스마트 공장 제품 품질 상태 분류 AI 온라인 해커톤

Binary는 호남선

dsjoh, Dolphin, dafas, YooooNn, 쿨병ENTP

INDEX

PART 1

개발 환경

- Google Colab -

Linux-5.10.147+-x86_64-with-glibc2.29

Ubuntu 20.04.5 LTS

런타임 유형 : CPU

Python 3.8.10

Numpy 1.22.4

Pandas 1.3.5

Sklearn 1.2.1

CatBoost 1.1.1

CatBoost

PART 2

데이터

2

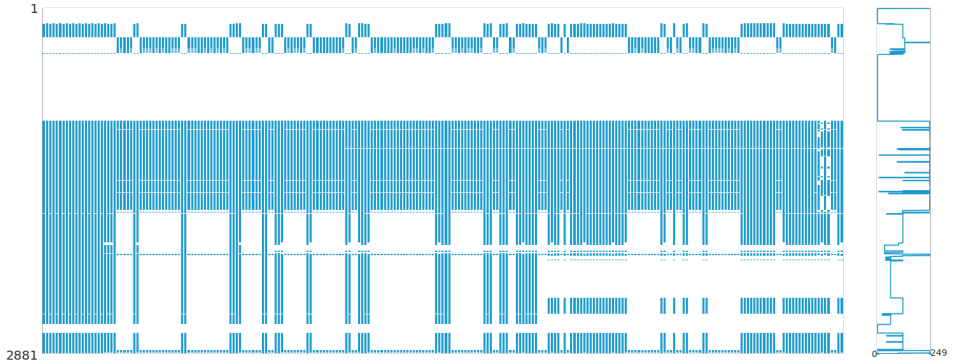
002 데이터

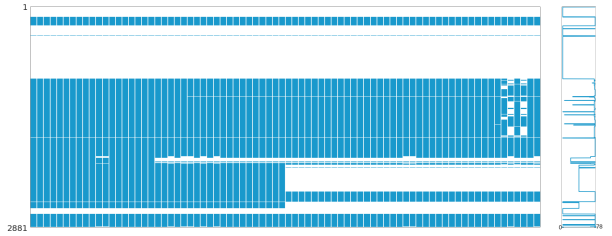
Train[X_87~X_117]

```
[409] train_0.iloc[:,92:128]
```

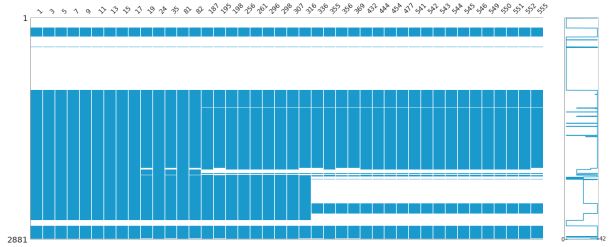
	X_87	X_88	X_89	X_90	X_91	X_92	X_93	X_94	X_95	X_96	X_97	X_98	X_99	X_100	X_101	X_102	X_103	X_104	X_105	X_106	X_107	X_108	X_109	X_110	X_111	X_112	X_113	X_114	X_115	X_116	X_117
569	130.0	130.0	130.0	35.0	999.0	28.0	NaN	NaN	NaN	2.0	0.00130	0.00130	0.00130	18.0	0.000039	0.000048	0.000029	NaN	0.000005	0.000007	0.000003	32.0	0.000003	0.000005	0.000002	40.0	0.19	0.19	0.19	44.0	0.000008
570	129.0	129.0	130.0	36.0	999.0	NaN	0.19	0.19	0.19	2.0	0.00077	0.00081	0.00072	18.0	0.000071	0.000083	0.000051	28.0	0.000009	0.000010	0.000003	32.0	0.000004	0.000005	0.000003	40.0	NaN	NaN	NaN	44.0	0.000034
571	130.0	130.0	130.0	35.0	999.0	28.0	NaN	NaN	NaN	2.0	0.00130	0.00130	0.00120	18.0	0.000042	0.000050	0.000029	NaN	0.000007	0.000012	0.000002	32.0	0.000004	0.000008	0.000001	40.0	0.19	0.20	0.19	44.0	0.000013
572	129.0	129.0	130.0	35.0	999.0	NaN	0.19	0.19	0.19	2.0	0.00077	0.00082	0.00072	18.0	0.000074	0.000088	0.000050	28.0	0.000009	0.000011	0.000003	32.0	0.000003	0.000005	0.000003	40.0	NaN	NaN	NaN	44.0	0.000034
596	130.0	130.0	130.0	35.0	999.0	28.0	NaN	NaN	NaN	2.0	0.00130	0.00150	0.00130	18.0	0.000057	0.000070	0.000048	NaN	0.000015	0.000046	0.000005	32.0	0.000003	0.000006	0.000001	40.0	0.19	0.20	0.19	44.0	0.000011
597	129.0	129.0	130.0	38.0	999.0	NaN	0.19	0.19	0.19	2.0	0.00077	0.00084	0.00072	18.0	0.000063	0.000078	0.000047	28.0	0.000007	0.000011	0.000003	32.0	0.000004	0.000005	0.000003	40.0	NaN	NaN	NaN	44.0	0.000034

