LG Aimers
스마트 공장 제품 품질 상태 분류
AI 온라인 해커톤

Team hyena
박혜원 이준혁

Data

- Train set: 598, Test set: 243
- Categorical feature (2) : Line, Product code

LINE	PRODUCT	Count
T010305	A_31	59
T010306	A_31	70
T050304	A_31	78
T050307	A_31	42
T100304	O_31	3
	T_31	172
T100306	O_31	3
	T_31	171

- Numeric feature (2875): X1~X2875

- Y_Class: Imbalanced multiclass

Y_Class	Count
0	88
1	407
2	103

Missing Value

- Train set 에 대해 각 변수 별 평균 407개, 총 1,172,834 개의 결측치 존재

- 결측치 대체: 휴리스틱한 실험을 진행
  - Mean 값
  - Median 값
  - Lgbm 모델을 이용한 imputation 값
  - 0
  - -1 (결측치 임을 나타내기 위함)
  - Simple Linear Regressor를 이용한 imputation 값

- 위 처럼 결측치에 대해서 여러가지 시도를 해보았으나 결측치를 0으로 채웠을 때 가장 성능이 좋았다.

Pycaret

- AutoML
  - 데이터 특징 추출 및 하이퍼 파라미터 최적화 등의 주요 프로세스를 자동화 하는데 이용되는 머신러닝 기술
  - 비식별화된 변수 X1~X2875에 대해 다양한 Feature 변형을 시도하고, 여러 모델에 같은 조건으로 실험을 해보고자 AutoML을 적용하게 됨

- 수 많은 AutoML 라이브러리들 중 Pycaret을 사용한 이유?
  - 사용이 쉽고 간편함
  - 전처리 단계에서 결측치 처리, 정규화 등의 자동화
  - Data의 Stratify Train/Test Split 이 가능함
  - GPU를 사용한 연산이 가능
  - 성능평가를 원했던 ML 모델들을 모두 포함하고 있음
  - Model Selection 및 Model Blending이 용이함

Pycaret

- Setup
  - Numeric features: 2875 (X1~X2875)
  - Categorical features: 2 (LINE, PRODUCT_CODE)
    - One-hot Encoding
  - 데이터 정규화: z-score
  - Train: 80%, Validation : 20%
  - 10-fold stratified cross validation
  - Feature selection
    - 2800여개의 Feature 중 중요한 것만 남기는 Feature selection 기법을 적용해보았으나, Validation 기준으로 성능 하락 확인하여 False 설정
  - Fix_imbalance
    - train 데이터셋의 Y_Class 를 통해 데이터 불균형이 있음을 확인하고, SMOTE 를 통해 이를 해결했으나, Validation 기준으로 성능이 하락하는 것을 확인하여 False 설정

Pycaret

- 실험 환경
  - Use GPU (GPU RTX 3070)
  - OS: Windows 10
  - Python 3.7.16
  - pycaret[full] 3.0.0rc8

- 비고
  - 사용한 라이브러리 버전은 requirements.txt 파일에 작성했습니다.
  - GPU에 의해 완벽한 재현이 어려울 수 있습니다.
  - pycaret 의 save_model 함수를 통해 pkl 파일 저장 및 파일 첨부
  - load_model 함수를 통해 pkl 파일을 불러올 수 있습니다. (코드 참고)

Pycaret

- Compare models
  - 총 16개의 머신러닝 모델에 대해 동일한 조건에서의 성능 평가
  - GBC, CatBoost, LGBM이 F1-score 기준 가장 좋은 성능을 보임

Model	Accuracy	AUC	Recall	Prec.	F1
Gradient Boosting Classifier	0.8097	0.8084	0.8097	0.8249	0.7922
CatBoost Classifier	0.7972	0.7900	0.7972	0.8074	0.7724
Light Gradient Boosting Machine	0.7741	0.7812	0.7741	0.7806	0.7598
Extra Trees Classifier	0.7887	0.7866	0.7887	0.8103	0.7571
Extreme Gradient Boosting	0.7637	0.7742	0.7637	0.7655	0.7439
Random Forest Classifier	0.7721	0.7941	0.7721	0.7947	0.7278
Ada Boost Classifier	0.7364	0.6574	0.7364	0.7245	0.6938
SVM - Linear Kernel	0.7155	0.0000	0.7155	0.7079	0.6875
K Neighbors Classifier	0.7198	0.6652	0.7198	0.7125	0.6850
Decision Tree Classifier	0.6799	0.6545	0.6799	0.6895	0.6749

...

