df['DOD'] - df['DOB']
df['Patient_Age'] = patient_age/365

DOD: NaN - 가입자 현재 생존해 있음
데이터셋 상의 최신 날짜로 처리해서 나이 산출

데이터
전처리
&
특성공학

데이터 전처리 & 특성공학

ChronicCon_*

RenalDiseaseIndicator

```
# ['Yes', 'No']  
[1, 2] > [1, 0]
```

수치형으로 변경 - Patient_Risk_Score 칼럼 신규 생성

```
.apply(pd_to_numeric)
```

RenalDiseaseIndicator

```
# ['Yes', 'No']  
[1, 2] > [1, 0]
```

데이터 전처리 & 특성공학

AdmissionDt, DischargeDt

AdmissionDt, DischargeDt: 입원, 퇴원 날짜
(YYYY-MM-DD)

수치형으로 변경 - **Duration_IP** 칼럼 신규 생성
Duration_IP: 입원기간

ClaimStartDt, ClaimEndDt

ClaimStartDt, ClaimEndDt: 보험금 청구 시작, 종료 날짜
(YYYY-MM-DD)

수치형으로 변경 - **Claim_Period** 칼럼 신규 생성
Claim_Period: 총 보험금 청구 기간

Claimed_extra 생성 : 실제보다 추가 청구된 기간이 있는지

데이터 전처리 & 특성공학

IPAnnualReimbursementAmt, OPAnnualReimbursementAmt

IPAnnualReimbursement, OPAnnualReimbursement
: 연간 입원/외래환자의 최대 보험금

TotalReimbursementAmt 칼럼 신규 생성

TotalReimbursementAmt : 총 배상가능 보험금액

IPAnnualDeductibleAmt, OPAnnualDeductibleAmt

IPAnnualDeductibleAmt, OPAnnualDeductibleAmt
: 연간 입원/외래환자의 보험료

TotalReimbursementAmt 칼럼 신규 생성

TotalReimbursementAmt : 총 납부 보험료 (연)

데이터 전처리 & 특성공학

Provider 별 feature의 합 칼럼 신규 생성

groupby() 메소드 활용

features

1) *InscClaimAmtReimbursed* 보험사배상액

2) *DeductibleAmtPaid* 보험료 납부 총액

3) IP/OP 배상액, 납부 보험료

4) *Patient_Risk_Score* (만성질환 여부 기반) 위험 점수

PerProvider_count_ClaimID : Provider별 청구 건수

불필요한 칼럼 삭제

ClaimStartDt, *ClaimEndDt* # DOD, DOB

Physician 정보 (3)

ClmDiagnosisCode (10) # *ClmProcedureCode* (6)

*ChronicCon_** (all) + *State*, *County* 등

인코딩 & 결측치 처리

성별, 인종 칼럼 인코딩

데이터타입 변경 (`int64 > category`)

카디널리티: 성별2, 인종4

원핫인코딩

결측치 처리

대부분 merge로 인해 발생 (예. 입원기간)

