머신러닝 분반

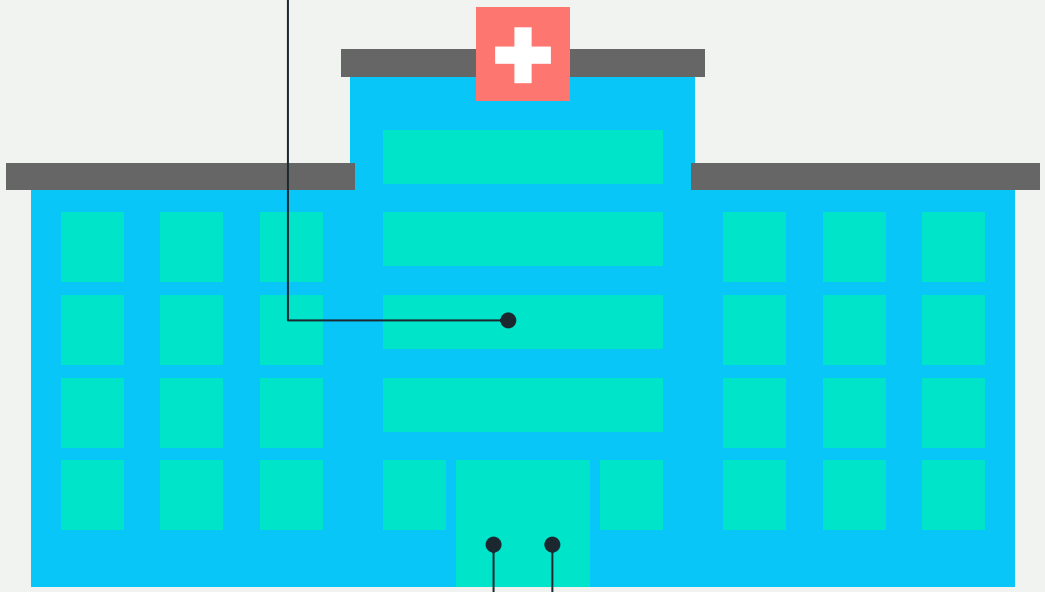
병원 개/폐업 분류 예측

14기 박상준 | 통계학과 2017250438
15기 김제성 | 산업경영공학부 2020170861
15기 신윤 | 통계학과 2020150450
15기 정영희 | 미디어학부 2020240032