A_31의 결측치





LINE = 'T050304'

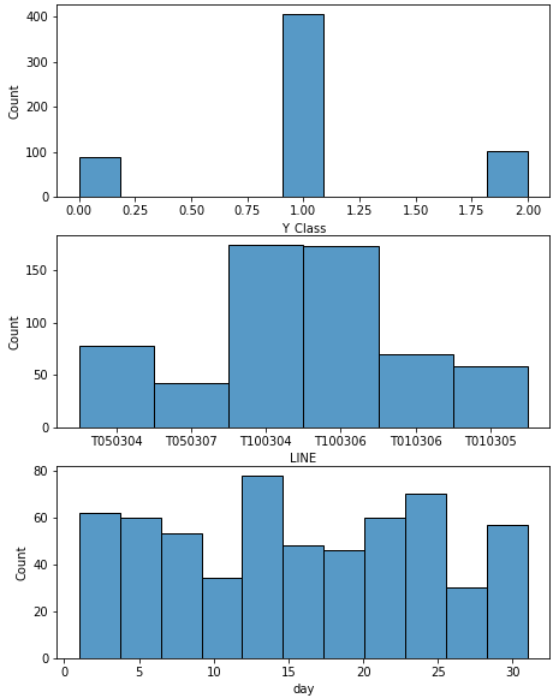


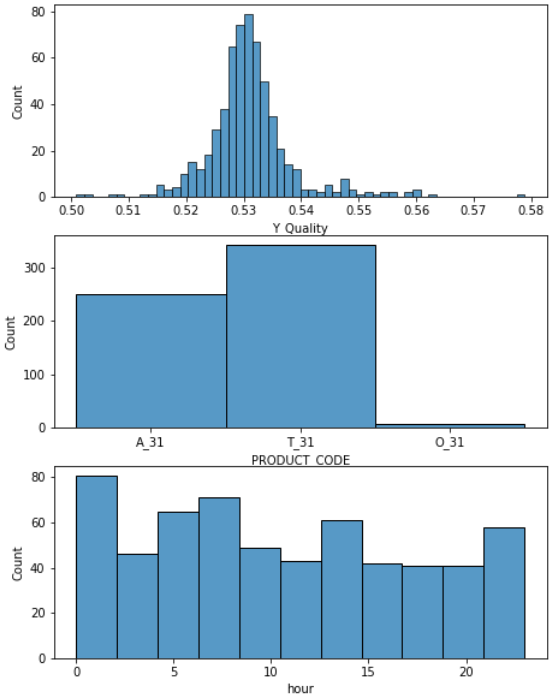
LINE = 'T050307'

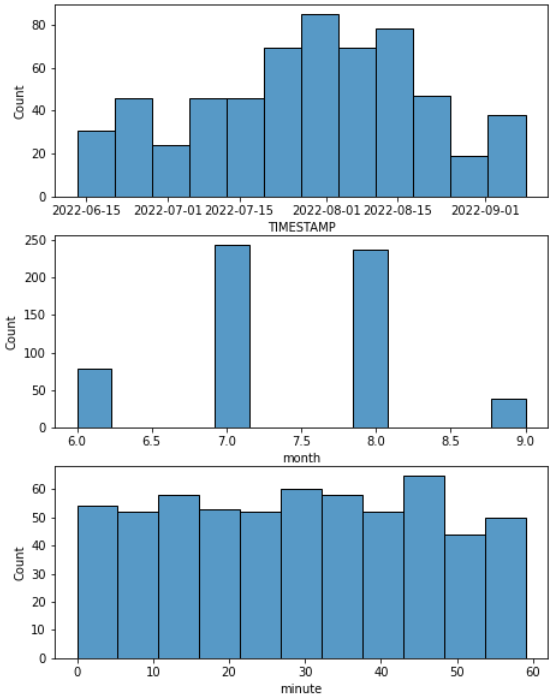
→ 결측치가 존재하는 일부 행을 비교하면, 생산 라인, 생산 제품, 생산 월에 따라 종속되는 변수들이 달라지는 것을 알 수 있으며, 그 값이 일부 결측치로 들어가 있는 경우도 다량 존재한다.(X_97, X_92~X_95)