: 0으로 처리

EDA

입원환자 배상 규모

총 배상금액: \$408,297,020

청구 건당 평균 배상금액: \$10,088

사기 청구 건 배상 비율: 59.10%

외래환자 배상 규모

총 배상금액: \$71,921,550

청구 건당 평균 배상금액: \$284

사기 청구 건 배상 비율: 37.42%

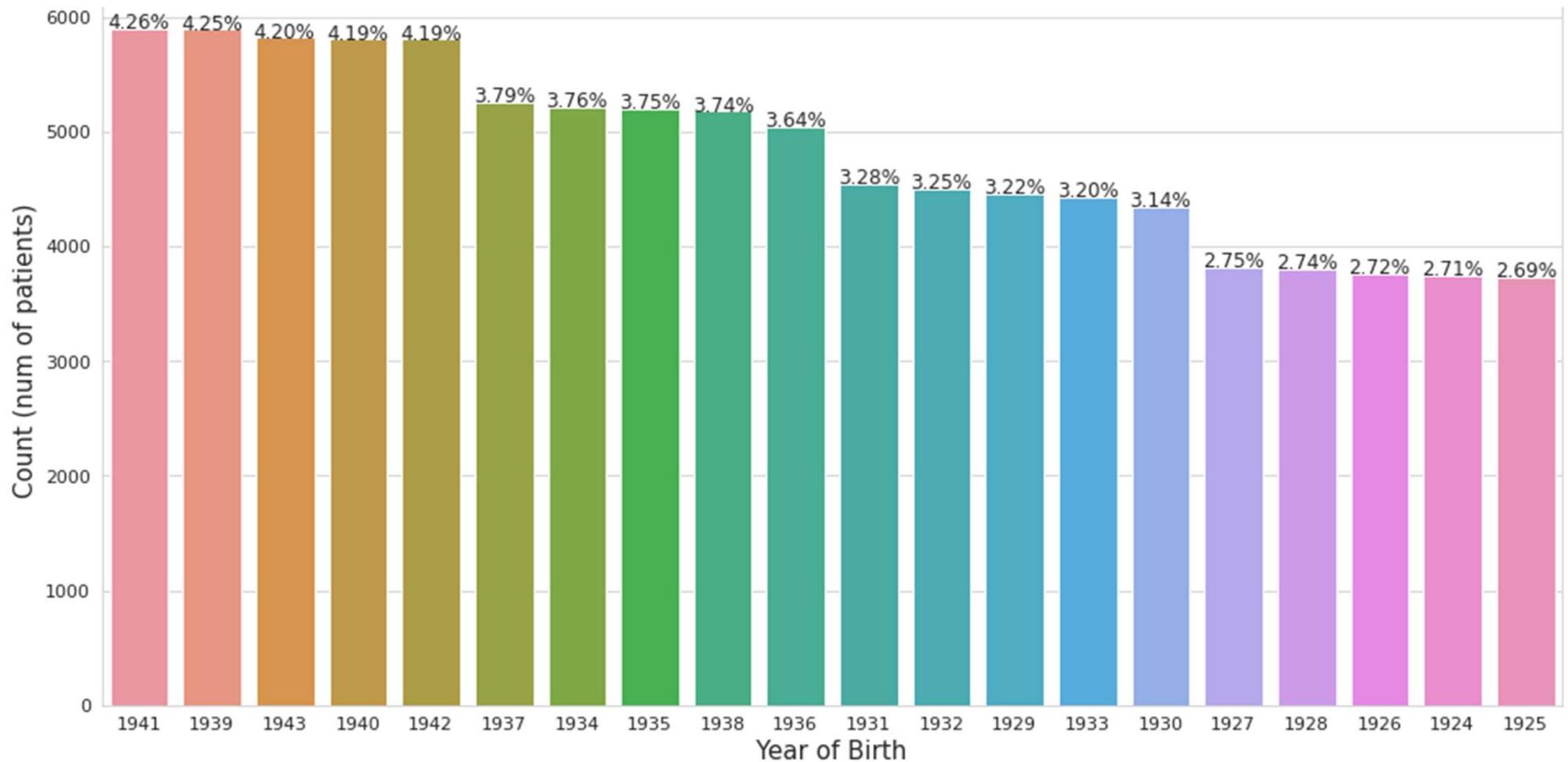
총 배상 규모

총 배상금액: \$480,218,570

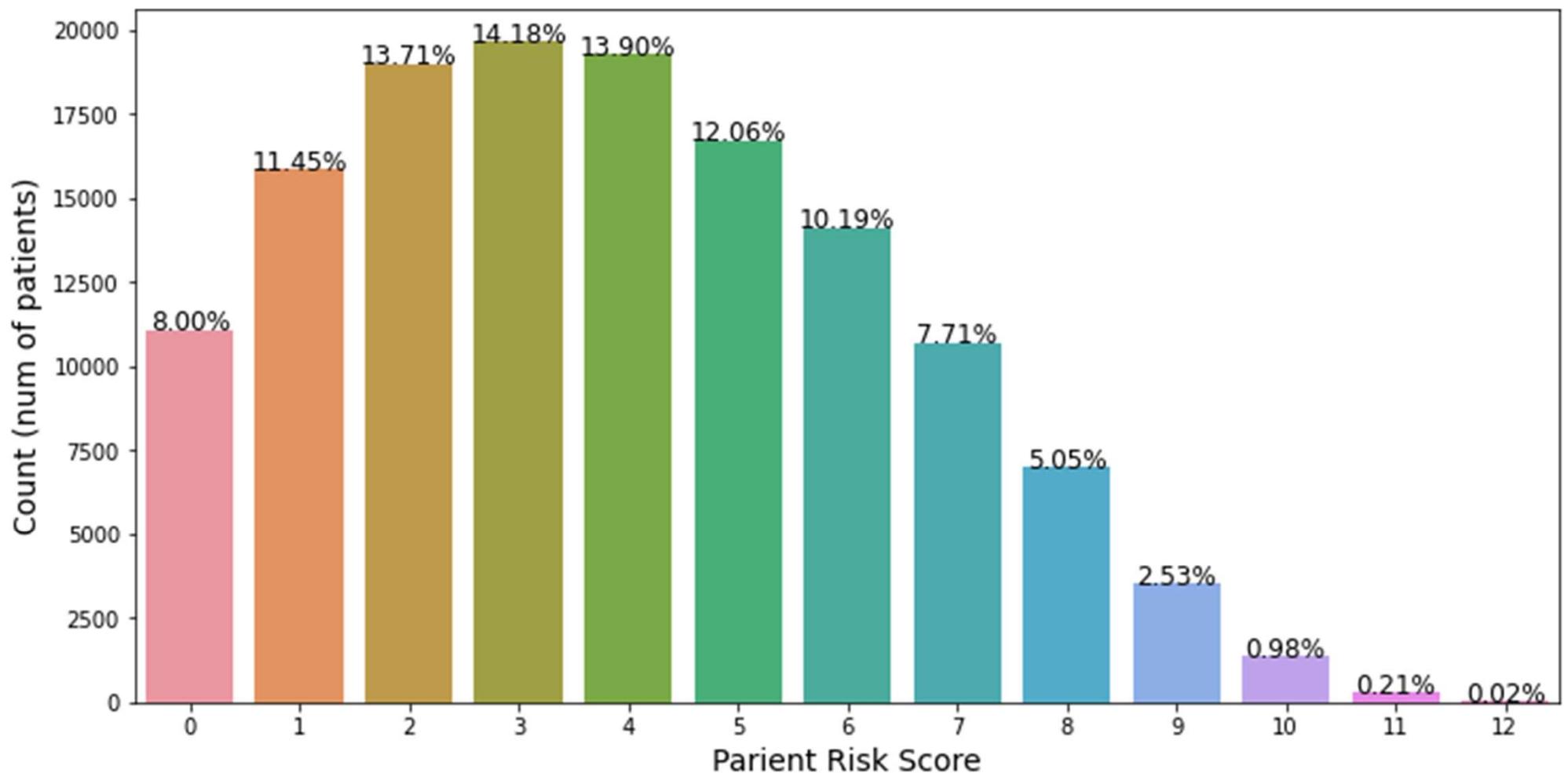
사기 청구 건 배상금액: \$268,200,290

사기 청구 건 배상 비율: 55.85%

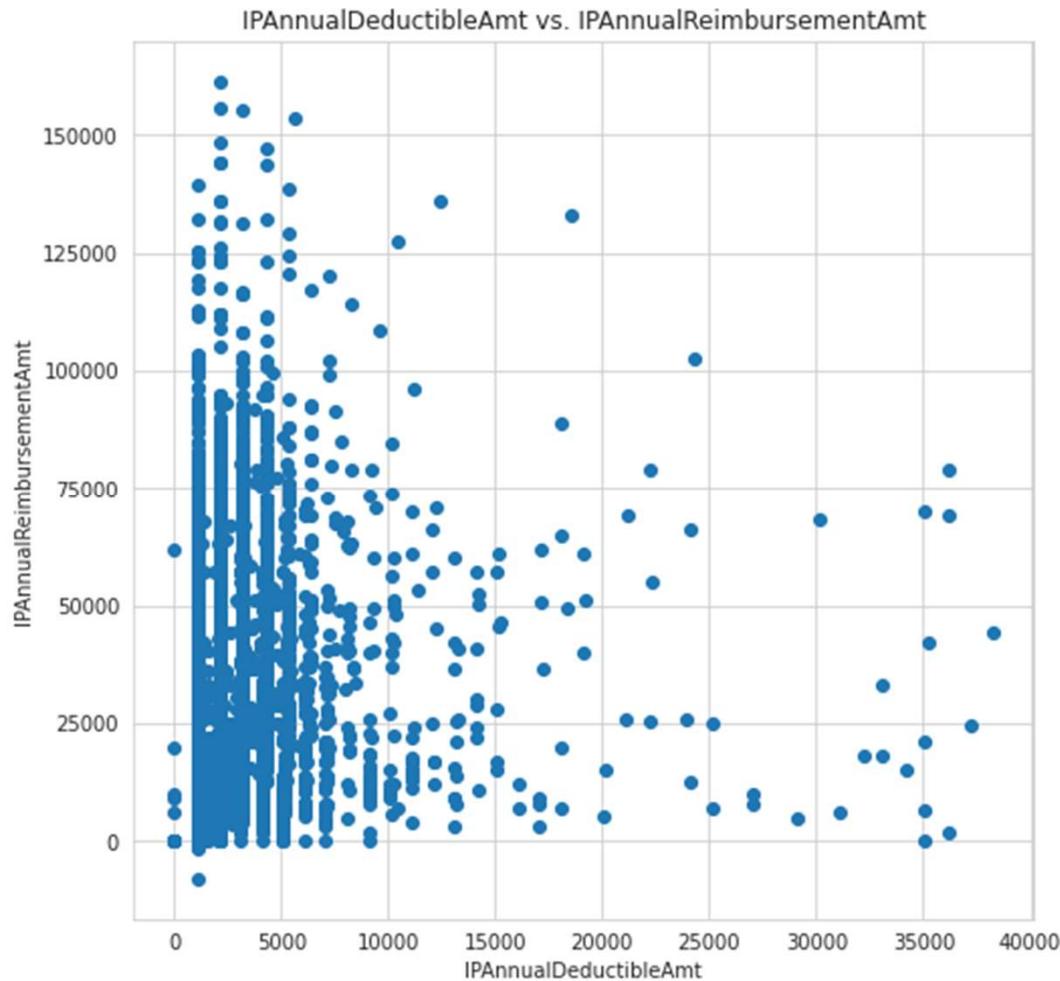