Contents

02 **Data preprocessing**

파생변수 & PCA & ADASYN



01 **Raw data EDA**

손익계산서 & 재무제표

03 **Modeling**

AutoML & Stacking

Raw data EDA

라이브러리 설치 및 데이터 불러오기

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from imblearn.over_sampling import ADASYN
```

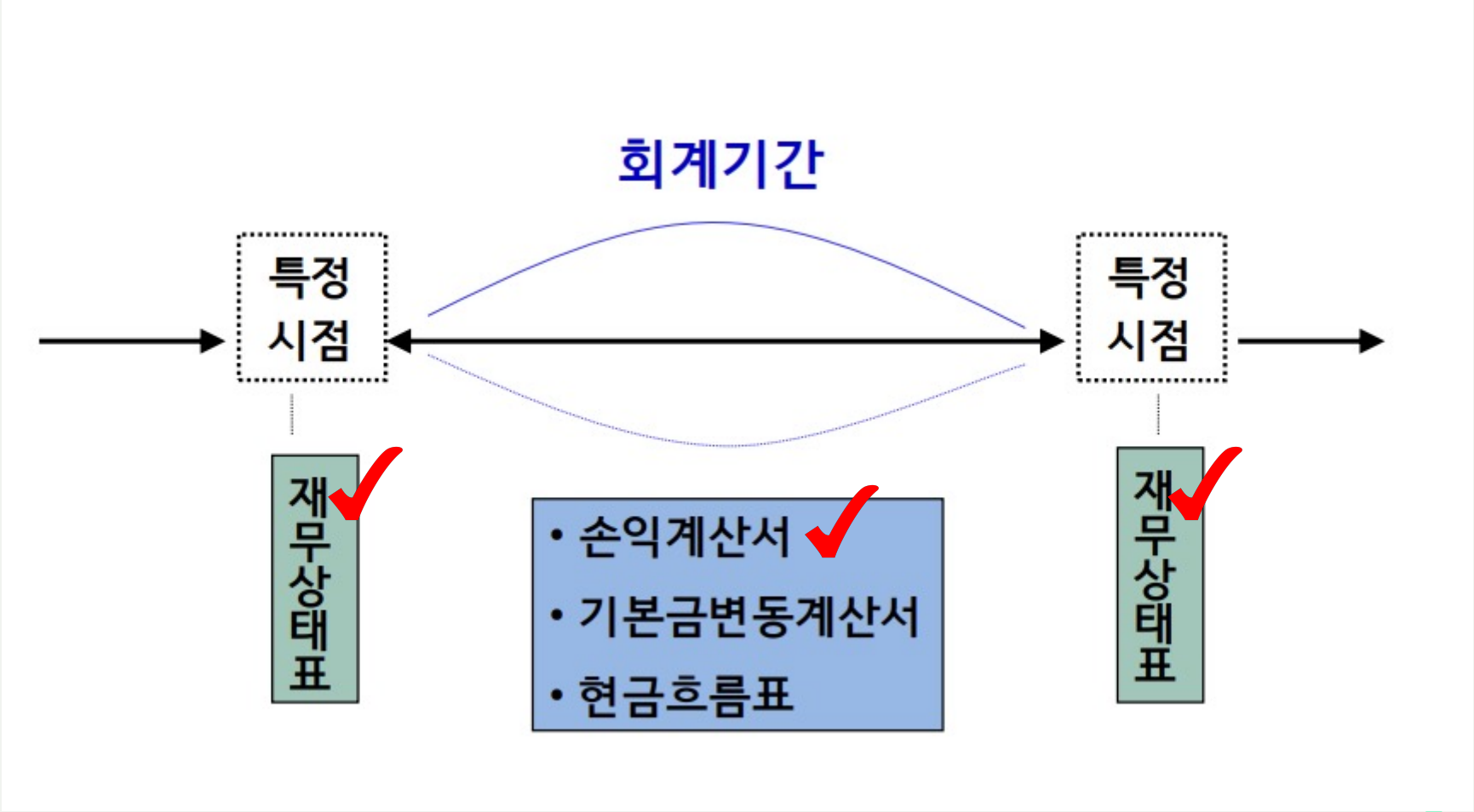
```
#model
from sklearn.model_selection import train_test_split, KFold
from sklearn.model_selection import StratifiedShuffleSplit, StratifiedKFold
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.model_selection import cross_val_score
import sklearn
from sklearn.metrics import accuracy_score
from sklearn.ensemble import ExtraTreesClassifier, RandomForestClassifier, VotingClassifier
from sklearn.model_selection import cross_val_score, cross_val_predict
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from sklearn.linear_model import LogisticRegression
from catboost import CatBoostClassifier

#import optuna
import optuna
from optuna.samplers import TPESampler
```

```
#import train file
train_df = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/2022-1 ML/9565_hospital_data/train.csv")
```

```
#import test file
test_df = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/2022-1 ML/9565_hospital_data/test.csv")
```

Raw data EDA



Raw data EDA

손익계산서

손익계산서
매출액
매출원가
매출총이익
판매비 및 일반관리비
영업이익
영업외수익
영업외비용
법인세차감전순이익
법인세
당기순이익

손익계산서
revenue
- salescost
매출총이익
- sga
영업이익
+ noi
- noe
법인세차감전순이익
- ctax
profit

Raw data EDA

손익계산서

손익계산서
revenue - salescost
매출총이익 - sga
영업이익 + noi - noe
법인세차감전순이익 - ctax
profit

판매비 및 일반관리비(sga)

+ 의료비용(salescost)

총의료비용
(new_salescost)

Raw data EDA

재무상태표

자산 변수 존재 X

- **유동자산** liquidAsset
  - 당좌자산 quickAsset
  - 재고자산 inventoryAsset
- **비유동자산** nonCAsset
  - 투자자산
  - 유형자산 tanAsset
  - 무형자산

부채 debt

- **유동부채** liquidLiabilities
- **비유동부채** NCLiabilities

자본 netAsset

- 자본금
- 이익잉여금 surplus

Raw data EDA

재무상태표

자산 변수 존재 X
- 유동자산 liquidAsset
  - 당좌자산 quickAsset
  - 재고자산 inventoryAsset
- 비유동자산 nonCAsset
  - 투자자산
  - 유형자산 tanAsset
  - 무형자산

부채 debt
- 유동부채 liquidLiabilities
- 비유동부채 NCLiabilities

자본 netAsset
- 자본금
- 이익잉여금 surplus

자산(totalAsset) = 유동자산 + 비유동자산

Raw data EDA

OC 변수 변환