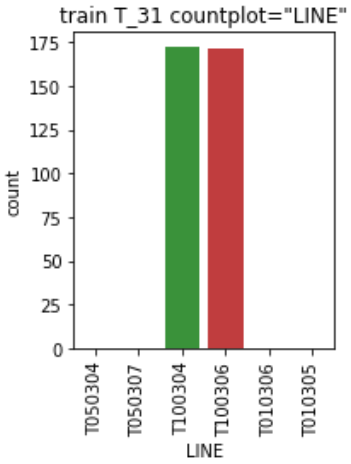
2

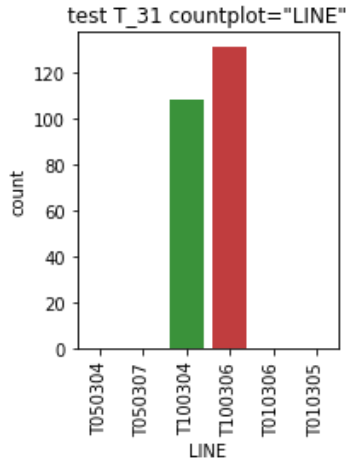
002 데이터

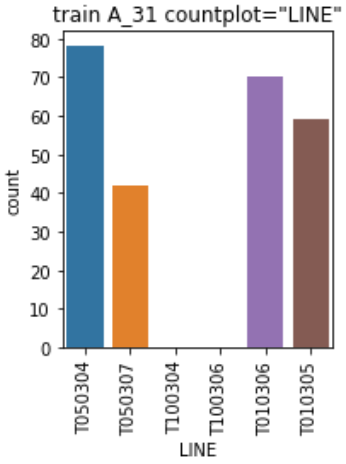


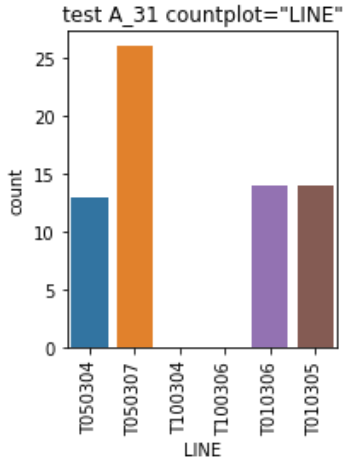




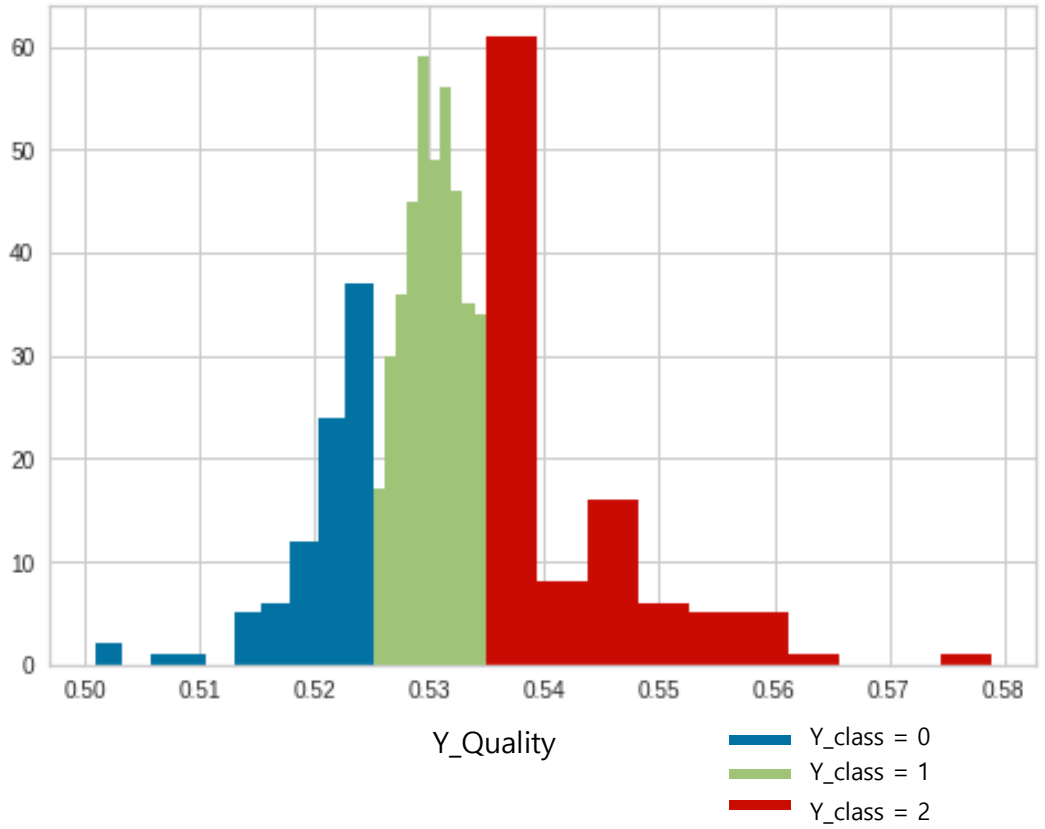


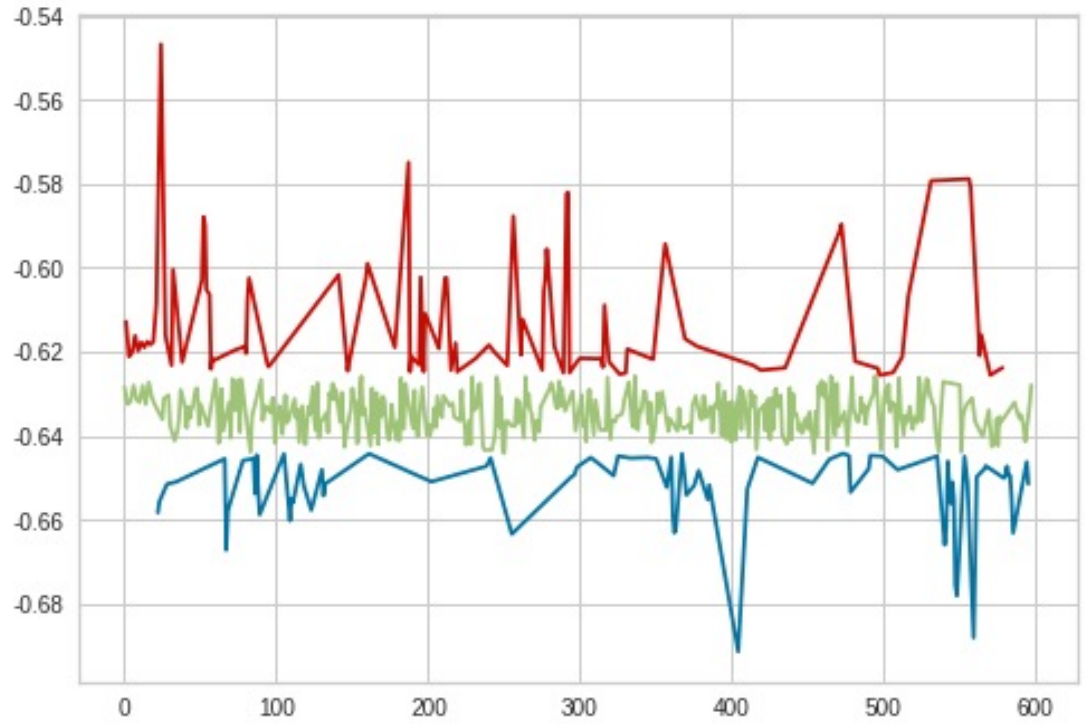






→ Categorical_feature에 해당하는 LINE과 PRODUCT_CODE 간에는 사전에 정해진 조합이 존재하는 것으로 보이며, Y_Class에서는 Class 불균형이 관찰된다.





→ Y_Quality의 경우는 Y_Class를 결정짓기 위한 정량적 수치이므로 학습에 도움이 되지 않을 것이라고 판단
(이후 Y_Quality를 기준으로 회귀값을 예측한 다음 얻은 결과를 이용하여 Y_Class를 따로 분류하는 시도를 하였으나, 유의미한 점수 향상은 나타나지 않았다.)