Distribution of Age



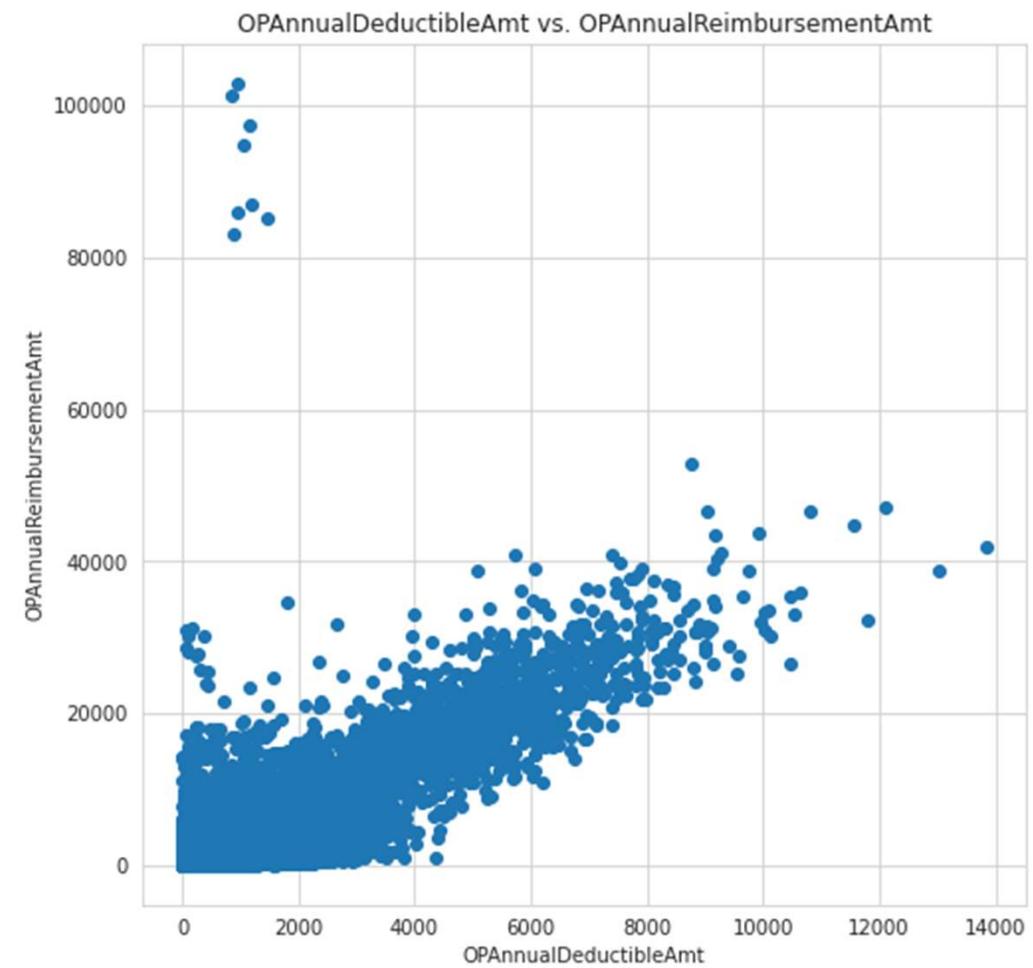
Distribution of Patient Risk Score



IPAnnualDeductibleAmt vs IPAnnualReimbursementAmt

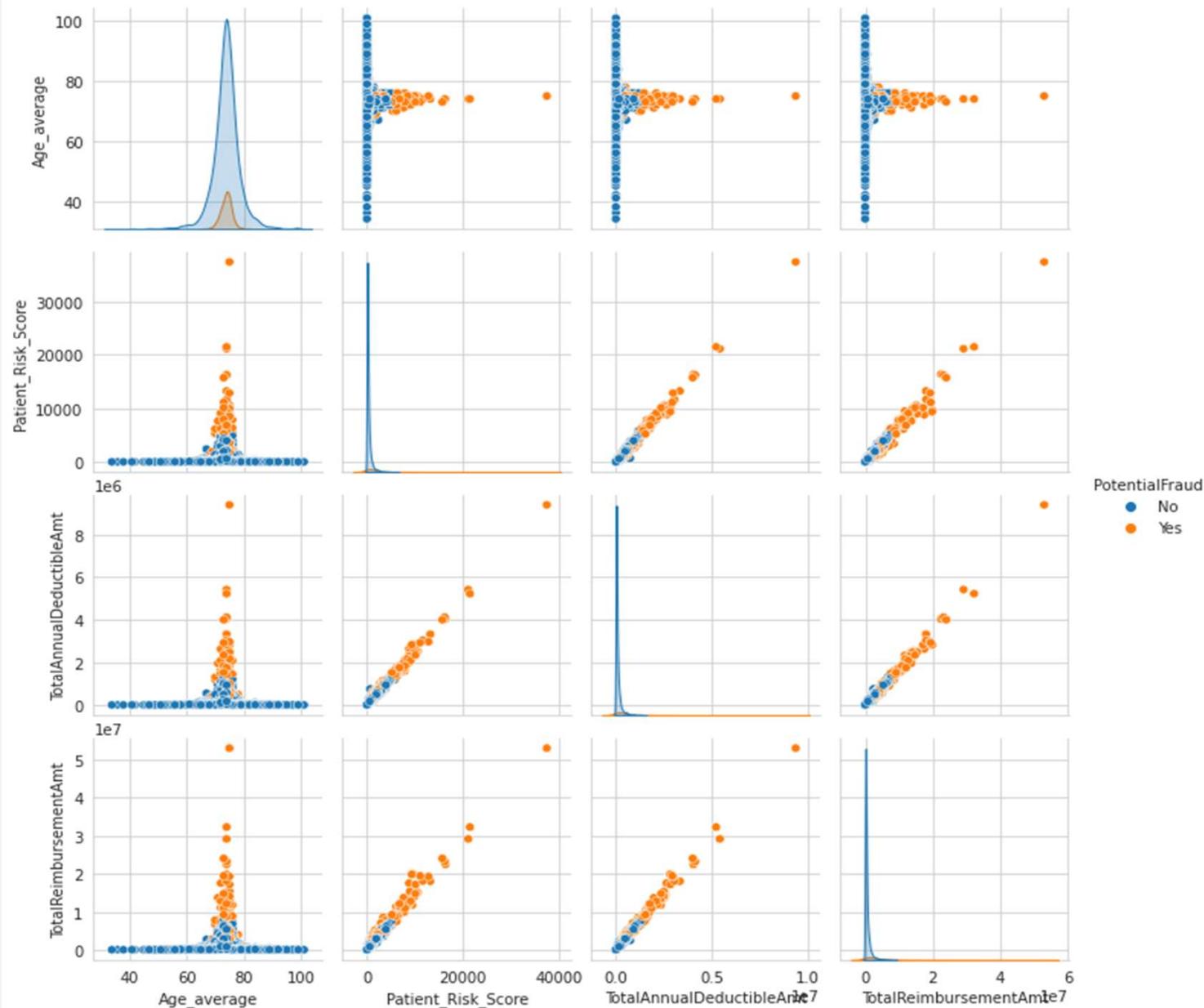


OPAnnualDeductibleAmt vs OPAnnualReimbursementAmt



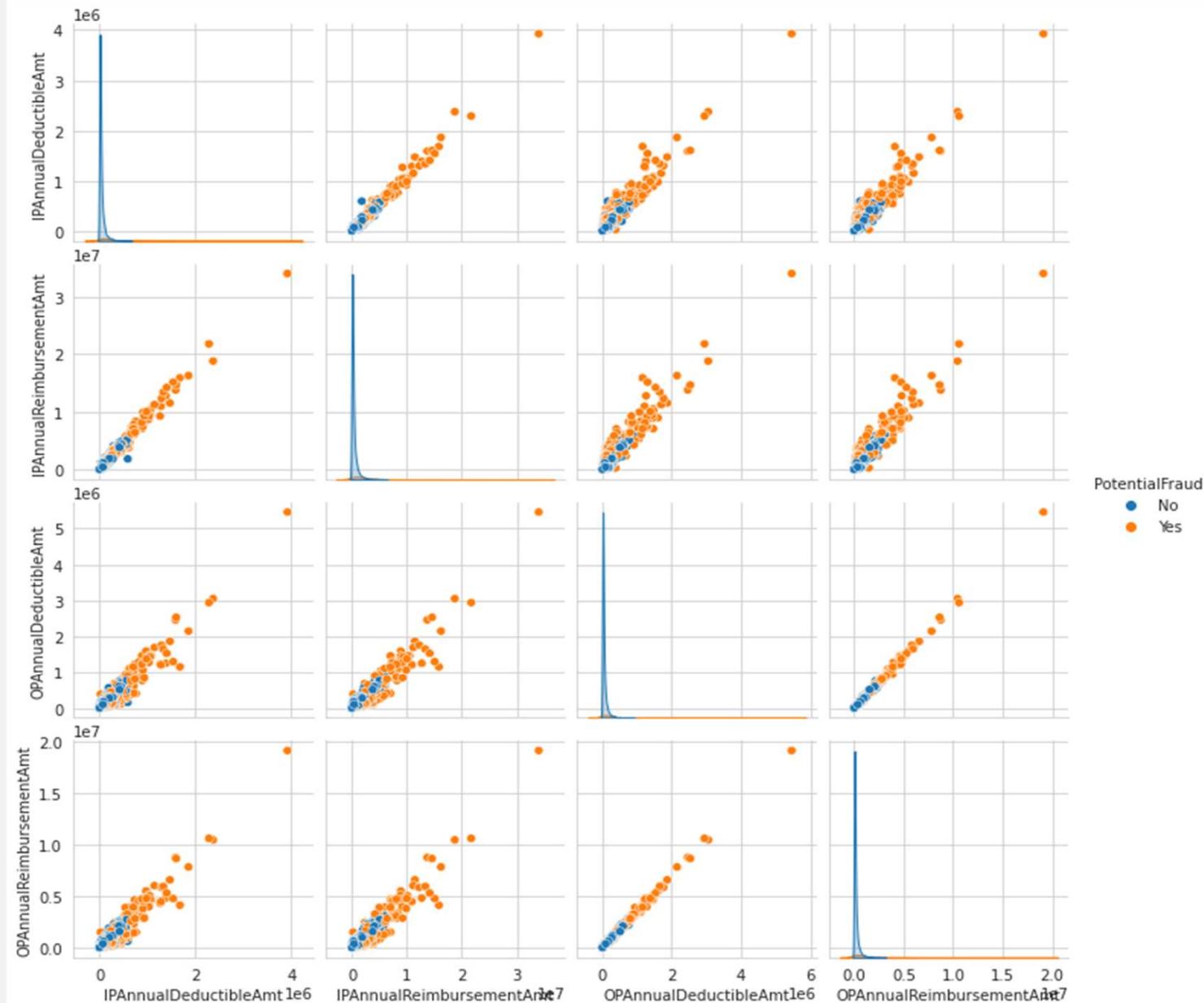
피처 간 상관관계

- 나이
- 환자 위험점수
(기저질환)
- 청구보험금(외래+입원)
- 납부보험료 (외래+입원)