```
X_num['OC'] = X_num['OC'].replace('open', 1)
X_num['OC'] = X_num['OC'].replace(' close', 0)
```

OC: 영업/폐업 분류, 2018년 폐업은 2017년 폐업으로 간주

open
1

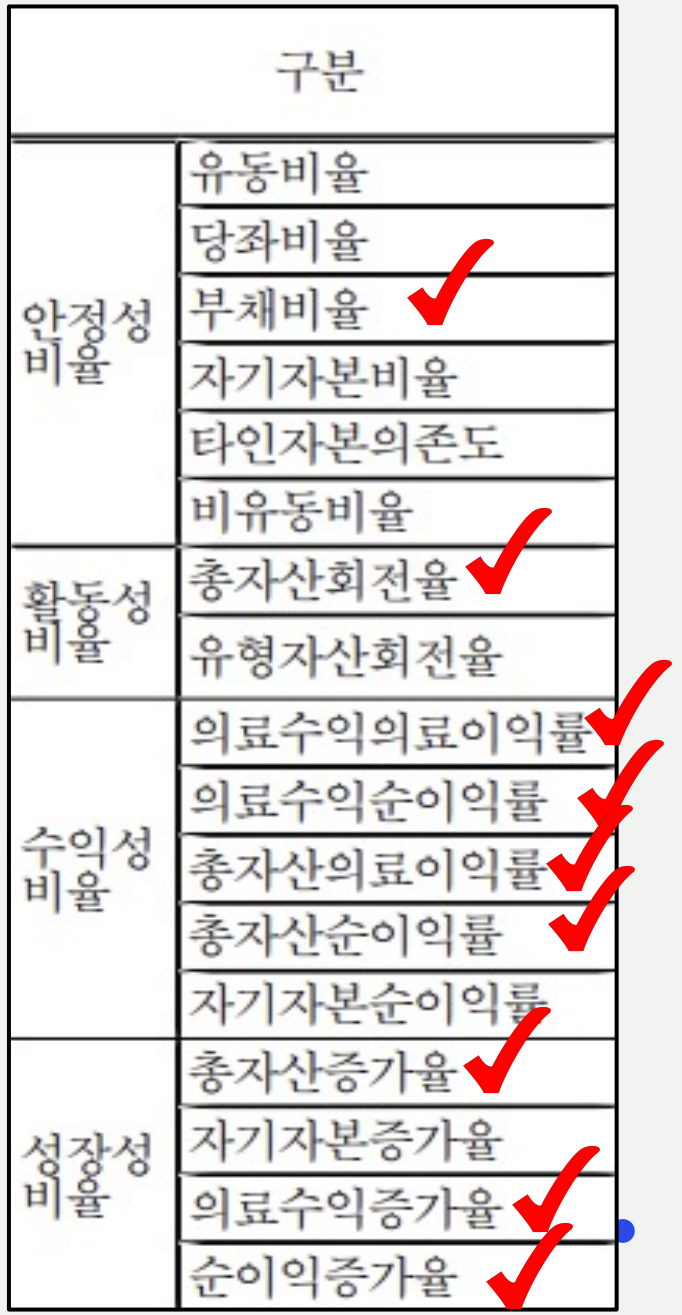
close
0

Data preprocessing

변수 추가: 연속형 변수

안정성, 활동성, 수익성, 성장성

4가지 측면의 다양한 재무비율 변수들을 생성해보면서
최적의 모델 구축



Data preprocessing

변수 추가: 연속형 변수

부채비율: 부채/자기자본 * 100

총자산회전율: 의료수익/총자산

의료수익의료이익률: (의료수익-의료비용)/의료수익 * 100

의료수익순이익률: 의료수익/의료이익 * 100

총자산의료이익률: (의료수익-의료비용) / 총자산 * 100

총자산의료수익률: 의료수익/ 총자산 * 100

총자산증가율: (당기총자산-전기총자산)/전기총자산 * 100

의료수익증가율: (당기의료수익-전기의료수익)/전기의료수익 * 100

