Python으로 머신러닝 입문

세션장: 구은아, 김은기

GBM

GBM 클래스

sklearn.ensemble.GradientBoostingClassifier

class sklearn.ensemble.GradientBoostingClassifier(*, *loss='deviance'*, *learning_rate=0.1*, *n_estimators=100*, *subsample=1.0*, *criterion='friedman_mse'*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_depth=3*, *min_impurity_decrease=0.0*, *init=None*, *random_state=None*, *max_features=None*, *verbose=0*, *max_leaf_nodes=None*, *warm_start=False*, *validation_fraction=0.1*, *n_iter_no_change=None*, *tol=0.0001*, *ccp_alpha=0.0*) [source]

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

XGboost

XGboost 하이퍼파라미터(부스터)

파라미터 이름	설명
learning_rate	경사하강법에서 최적값을 찾아가는 속도 디폴트는 0.1
n_estimators	기반이 되는 약한 학습기의 개수(반복 횟수) 디폴트는 100
subsample	약한 학습기가 학습에 사용하는 데이터 샘플링의 비율 디폴트는 1(전체 학습 데이터 사용)
max_depth	깊이를 결정 / 디폴트는 6
min_child_weight	트리에서 더 가지를 나눌지 결정하기 위해 필요한 데이터들의 weight 총합 클수록 분할을 자제 / 디폴트는 1

XGboost 하이퍼파라미터(학습)

파라미터 이름	설명
eval_metric	검증(validation)에 사용되는 함수 디폴트는 rmse(회귀) 또는 error(분류)
early_stopping_rounds	조기종료할 반복수를 지정 지정된 수만큼 반복하면서 검증이 향상되지 않으면 반복을 멈춤

LGBM

LGBM 하이퍼파라미터(부스터)

파라미터 이름	설명
learning_rate	경사하강법에서 최적값을 찾아가는 속도 디폴트는 0.1
n_estimators	기반이 되는 약한 학습기의 개수(반복 횟수) 디폴트는 100
subsample	약한 학습기가 학습에 사용하는 데이터 샘플링의 비율 디폴트는 1(전체 학습 데이터 사용)
max_depth	깊이를 결정 / 디폴트는 1
num_leaves	개별 트리가 가질 수 있는 최대 리프의 개수 / 디폴트는 31
min_child_samples	리프 노드가 되기 위해서 최소한으로 필요한 데이터 수 / 디폴트는 20

LGBM 하이퍼파라미터(학습)

파라미터 이름	설명
eval_metric	검증(validation)에 사용되는 함수 디폴트는 rmse(회귀) 또는 error(분류)
early_stopping_rounds	조기종료할 반복수를 지정 지정된 수만큼 반복하면서 검증이 향상되지 않으면 반복을 멈춤

하이퍼파라미터 튜닝

하이퍼파라미터 튜닝

하이퍼파라미터 튜닝이란?

나의 목적에 맞게 하이퍼파라미터를 변경하는 것!
좋은 예측 성능을 위해서 필수적인 과정이다.
특히 과적합을 방지하기 위해 수행된다.

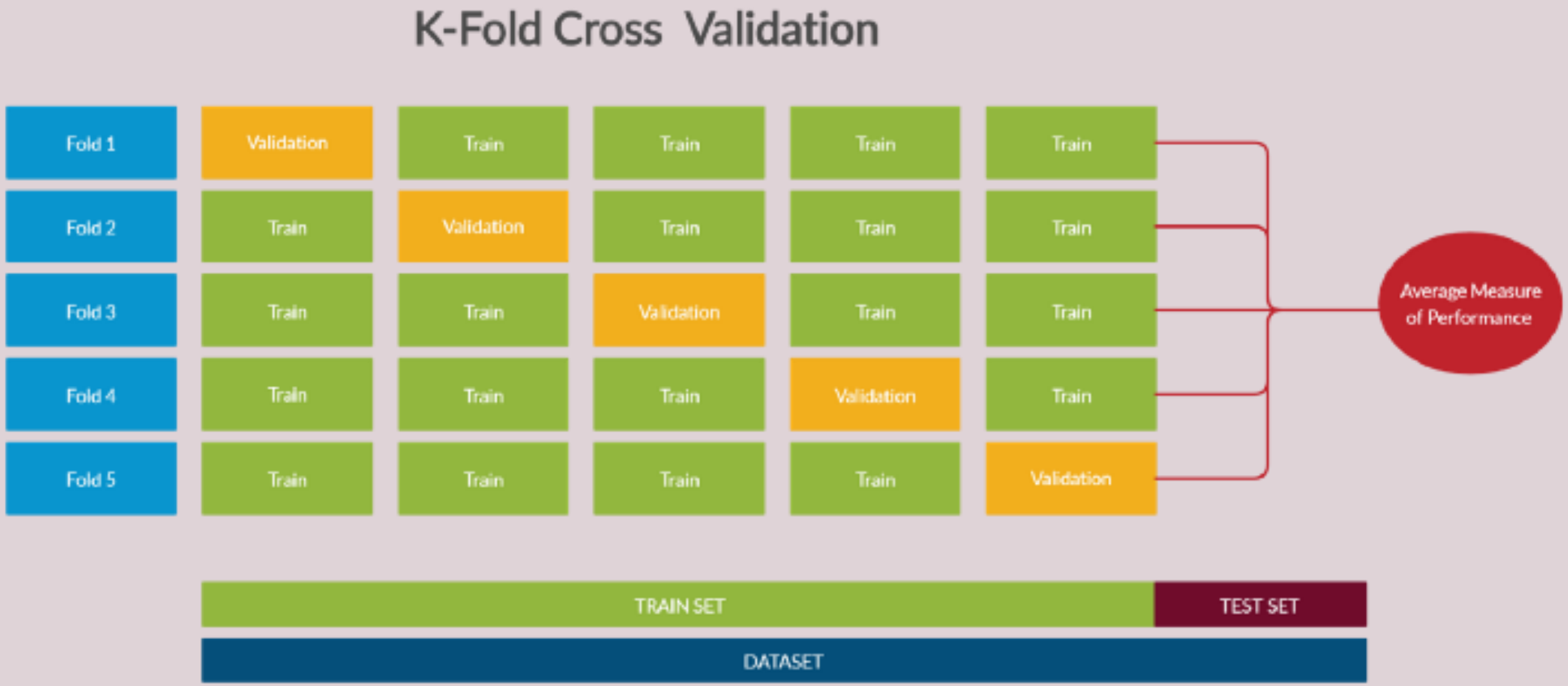
여러가지 값을 적용하여 학습시켜보고, 성능을 비교하면서 최적의 하이퍼파라미터를 찾는다.