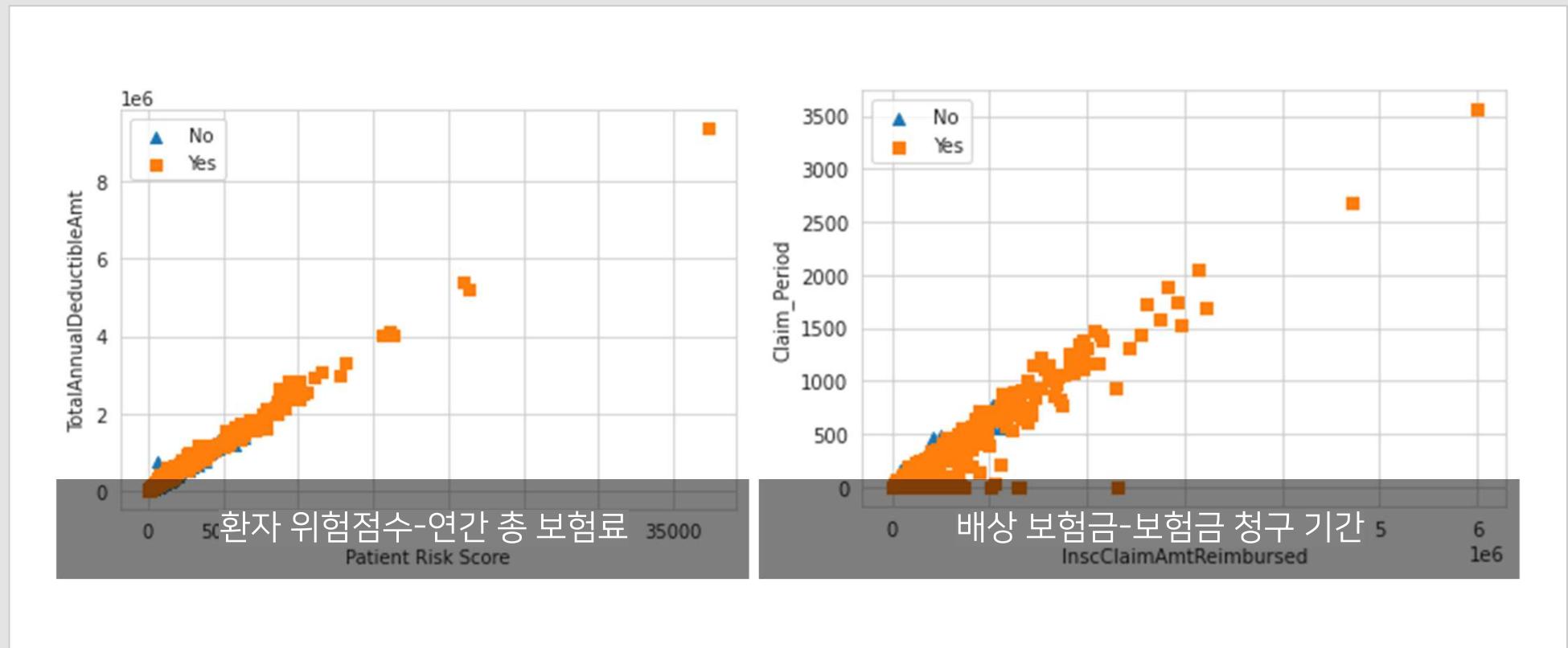


피처 간 상관관계

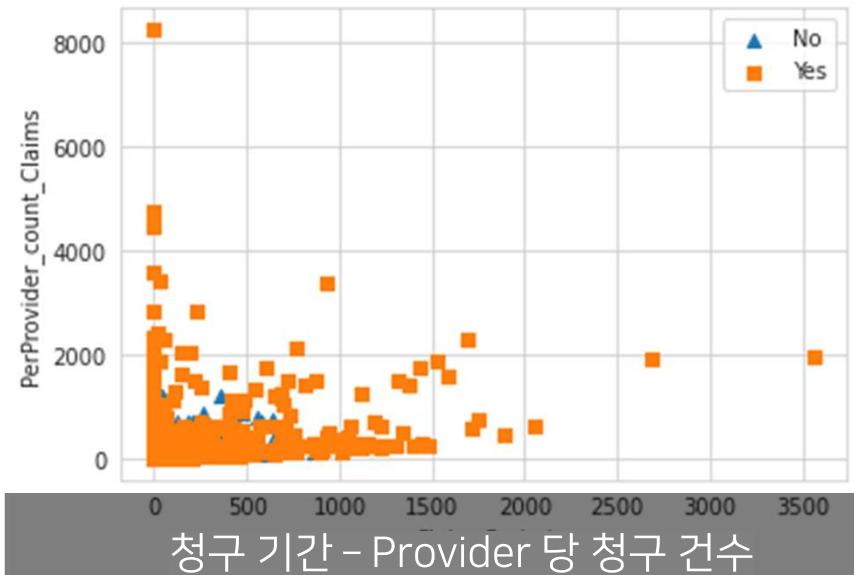
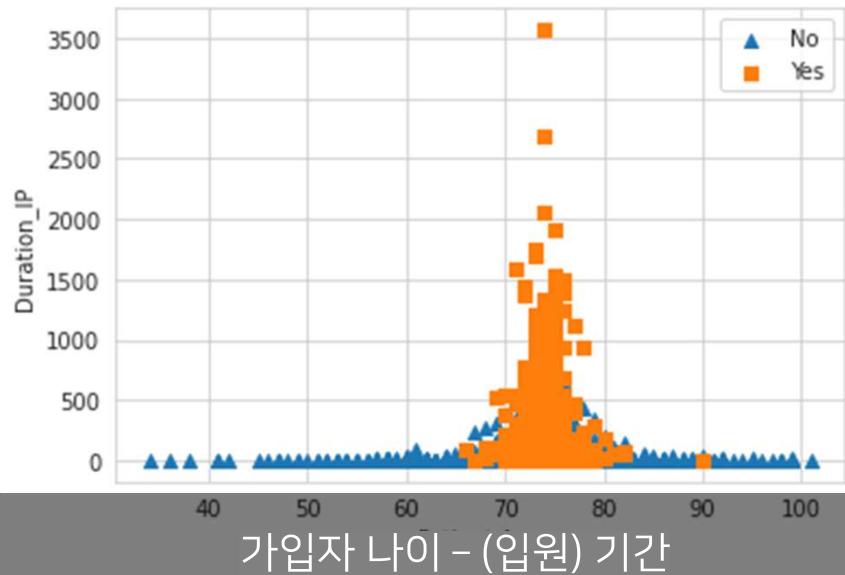
- (입원)배상 최대보험금
- (입원)보험료
- (외래)배상 최대보험금
- (외래)보험료



타겟 클래스 별 데이터 분포



타겟 클래스 별 데이터 분포





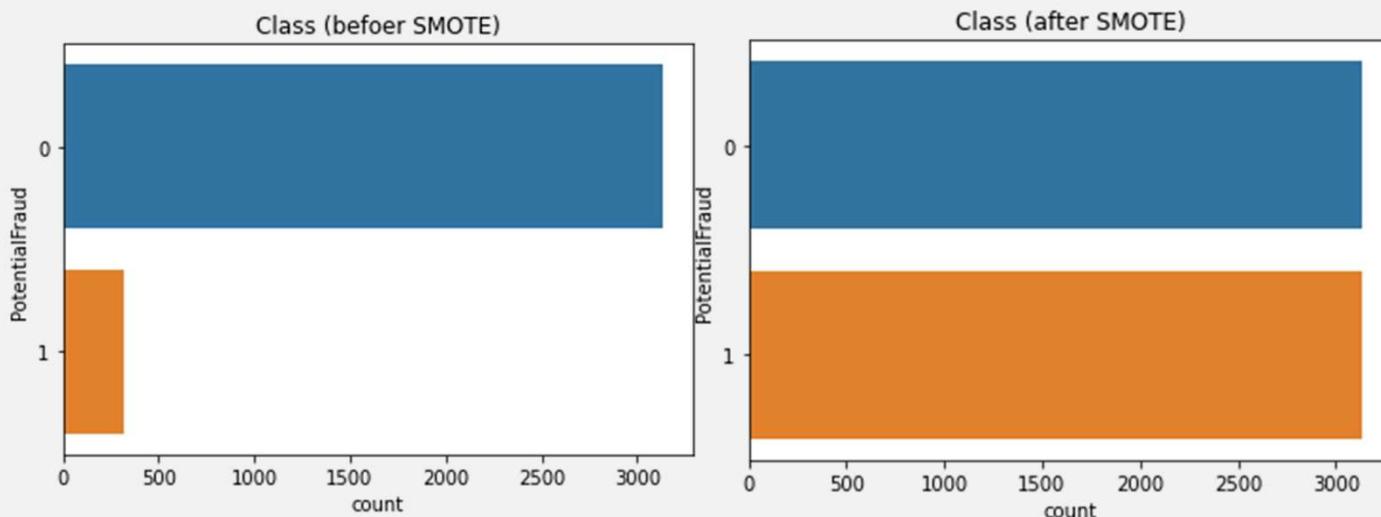
모델링

기준모델 선정
모델 구축
모델 검증
최적화
모델 선택

기준모델의 선정. 모델링 성능 평가

```
# 0 = No, 1 = Yes  
target = 'PotentialFraud'  
  
major = y_train.mode()[0]  
y_pred = [major] * len(y_train)  
accuracy_score(y_train, y_pred)
```