순이익증가율: (당기순이익-전기순이익)/전기순이익 * 100

Data preprocessing

변수 추가: 범주형 변수

지역변수 도입

남부	중부	광역시
경상북도, 경상남도, 전라북도, 전라남도, 제주도	충청북도, 충청남도, 경기도, 강원도	서울시, 인천시, 대전시, 대구시, 부산시, 울산시, 세종시, 광주시

Data preprocessing

변수 값 조정

재무상태표 식 일치시키기

부채(debt) = 유동부채(liquidLiabilities) + 고정부채 (NCLiabilities)

총자산(totalAsset) = 유동자산(liquidAsset) + 비유동자산(nonCAsset)

자기자본(netAsset) = 총자산(totalAsset) - 부채(debt)

Data preprocessing

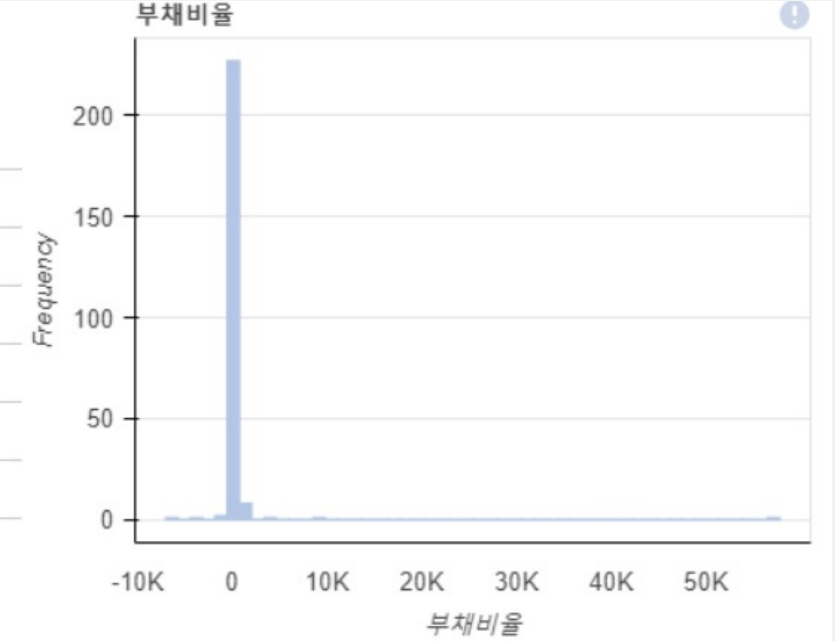
행 삭제 및 결측치 변환

Train set의 행에 결측치가 하나라도 있으면 삭제

새로 생성한 연속형 변수에 이상치가 있을 경우 행 삭제

손익계산서/재무상태표의 변수들이 다 0일 때:

