트리 기반 모델들

1. LightGBM

- 장점 :

메모리는 적게 차지하고, 속도는 빠르다
결과의 정확도가 높다
GPU를 활용할 수 있다

- 단점 :

overfitting에 민감하여 데이터의 크기가 작을 경우 기존의 머신러닝 알고리즘이 더 좋다
데이터 행 수가 10000개 이상일 때 추천

2. HistGBM

```
def algorithm_pipeline(X_train_data, X_test_data, y_train_data, y_test_data,
                       model, param_grid, cv=10, scoring_fit='accuracy',
                       do_probabilities = False):
    gs = GridSearchCV(
        estimator=model,
        param_grid=param_grid,
        cv=cv,
        n_jobs=8,
        scoring=scoring_fit,
        verbose=2
    )
    fitted_model = gs.fit(X_train_data, y_train_data)

    if do_probabilities:
        pred = fitted_model.predict_proba(X_test_data)
    else:
        pred = fitted_model.predict(X_test_data)

    return fitted_model, pred
```

3. XGB

- 여러 개의 decision tree를 조합해서 사용하는 ensemble 알고리즘

- 장점 :

GBM 대비 빠른 수행시간
표준 GBM 경우 과적합 규제 기능이 없으나, XGB는 자체에 과적합 규제 기능으로 강한 내구성 가진다

4. CATBOOST

장점 :
leaf-wise(DFS) 가 아닌 level-wise(BFS) tree (max_depth = -1이면 같은 형태지만, max_depth != -1이면 다른 형태)
범주형 변수 처리 방법의 용이 (수치형 변수로 변환해서 one hout 시키지 않아도 된다, 단 작은 cardinality는 one hot이 더 효율적일 수도 있다)
overfitting을 방지하는 random permutation
기본 파라미터가 기본적으로 최적화가 잘 되어 있다

단점 :
sparse 한 matrix는 처리하지 못한다
데이터가 수치형 변수 인 경우 LGBM보다 학습 속도가 느리다

5. EXTRATREESCLASSIFIER

- 각 노드는 랜덤하게 특성의 서브셋을 만들어서 분할
- 랜덤포레스트 모델의 변종으로, 포레스트 트리의 각 후보 특성을 무작위로 분할하는 방식

- 장점 :

Feature Importance는 불확실도를 많이 낮출수록 증가하므로 이를 기준으로 변수를 선택