기준모델 학습 정확도: 0.9064 (약 90.6%)



기준모델

모델 검증

- 문제의 유형: 분류
- 모델군
 - LogisticRegression
 - DecisionTreeClassifier
 - RandomForestClassifier
 - XGBClassifier
 - AdaBoost
- 평가지표: ROC AUC score, f1-score 등

*타겟 클래스 불균형

모델링

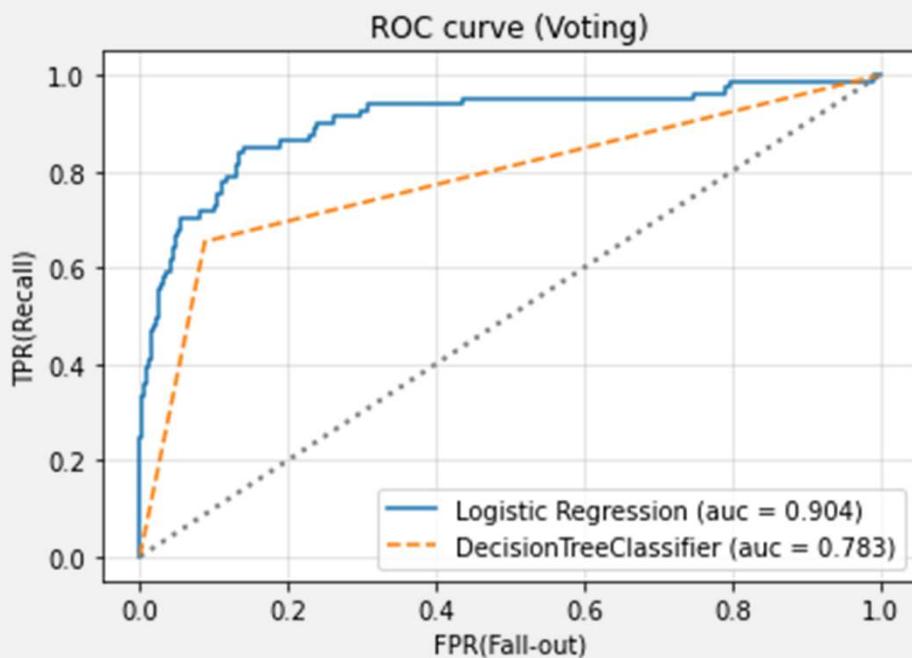
- 모델 구축
- 모델 검증
 - Confusion Matrix
 - Classification Report
 - cross_val_score
 - ROC curve

모델 검증

- LogisticRegression vs. DecisionTree

ROC AUC: 0.933 (+/- 0.012) [Logistic Regression]

ROC AUC: 0.925 (+/- 0.028) [DecisionTreeClassifier]



```
Confusion Matrix :  
[[656 129]  
[ 12  69]]  
Accuracy : 0.8371824480369515  
Precision : 0.3484848484848485  
Recall : 0.8518518518518519  
f1 score : 0.4946236559139786  
roc_auc score : 0.9040025163167413  
-----  
Confusion Matrix :  
[[709  76]  
[ 28  53]]  
Accuracy : 0.8799076212471132  
Precision : 0.4108527131782946  
Recall : 0.654320987654321  
f1 score : 0.5047619047619047  
roc_auc score : 0.7787528505150586
```

모델링

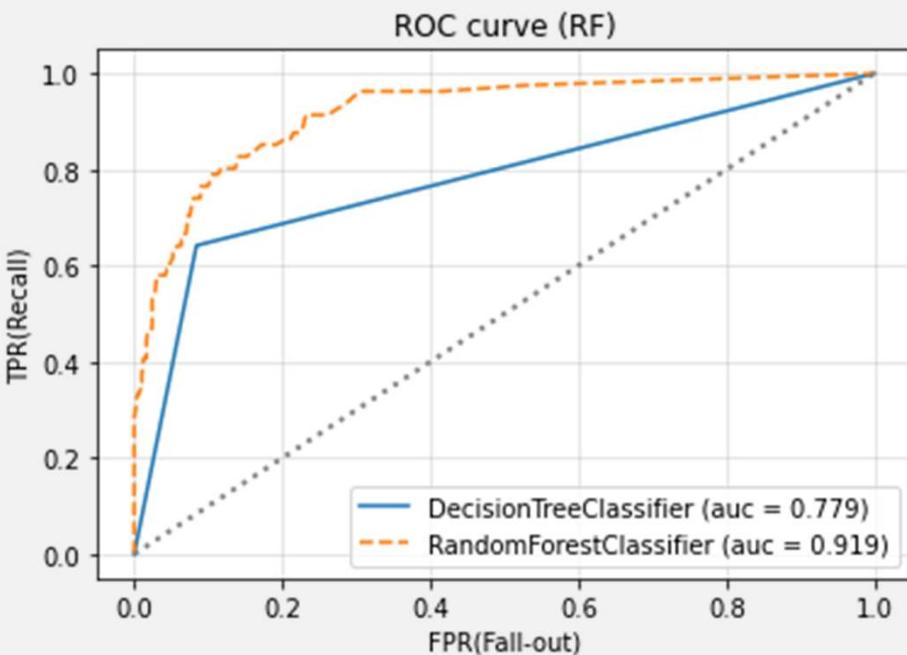
- 모델 구축
- 모델 검증
 - Confusion Matrix
 - Classification Report
 - cross_val_score
 - ROC curve

모델 검증

- DecisionTree vs. RandomForest

ROC AUC: 0.925 (+/- 0.027) [DecisionTreeClassifier]

ROC AUC: 0.990 (+/- 0.009) [RandomForestClassifier]



```
Confusion Matrix :  
[[712  73]  
 [ 32  49]]  
Accuracy : 0.8787528868360277  
Precision : 0.4016393442622951  
Recall : 0.6049382716049383  
f1 score : 0.48275862068965525  
roc_auc score : 0.755972320515845  
  
-----  
Confusion Matrix :  
[[738  47]  
 [ 29  52]]  
Accuracy : 0.9122401847575058  
Precision : 0.5252525252525253  
Recall : 0.6419753086419753  
f1 score : 0.5777777777777777  
roc_auc score : 0.9170716363922309
```

모델링

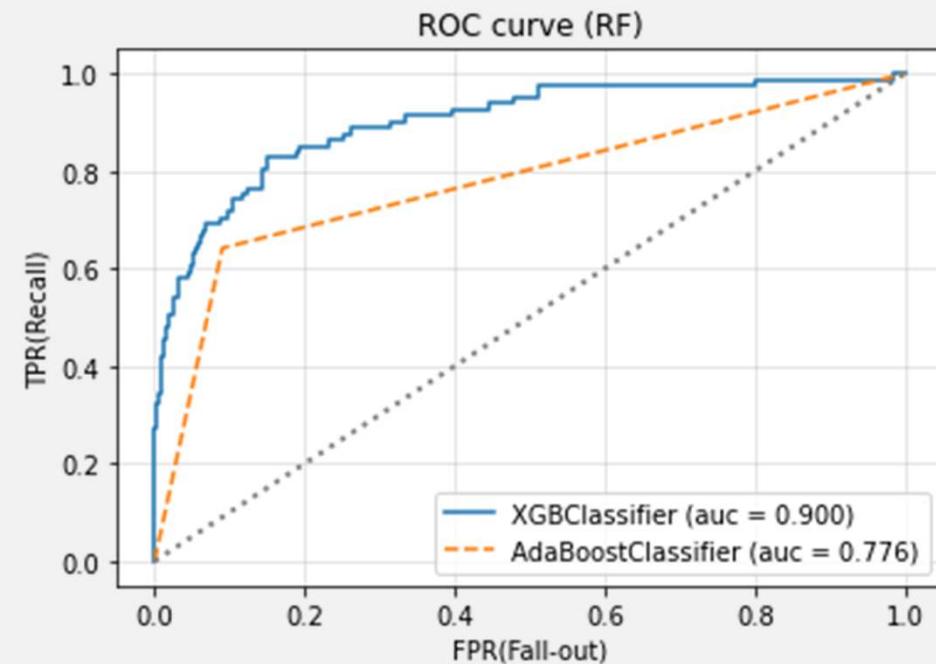
- 모델 구축
- 모델 검증
 - Confusion Matrix
 - Classification Report
 - cross_val_score
 - ROC curve