- 결측값으로 변환하고 삭제하지 않음
- 값을 채우기 위하여 Iterative Imputer 진행



<이상치가 있었던 부채비율 변수의 히스토그램>

Data preprocessing

최종 선택 변수 ✓

새로 생성한
연속형 변수

+

지역 범주화
변수

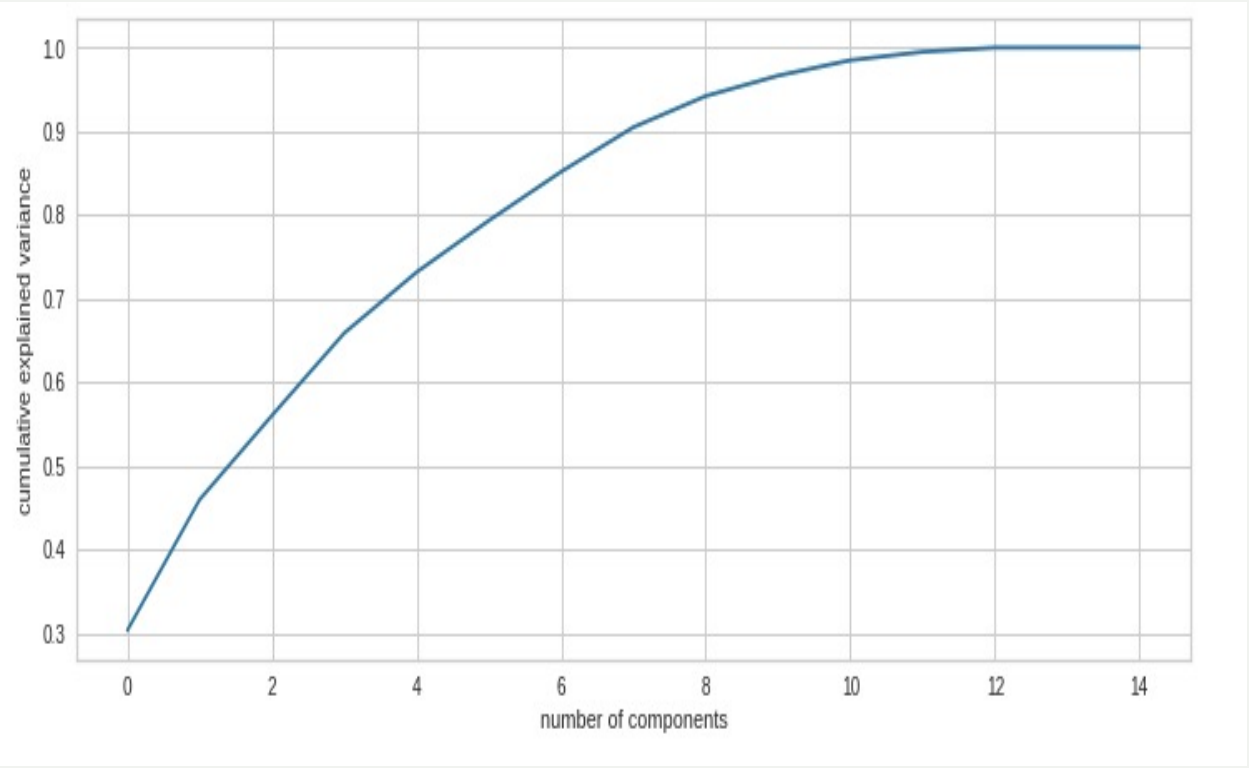
+

2017년의
의료수익
당기순이익
총자본
부채
자기자본

Data preprocessing

PCA

연속형 변수들에 대하여 PCA 진행 → 총 7개의 주성분 선택

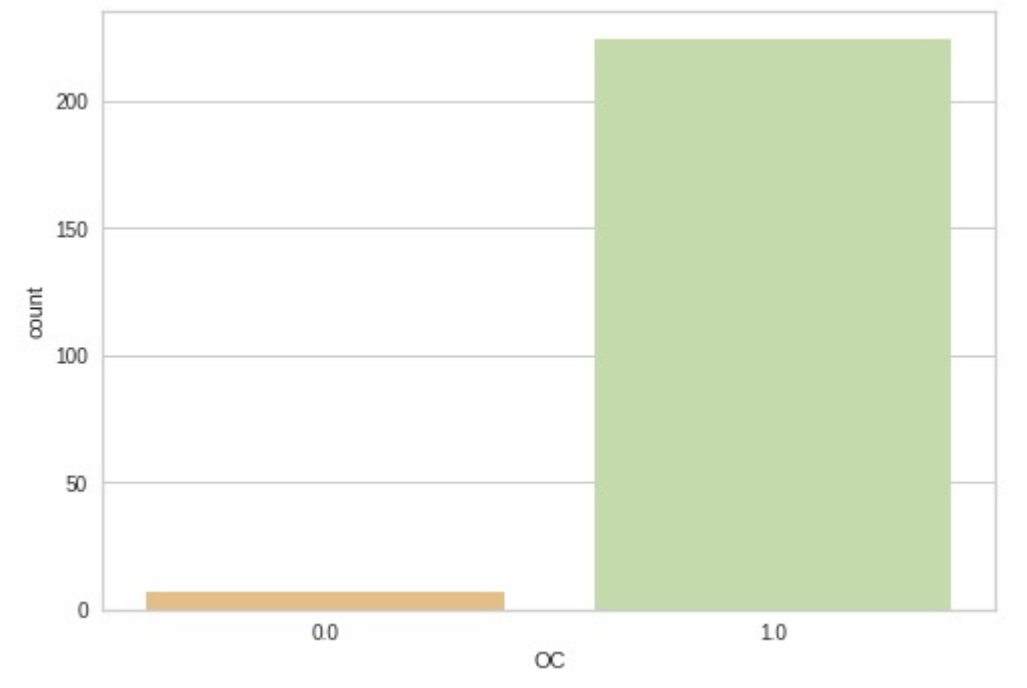


	설명가능한 분산 비율	기여율	누적기여율
pca1	4.563353e+00	3.033738e-01	0.303374
pca2	2.351461e+00	1.563262e-01	0.459700
pca3	1.511853e+00	1.005087e-01	0.560209
pca4	1.482646e+00	9.856699e-02	0.658776
pca5	1.094021e+00	7.273102e-02	0.731507