Gradient Boosting Classifier (GBC)

- Machine Learning의 Ensemble Method 중 Boosting에 속하는 Method

- Misclassification 된 샘플에 대한 가중치를 적용하고, 이전에 학습된 모델의 residual을 새로운 모델에 반영하여 약한 분류기를 강한 분류기로 조합하는 방식
- GBC는 일반적으로 다른 분류 모델보다 높은 성능을 보이고 높은 정확도와 일반화 능력을 가지고 있음
- Cost 가 큰 알고리즘 이며 Overfitting 의 가능성 존재

Light Gradient Boosting Machine (LGBM)

- Gradient Boosting 방법을 경량화한 방식

- 타 방법들은 수평으로 Tree를 확장(level-wise) 하지만, LGBM은 수직으로 Tree를 확장함(leaf-wise)
- 학습속도가 빠르고 메모리를 적게 차지함
- Overfitting에 민감함

CatBoost Classifier

- 기존의 문제점을 개선한 Boosting Ensemble 기법 중 하나
- 수평으로 Tree를 확장(level-wise) 하는 방식
- 학습데이터의 일부만으로 residual을 계산함
- 데이터에 Random Permutation을 적용하여 Overfitting을 방지
- Ordered Target Encoding 적용
- 불균형한 데이터셋에서도 파라미터 조정을 통해 성능을 높일 수 있음

- 단점: 학습에 시간이 많이 소요됨

Pycaret

- Parameter tuning 전 각 모델의 성능 비교
- 10 fold cross-validation의 평균 성능
- Optimized by F1-score

	Accuracy	AUC	Recall	Prec.	F1
Gradient Boosting Classifier	0.8097	0.8084	0.8097	0.8249	0.7922
CatBoost Classifier	0.7762	0.7829	0.7762	0.7832	0.7616
Light Gradient Boosting Machine	0.7972	0.7900	0.7972	0.8074	0.7724

Pycaret

- Parameter tuning 이후 각 모델의 성능 비교
- 10 fold cross-validation의 평균 성능
- Optimized by F1-score
- Iteration number for parameter tuning : 10 times

	Accuracy	AUC	Recall	Prec.	F1
Gradient Boosting Classifier	0.7951	0.7898	0.7951	0.8049	0.7715
CatBoost Classifier	0.7781	0.7750	0.7781	0.7883	0.7547
Light Gradient Boosting Machine	0.7657	0.7665	0.7657	0.7674	0.7448

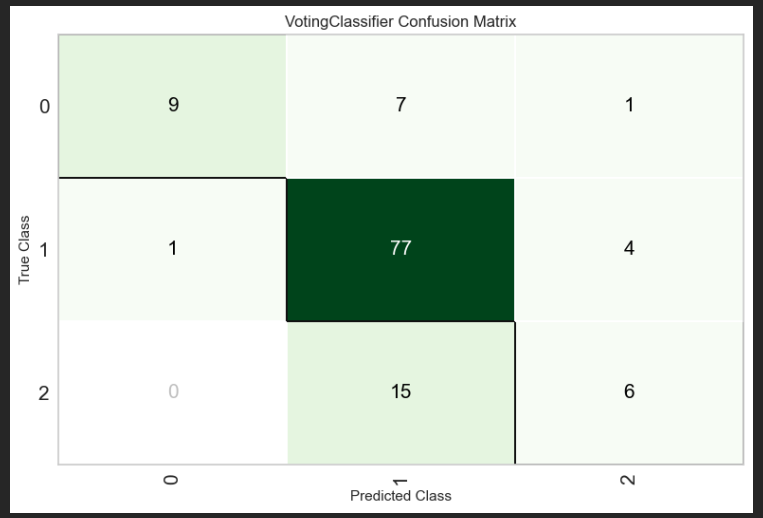
- Hyperparameter tuning 전보다 F1 score 기준 GBC 약 2%, CatBoost 1%, LGBM 3% 성능 하락
- 그러나 모델의 안정성을 위해 tuning을 진행함

Results

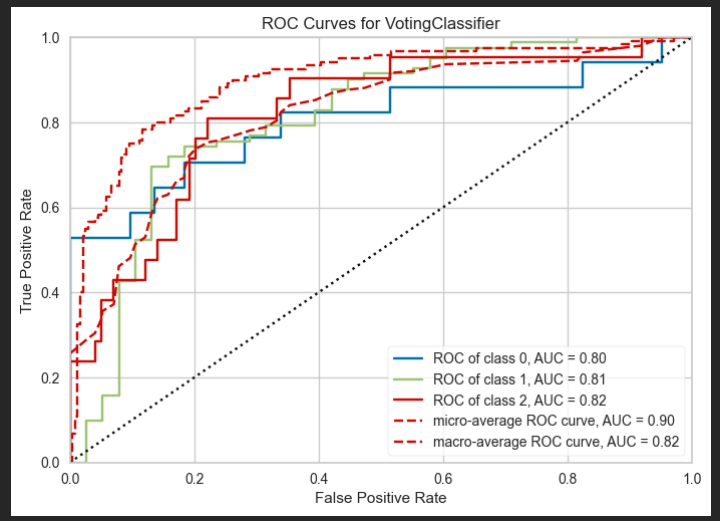
- 세 모델의 Blending 이후 성능 평가
- 10 fold cross-validation의 평균 성능
- Optimized by F1-score

	Accuracy	AUC	Recall	Prec.	F1
Blended model	0.8055	0.8000	0.8055	0.8137	0.7876

- Confusion Matrix (from validation set)



- ROC Curve



Results

- Validation set 을 포함하여 blended 모델 재 학습
- Test set 을 통한 모델 예측

- Performance
  - Public F1-score: 0.72546
  - Private F1-score: 0.65202