모델링

모델 검증

- XGBClassifier vs. AdaBoost

ROC AUC: 0.992 (+/- 0.015) [XGBClassifier]
ROC AUC: 0.925 (+/- 0.027) [AdaBoostClassifier]



```
Confusion Matrix :  
[[752 33]  
 [ 34 47]]  
Accuracy : 0.9226327944572749  
Precision : 0.5875  
Recall : 0.5802469135802469  
f1 score : 0.5838509316770186  
roc_auc score : 0.899646142958245  
  
-----  
Confusion Matrix :  
[[722 63]  
 [ 34 47]]  
Accuracy : 0.8879907621247113  
Precision : 0.42727272727272725  
Recall : 0.5802469135802469  
f1 score : 0.49214659685863876  
roc_auc score : 0.7499960682550916
```

- 모델 구축
- 모델 검증
 - Confusion Matrix
 - Classification Report
 - cross_val_score
 - ROC curve

모델링

모델 검증 – 학습 정확도 accuracy

LogisticRegression	0.8372
DecisionTree	0.8800
RandomForestClassifier	0.9122
XGBClassifier	0.9226
AdaBoost	0.8880

- RandomForest, XGBClassifier 모델
> 기준모델(학습 정확도 90.6%) 대비 더 나은 성능

- 모델 구축
- 모델 검증
 - Confusion Matrix
 - Classification Report
 - cross_val_score
 - ROC curve

모델링

- 모델 구축
- 모델 검증
 - Confusion Matrix
 - Classification Report
 - cross_val_score
 - ROC curve

모델 검증 - 종합

Model Name	Precision	Recall	F1 Score	ROC AUC Score
Logistic Regression	0.348	0.852	0.495	0.904
Decision Tree	0.411	0.654	0.505	0.779
RandomForest	0.525	0.642	0.578	0.917
XGBoost	0.588	0.58	0.584	0.9
AdaBoost	0.348	0.852	0.495	0.904

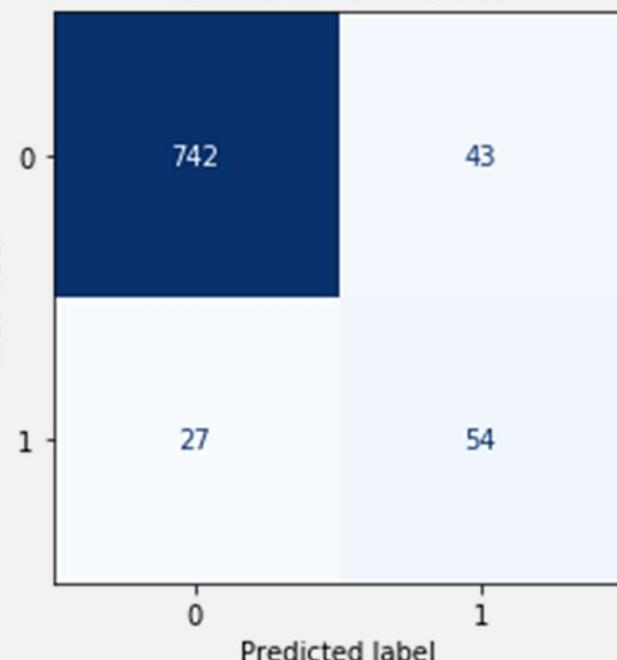
- 가장 높은 성능: RandomForestClassifier 모델

모델링

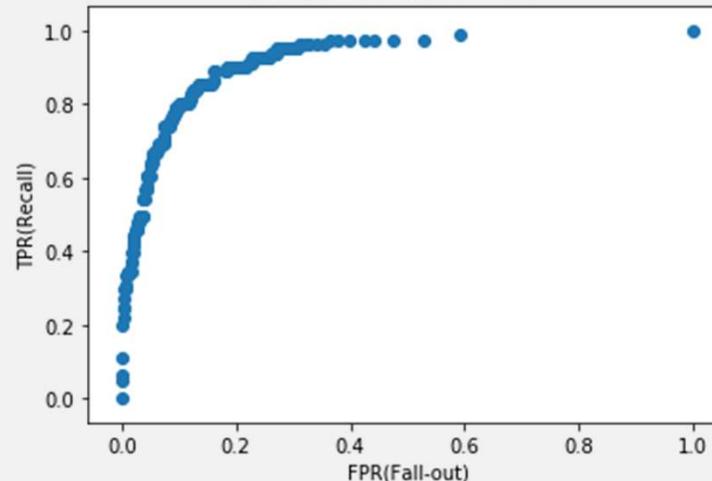
모델 성능 최적화

- 하이퍼파라미터 튜닝(RandomizedSearchCV)

Confusion Matrix



ROC curve (RandomForest)

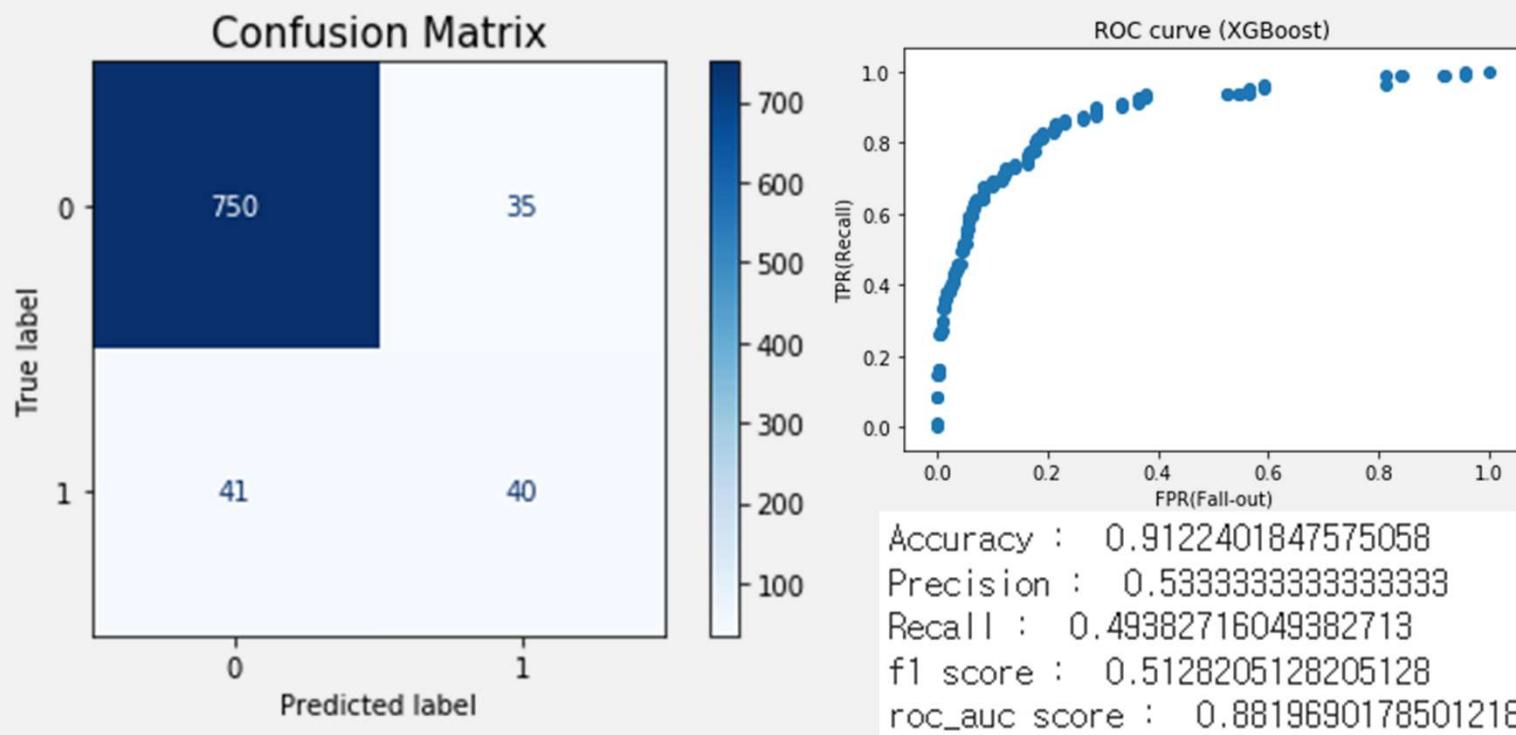


```
Accuracy : 0.9191685912240185
Precision : 0.5567010309278351
Recall : 0.6666666666666666
f1 score : 0.6067415730337078
roc_auc score : 0.9264606432334669
```