pca6	9.289376e-01	6.175618e-02	0.793263
pca7	8.789552e-01	5.843334e-02	0.851696
pca8	8.000715e-01	5.318911e-02	0.904885
pca9	5.602363e-01	3.724476e-02	0.942130
pca10	3.663000e-01	2.435179e-02	0.966482
pca11	2.773468e-01	1.843814e-02	0.984920
pca12	1.514136e-01	1.006604e-02	0.994986
pca13	7.502153e-02	4.987465e-03	0.999973
pca14	3.989785e-04	2.652427e-05	1.000000
pca15	2.031653e-29	1.350652e-30	1.000000

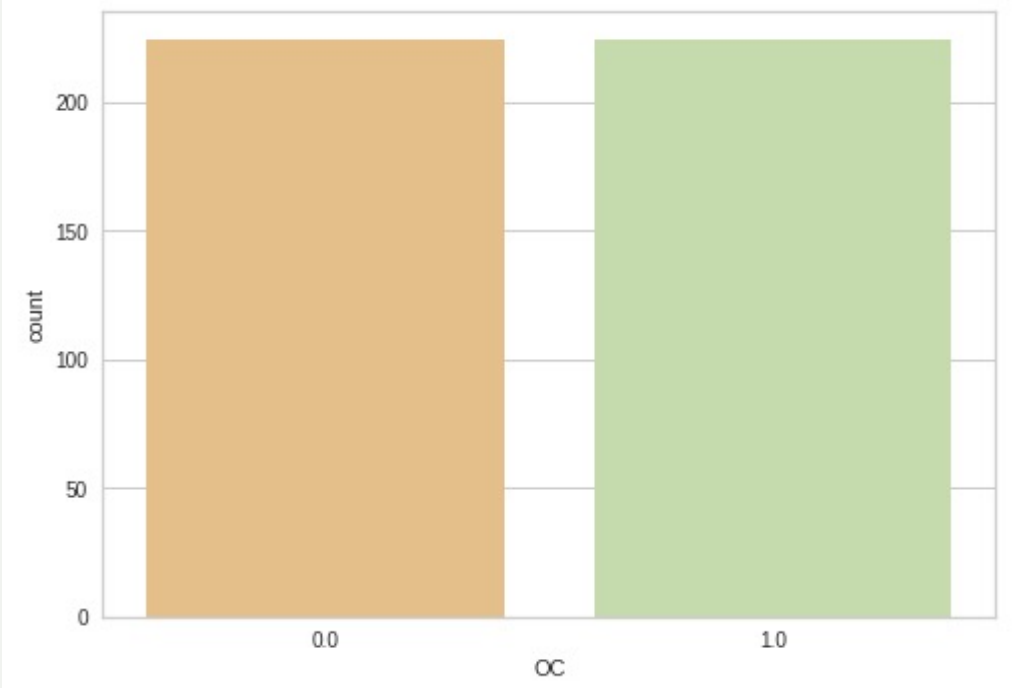
Data preprocessing

ADASYN

데이터 불균형 해결



ADASYN 전



ADASYN 후

Modeling

AutoML

| PyCaret을 통한 모델 선택