PART 3

솔루션

TimeStamp 전처리

```
# TIMESTAMP 전처리
train2['TIMESTAMP'] = pd.to_datetime(train2['TIMESTAMP'])
train2['month'] = train2['TIMESTAMP'].dt.month # 월
train2['day'] = train2['TIMESTAMP'].dt.day # 일
train2['hour'] = train2['TIMESTAMP'].dt.hour # 시간
train2['minute'] = train2['TIMESTAMP'].dt.minute # 분

test2['TIMESTAMP'] = pd.to_datetime(test2['TIMESTAMP'])
test2['month'] = test2['TIMESTAMP'].dt.month # 월
test2['day'] = test2['TIMESTAMP'].dt.day # 일
test2['hour'] = test2['TIMESTAMP'].dt.hour # 시간
test2['minute'] = test2['TIMESTAMP'].dt.minute # 분
```

앞서 생산 시점에 따라서도 X_feature들의 조합이 달라지는 것을 확인

a) 연, 월, 일, 시의 정보들이 도움이 될 것이라 판단하여 TIMESTAMP column을 전처리 진행

b) Train 데이터는 모두 2022년의 데이터이므로 year column은 별도 전처리 진행하지 않음

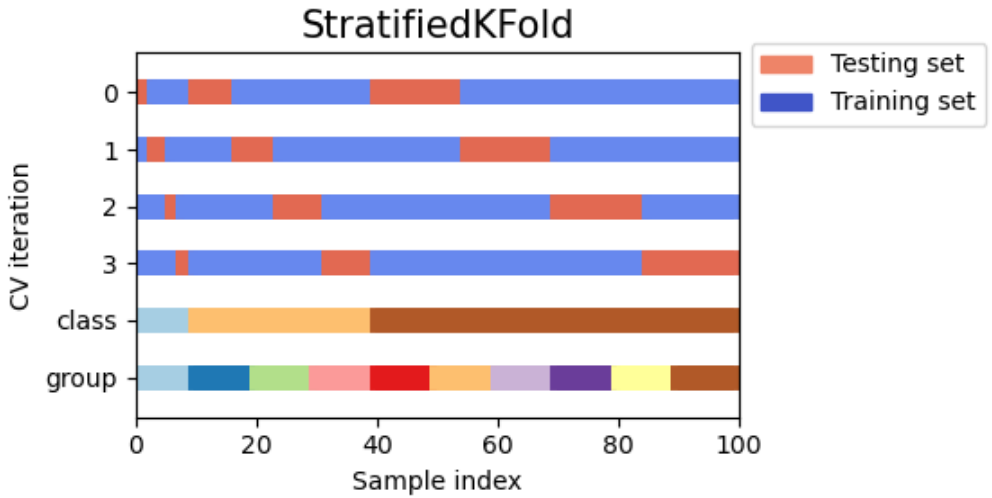
CatBoost

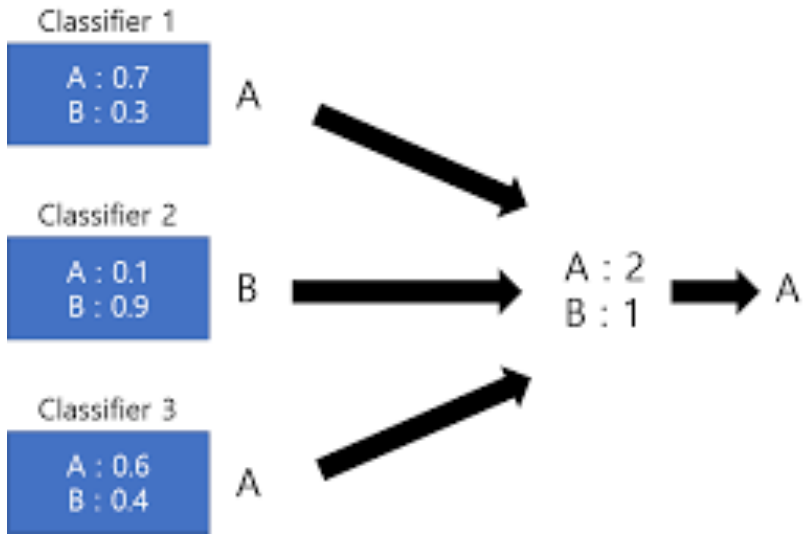
- X_feaure와의 종속성이 가장 두드러지게 나타나는 'LINE'과 'PRODUCT_CODE'를 'cat_features' 파라미터로 지정하여 학습할 수 있고, 상대적으로 하이퍼파라미터 튜닝의 영향을 적게 받는 CatBoostClassifier를 메인 model로 사용

- CatBoost 모델은 결측치를 스스로 처리하기 때문에 실제 결측치와 처음부터 값이 존재하지 않는 값들의 imputation 고민 없이 학습을 진행할 수 있다.