GridSearch CV

사이킷런 내장 함수
파라미터를 딕셔너리 형태로 입력하면, 모든 가능한 조합으로 학습을 진행하고, 가장 좋은 하이퍼파라미터 조합을 찾는다.

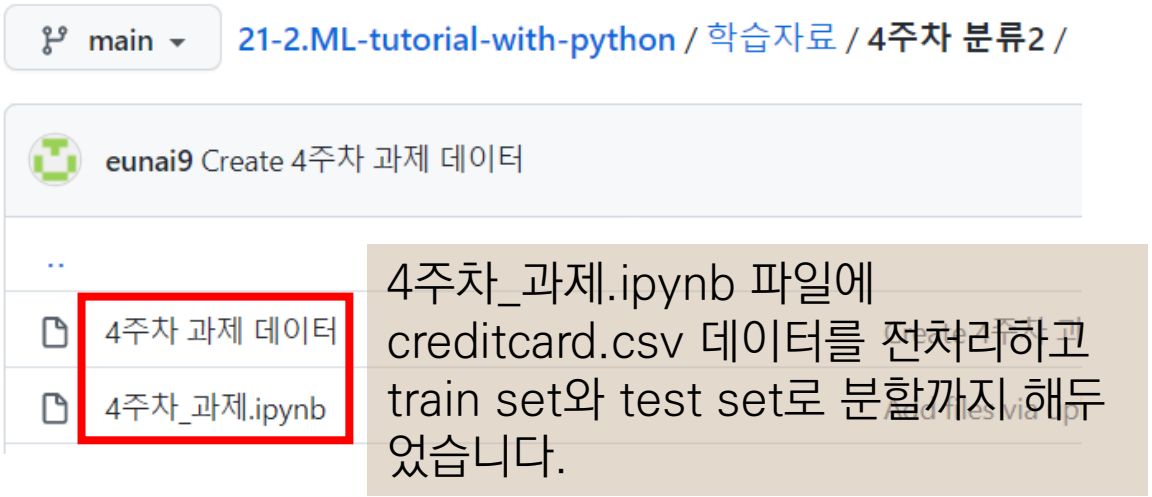
학습은 교차 검증으로 진행된다.

교차 검증(Cross Validation)



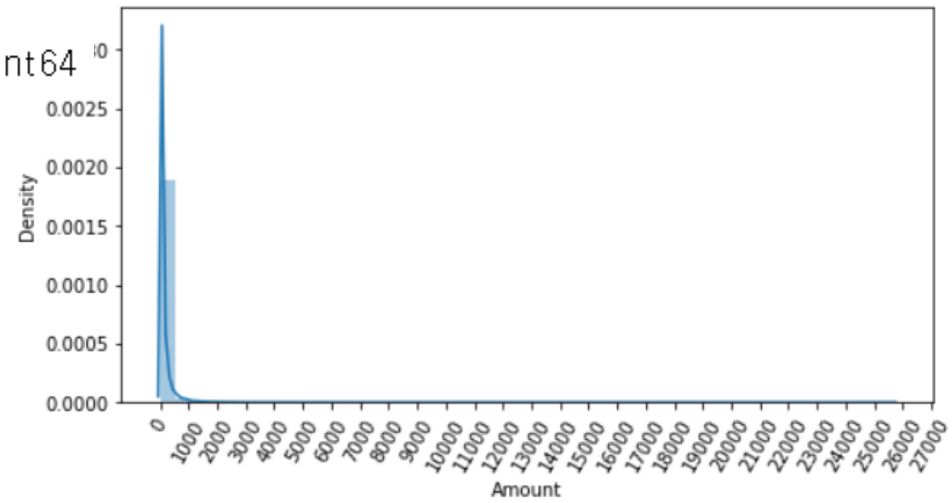
과제 안내

과제: 신용카드 사기 검출 실습



4주차_과제.ipynb 파일에 creditcard.csv 데이터를 전처리하고 train set와 test set로 분할까지 해두었습니다.

```
0     284315
1        492
Name: Class, dtype: int64
```



1) 비대칭이 심한 열 'Amount'은 예측 성능에 불이익을 줍니다. 이 문제를 해결할 수 있는 방법을 찾아 적용해봅시다.

2) Xgboost, LightGBM 등 다양한 분류 모델을 사용하여 예측해보고, 저저번주에 배웠던 평가지표로 성능을 확인해봅시다.

target이 극도로 불균형한 경우, LGBMClassifier의 하이퍼파라미터 boost_from_average를 True로 설정해야 합니다.

3) GridCV로 하이퍼파라미터 튜닝을 하며 최적의 파라미터 조합을 찾아봅시다.

수고하셨습니다!
과제 열심히 하시고 다음 주에 뵈어요~