```
[ ]  # pycaret에 없는 모델 설치
     !pip install xgboost
     !pip install --upgrade xgboost
     !pip install catboost

[ ]  !pip install pycaret[full]

[ ]  from pycaret.utils import enable_colab
     enable_colab()

[ ]  !pip install pycaret[full]

[ ]  # import pycaret
     import pycaret

     from pycaret.classification import *
```

Modeling

AutoML

| PyCaret을 통한 모델 선택

```
model = setup(
    data = X_train,
    target = "OC",
    fold = 5
)
```

	Description	Value
0	session_id	3633
1	Target	OC
2	Target Type	Binary
3	Label Encoded	0.0: 0, 1.0: 1
4	Original Data	(448, 11)
5	Missing Values	False
6	Numeric Features	10
7	Categorical Features	0
8	Ordinal Features	False
9	High Cardinality Features	False
10	High Cardinality Method	None
11	Transformed Train Set	(313, 10)

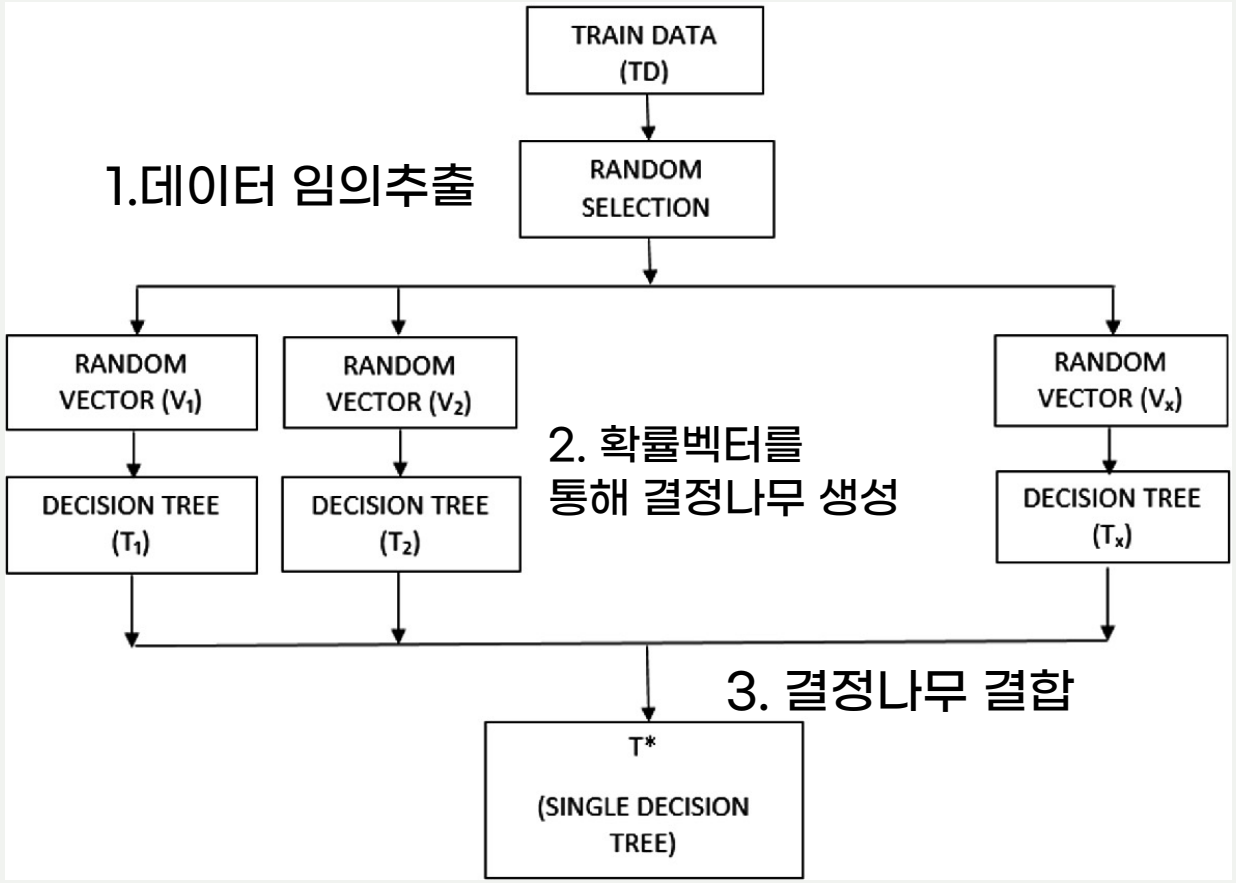
```
[ ]  top4_model = compare_models(
                        round=4,
                        sort="Accuracy",
                        n_select = 4 )
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	0.9681	0.9720	0.9625	0.9752	0.9683	0.9362	0.9372	0.538
gbc	Gradient Boosting Classifier	0.9554	0.9890	0.9498	0.9640	0.9562	0.9107	0.9120	0.100
lightgbm	Light Gradient Boosting Machine	0.9521	0.9916	0.9498	0.9575	0.9529	0.9042	0.9056	0.144
rf	Random Forest Classifier	0.9490	0.9887	0.9435	0.9573	0.9498	0.8979	0.8992	0.546
catboost	CatBoost Classifier	0.9489	0.9910	0.9435	0.9575	0.9495	0.8978	0.8996	2.944
ada	Ada Boost Classifier	0.9458	0.9737	0.9058	0.9869	0.9441	0.8917	0.8955	0.102
xgboost	Extreme Gradient Boosting	0.9456	0.9893	0.9433	0.9497	0.9462	0.8912	0.8919	1.074
dt	Decision Tree Classifier	0.9329	0.9333	0.9121	0.9542	0.9324	0.8658	0.8672	0.016
svm	SVM - Linear Kernel	0.9074	0.0000	0.8871	0.9316	0.9069	0.8149	0.8189	0.014
knn	K Neighbors Classifier	0.9041	0.9635	0.8429	0.9645	0.8991	0.8087	0.8161	0.134
lr	Logistic Regression	0.9040	0.9719	0.8746	0.9371	0.9010	0.8086	0.8153	0.840