- 모델 구축
- 모델 검증
- 최적화
 - 하이퍼파라미터 튜닝 (RandomizedSearchCV)

모델링

모델 성능 최적화 (추가. XGBClassifier) - 하이퍼파라미터 튜닝(RandomizedSearchCV)



- 모델 구축
- 모델 검증
- 최적화
 - 하이퍼파라미터 튜닝 (RandomizedSearchCV)

모델 성능 최적화 - 결론

- 최고 성능: **RandomForestClassifier**
(모델 선택)
- 훈련, 검증 데이터셋에서 최적화된
하이퍼파라미터로 최종 모델을 재학습 refit

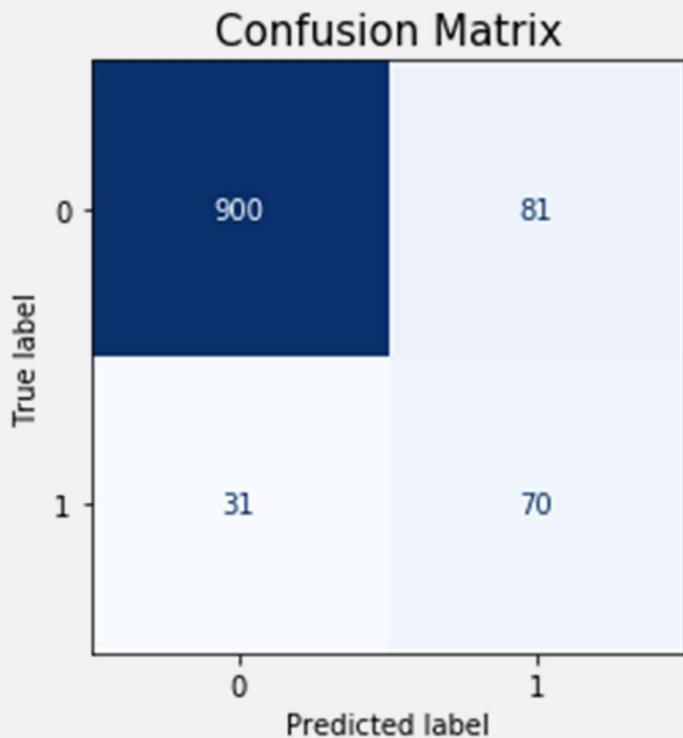
모델링

- 모델 구축
- 모델 검증
- 최적화
- 모델 선택

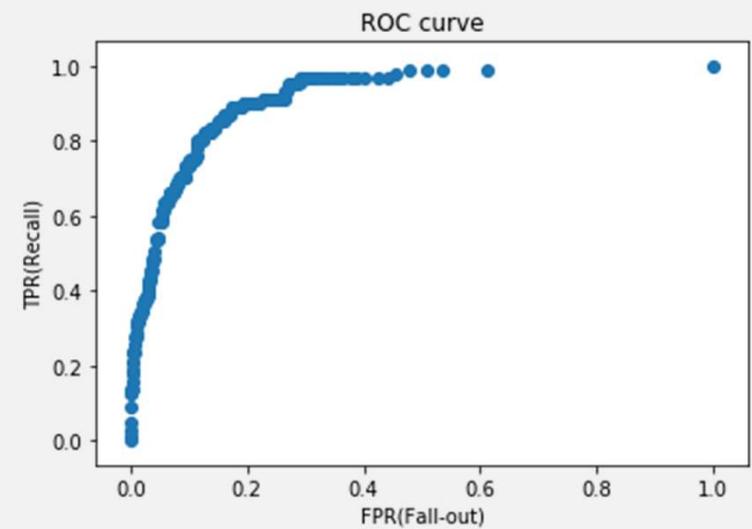
최종 모델

RandomForestClassifier

Confusion Matrix



ROC curve



Metrics

Accuracy : 0.8964879852125693
Precision : 0.46357615894039733
Recall : 0.693069306930693
f1 score : 0.5555555555555556
roc_auc score : 0.9209636559986273

최종 모델 - Permutation Importance

RandomForestClassifier

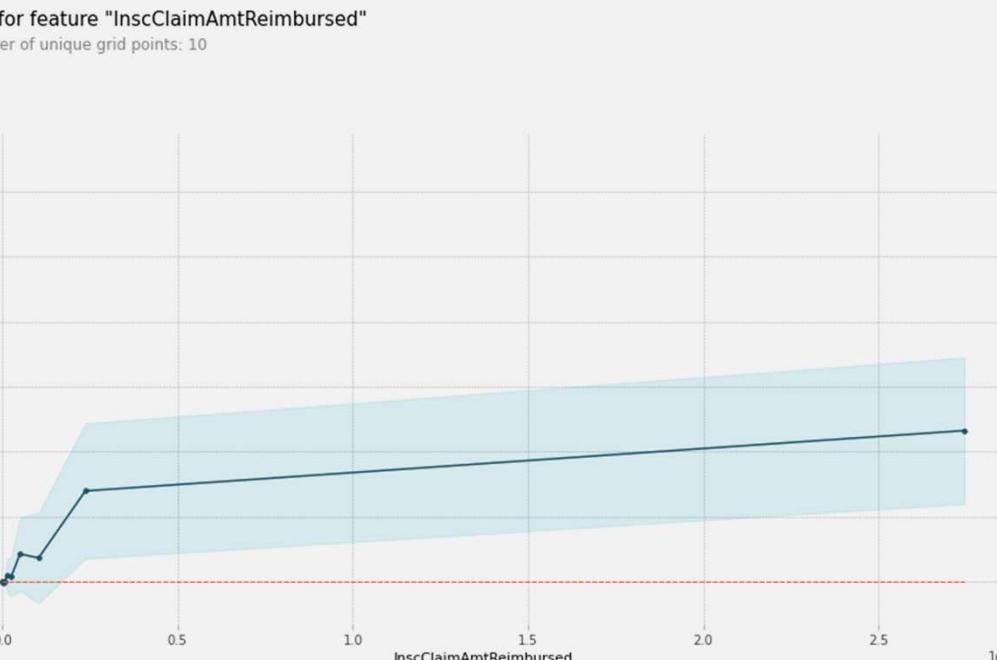
Weight	Feature
0.0615 ± 0.0042	Claim_Period
0.0605 ± 0.0102	Duration_IP
0.0488 ± 0.0048	InscClaimAmtReimbursed
0.0396 ± 0.0042	Inpatient_or_Outpatient
0.0241 ± 0.0134	DeductibleAmtPaid
0.0194 ± 0.0013	Race_2
0.0126 ± 0.0034	RenalDiseaseIndicator
0.0125 ± 0.0048	PerProvider_mean_Duration_IP
0.0121 ± 0.0014	PerProvider_mean_DeductibleAmtPaid
0.0121 ± 0.0030	PerProvider_mean_Claim_Period

Weight	Feature
-0.0002 ± 0.0047	IPAnnualReimbursementAmt
-0.0003 ± 0.0007	PerProvider_mean_IPAnnualDeductibleAmt
-0.0003 ± 0.0004	Claimed_extra
-0.0004 ± 0.0010	PerProvider_mean_OPAnnualReimbursementAmt
-0.0005 ± 0.0001	PerProvider_count_ClaimID
-0.0007 ± 0.0002	PerProvider_mean_Patient_Age
-0.0015 ± 0.0023	TotalAnnualDeductibleAmt
-0.0017 ± 0.0016	PerProvider_mean_OPAnnualDeductibleAmt
-0.0019 ± 0.0016	TotalReimbursementAmt
-0.0027 ± 0.0022	ChronicCond_Alzheimer