- 다양한 imputation을 추가적용해 보았으나, 성능 향상은 나타나지 않음
  e.g. mice, zero, linear

Stratified K-Fold & Soft-Voting

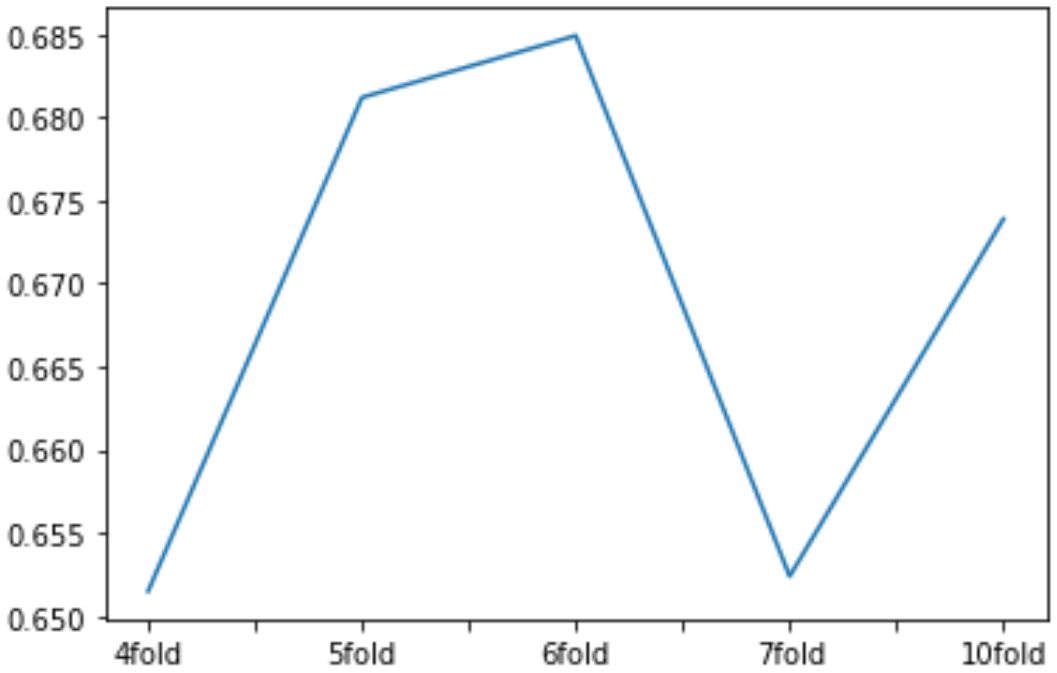




- Class 불균형으로 인한 문제를 사전에 방지하기 위해 Stratified K-Fold 기법을 사용하여 fold를 나누고, 각 fold별로 학습된 모델들을 Soft-Voting하여 보다 robust 한 예측 결과를 얻는다.

```
or train_index,valid_index in skf.split(train2,label_df):
  fold = fold + 1
  x_train=train2.iloc[train_index]
  x_valid=train2.iloc[valid_index]
  y_train=label_df.iloc[train_index]
  y_valid=label_df.iloc[valid_index]
  cbc.fit(x_train, y_train, eval_set=(x_valid,y_valid), early_stopping
  result+=cbc.predict_proba(test2)/PAR['FOLD'] # Soft-Voting
  cbc.save_model(PAR['OUTPUT_PATH']+'{}_fold_model.cbm'.format(fold))
```

Stratified 6 Fold



- 모델이 전반적으로 T_31 Product의 데이터에서 정답률이 낮게 나타남
- O_31 Product의 경우 데이터가 6개만 존재
- 5Fold_Cross_val_Score에서 Fold별 Score의 편차가 크게 나타난다는 점을 통해 분할된 데이터의 분포가 모델 성능에 큰 영향을 줄 수 있음을 확인

fold별 cross_val_score의 평균값을 기준으로
최적의 Fold 수를 찾는다.

PART 4

결과

< Public 29위, Private 16위 >

Public Score : 0.7254666855

Private Score : 0.67455

END

Thank you