Modeling

모델 설명

Extra Trees Classifier



주요 하이퍼파라미터

n_estimators
: 전체 트리 개수

max_depth
: 트리의 최대 깊이

min_samples_split
: 내부 노드를 분할하는데 필요한 최소 샘플 수

Modeling

최종 모델

Stack_models()

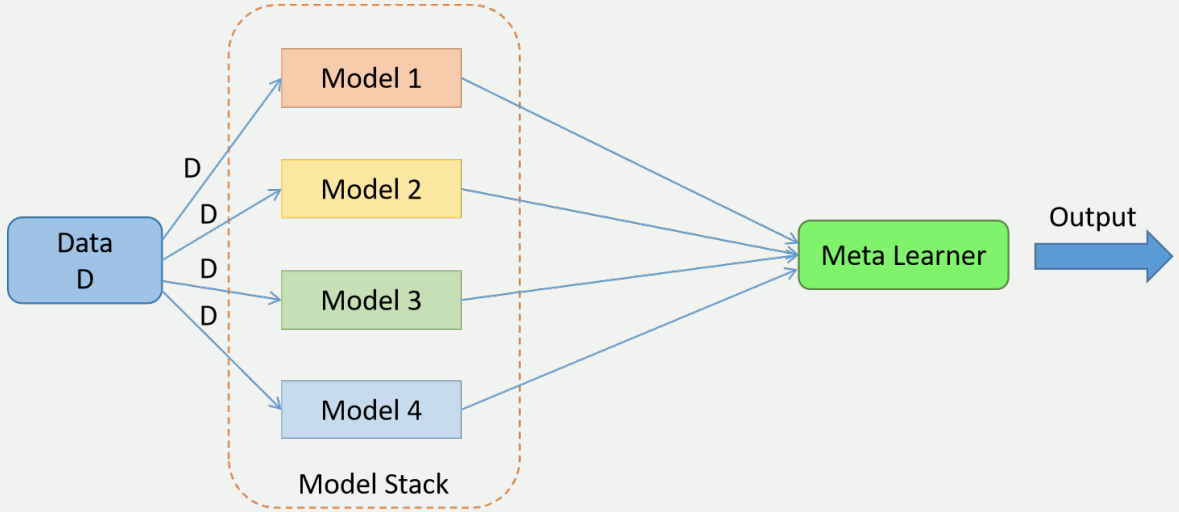
```
[70] blend4_stack = stack_models(estimator_list=top4_model)
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9683	0.9667	0.9355	1.0000	0.9667	0.9364	0.9383
1	0.9524	0.9808	0.9688	0.9394	0.9538	0.9047	0.9051
2	0.9683	0.9929	0.9688	0.9688	0.9688	0.9365	0.9365
3	0.9677	1.0000	0.9355	1.0000	0.9667	0.9355	0.9374
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Mean	0.9713	0.9881	0.9617	0.9816	0.9712	0.9426	0.9435
SD	0.0156	0.0128	0.0243	0.0243	0.0153	0.0312	0.0309

```
[71] final_model = finalize_model(blend4_stack)
```

```
[72] prediction = predict_model(final_model, data = X_test)
```

Stacking 기법으로 새로운 모델 구축



Conclusion

| 제출결과

131	qsongeel	qs	0.90476	20	9시간 전

데이콘 최종 제출 결과
0.90476의 점수로 131위

Conclusion

| 프로젝트 소감 및 의의

- 새로운 변수를 생성하고 의미 없는 변수를 골라내는 과정에서
  데이터에 대한 도메인 지식이 매우 중요함을 깨달음

- 결측치를 제거하고 대체하는 기준에 대해서 많은 고민을 해보았고
  또 이를 수행하는 과정도 매우 다양한 방법이 존재함을 느낌

- AutoML을 사용해봄으로써 모델 선택 과정에 새로운 접근 방식을 접해볼 수 있었음