최종 모델 - PDP plot

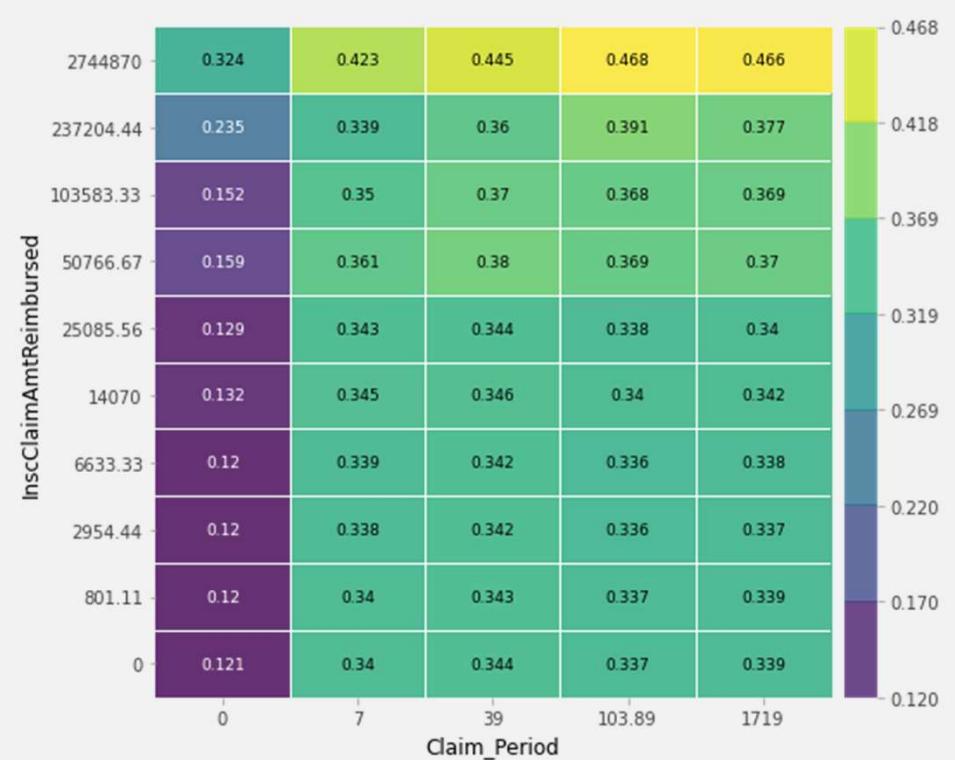
RandomForestClassifier

1 PDP



2 PDP

PDP interact for "Claim_Period" and "InscClaimAmtReimbursed"
Number of unique grid points: (Claim_Period: 5, InscClaimAmtReimbursed: 10)



최종 모델 - SHAP values

RandomForestClassifier

분류기 학습 결과1 (Test AUC)

```
# from sklearn.metrics import roc_auc_score
class_index = 1
y_pred_proba2 = model2.predict_proba(X_test)[:, class_index]
print(f'Test AUC for class (Oversampling) "{model2.classes_[clas
print(roc_auc_score(y_test, y_pred_proba2)) # 범위는 0~1, 수치는
```

Test AUC for class (Oversampling) "1":

0.9442779140299351

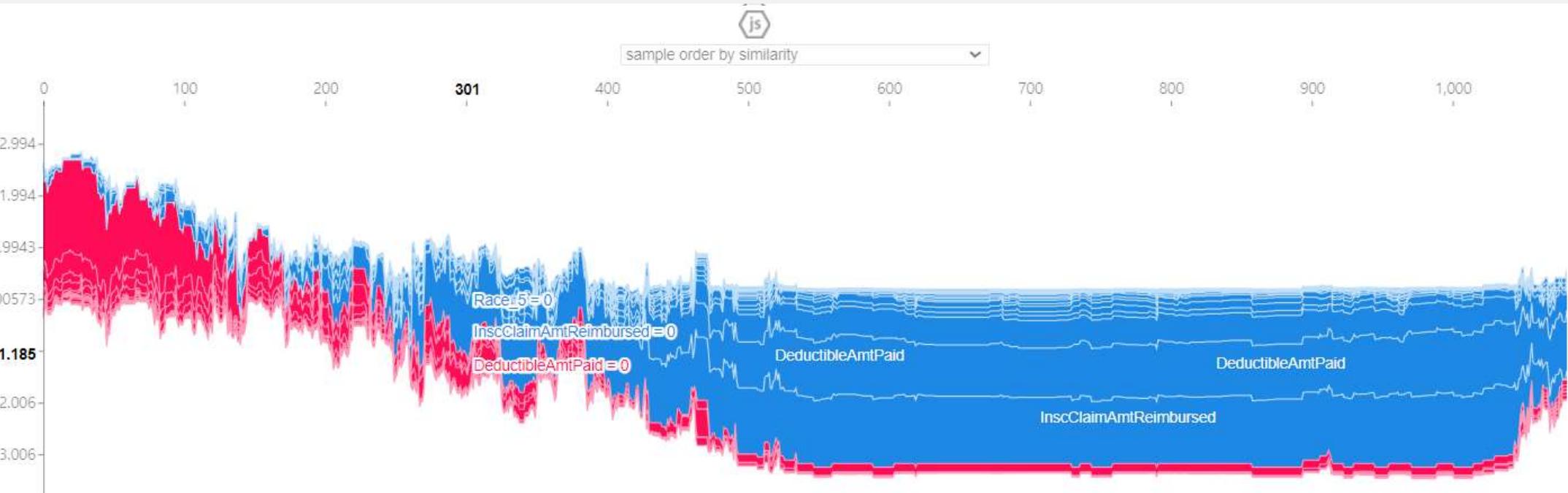
학습 결과2 (classification_report)

```
# Confusion Matrix 확인2
# from sklearn.metrics import classification_report
y_test_pred2 = model2.predict(X_test)
print(classification_report(y_test, y_test_pred2))
```

	precision	recall	f1-score	support
0	0.99	0.86	0.92	981
1	0.39	0.90	0.55	101
accuracy			0.86	1082
macro avg	0.69	0.88	0.73	1082
weighted avg	0.93	0.86	0.88	1082

최종 모델 - SHAP values

RandomForestClassifier



최종 모델 - SHAP values

RandomForestClassifier

구분

예시. 테스트셋 301번 데이터

fraud & right

not_fraud & right

not_fraud & wrong

fraud & wrong

```
# 테스트셋 301번 데이터 .shape: (1, 37)
```

```
row = X_test.iloc[[301]]
```

```
row
```

	InscClaimAmtReimbursed	DeductibleAmtPaid	Duration_IP	Claim_Period	Claimed_extra	Inpatient_or_Outpat
1093	280880	27190.0	134.0	134.0	0.0	

js

base value

0.2678

0.3762

0.4986

0.6211

0.7299

higher ↗ lower
f(x)
0.803167

0

Claim_Period = 134 | Duration_IP = 134 | DeductibleAmtPaid = 2.719e+4 | InscClaimAmtReimbursed = 2.809e+5 | Race_5 = 0

결론

보완사항
(개선할 점, 아쉬운 점)

보완이 필요하다고 생각되는 부분1

카디널리티

분석 데이터셋의 피처 전반에서 상당히 높은 카디널리티에 대한 고민

교차검증

k-fold/hold-out 등 다양한 교차검증 방법에 대한 이해

순열중요도 PDP, Shap

최종 모델을 설명하는 데 있어 다양한 수치적 지표의 해석방법에 대한 이해

문제 정의 가설 수립

-해결하고자 하는 문제를 보다 명확하고 구체적으로 정의하고 시작할 필요성
-관심 피처 설정, 분석 진행 방향 유념 필요

추가적인 특성 분석

기타 특성(진단코드, 진료코드 등)이 타겟에 미치는 영향에 대한 추가 분석

결론

보완사항
(아쉬운 점, 개선점 등)

보완이 필요하다고 생각되는 부분2

모델 선택

F1-score, ROC-AUC score 간 선택의 문제
(랜덤포레스트, XGBClassifier)

아쉬웠던 부분

도메인 지식

기본적인 도메인 지식 부족 > 인사이트 부족
(의료보험 생태계, 보험금 청구 체계 등)

데이터셋 활용도

-근본적으로 해결하려는 문제를 명확히 정의
하지 않아 적당한 데이터라도 충분히 활용하지
못한 느낌
-하나의 데이터를 다양한 관점에서 바라보는
연습 필요해 보임

감사합니다!

Codestates AIB 14
박성희



maeve.ep01@gmail.com

