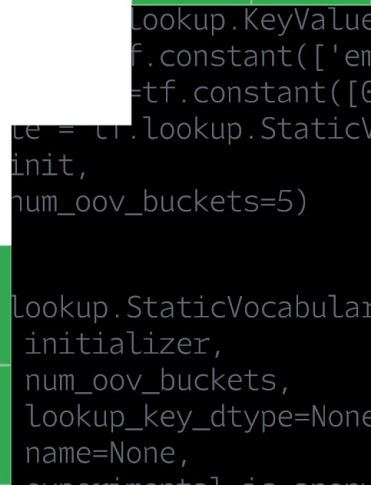
Google Developer Student Clubs

2주차 A조

팀원: 강용진, 조현진, 조선빈



강용진

Hyperparameter Tuning

hyperparameter들의 적절한 값들의 조합을 찾아가는 과정

1. **Manual Research**

   사람이 직접 하나하나 바꿔보면서 좋은 조합을 찾아가는 방법

2. **Grid Search**

   hyperparameter마다 몇 개의 값들을 정해둔 뒤 모든 경우의 수를 시도해보는 방법

```
param_gbm = {"max_depth" : [2,3,4,5,6],
             "min_samples_split" : [2,3,4,5,6],
             "learning_rate" : [0.01,0.05,0.1,0.2,0.3],
             "n_estimators" : [100,200,300,500,1000]
              }
```

<< 이러면 총 5*5*5*5=625개의 조합을 모두 시도함

→ 하나하나 다 시도하느라 시간이 오래 걸림

Hyperparameter Tuning

3. **Random Research**

hyperparameter마다 범위를 정해주면 그 범위 안에서 무작위의 값을 시도해보며 최적의 조합을 찾는 방법

```
param_distribs = {
    'n_estimators' : randint(low=1, high=500),
    'max_depth' : randint(low=3, high=20),
    'min_samples_split' : randint(low=2, high=20)
}
```

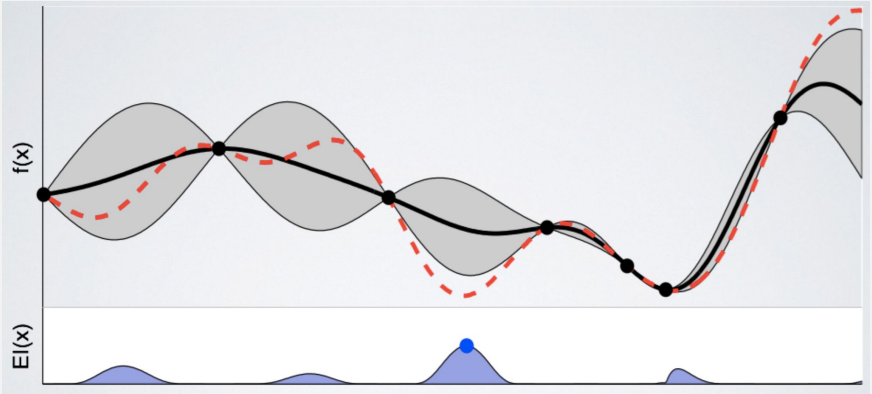
<< 범위를 입력

→ Grid Search보다 빠르나, 이전 탐색에 대한 정보를 다음 시도에서 활용하지 않음

Hyperparameter Tuning

4. Bayesian Optimization

이전에 탐색해본 정보를 이후에도 활용하며 최적의 값을 찾아가는 방법



- f(x)의 역할을 할 값(accuracy 등)을 정하고, 확률분포를 활용해 그 값을 추정하는 Surrogate Model과
- 더 나은 값이 나올 듯한 x의 값을 확률적으로 조사하는 Acquisition Function으로 구성

조현진

데이터 처리

- PassengerId
  - 이거 group 별로 쪼개 봤는데 크게 개선은 안됨
  - Submission 점수 0.78863 나옴
  - group에 대해서 오버피팅의 경향성이 나옴
- distance
  - 일직선에 있다고 가정해고 거리별로 이것저것해봄
  - 했는데 0.7912 나옴
- selectFromModel
  - threshold값을 바꾸어보았는데
  - 늘어나면 성능저하
  - 줄이면 진짜 조금 개선됨
- 지출
  - 어떤 사람이 지출이라는 col을 만들어서 나도 해봄
  - score가 증가 했음
- **GroupId**
  - 이거 하나만 써도 Acc 0.8 후반이 나오는 치트키
  - 근데 test.csv에서는 힘을 못씀

앙상블 모델

앙상블 모델이 여러 모델을 동시에 적용하는거야?

네, 앙상블 모델(Ensemble Model)은 여러 다른 기본 모델을 조합하여 예측을 수행하는 기법입니다. 이러한 다양한 모델을 조합하면 단일 모델보다 더 나은 예측 성능을 얻을 수 있습니다. 앙상블 모델은 주로 다음과 같은 두 가지 주요 방법으로 구성됩니다:

1. **배깅 (Bagging)**:
   - 배깅은 같은 유형의 다수의 모델을 동시에 적용하는 방법입니다.
   - 예를 들어, 랜덤 포레스트(Random Forest)는 여러 개의 의사결정 트리(Decision Tree) 모델을 배깅하여 만든 모델입니다. 각 트리는 서로 다른 부분 데이터셋에 대해 학습하고, 그 결과를 평균 또는 다수결 투표 등으로 결합하여 최종 예측을 수행합니다.
2. **부스팅 (Boosting)**:
   - 부스팅은 약한 모델을 순차적으로 학습하고 강력한 모델을 만드는 방법입니다.
   - 예를 들어, 그래디언트 부스팅(Gradient Boosting)은 이전 모델이 만든 오차를 보정하는 새로운 모델을 반복적으로 추가하여 앙상블을 형성합니다. 이로써 예측 성능을 향상시킵니다.

앙상블 모델 중 voting classifier모델을 사용함

voting의 두가지 방식

1. **Hard Voting (단순 투표)**:
   - Hard Voting은 다양한 기본 모델들이 예측한 클래스 중 가장 많이 나온 클래스를 최종 예측값으로 선택합니다. 다수결 투표와 비슷한 개념입니다.
   - 예를 들어, 세 개의 다른 모델이 예측한 클래스가 A, B, B인 경우, Hard Voting은 클래스 B를 최종 예측값으로 선택합니다.
2. **Soft Voting (가중 투표)**:
   - Soft Voting은 다양한 기본 모델들이 예측한 확률 값을 평균하여 가장 높은 확률을 가진 클래스를 최종 예측값으로 선택합니다.
   - 예를 들어, 세 개의 다른 모델이 예측한 클래스 A의 확률이 0.3, 클래스 B의 확률이 0.4, 클래스 C의 확률이 0.6인 경우, Soft Voting은 클래스 C를 최종 예측값으로 선택합니다.

조선빈

Overfitting

PassengerId → Group + GroupId

다양한 Encoder 적용

- OriginalEncoder()
  : 텍스트 데이터를 처리하는 텍스트 인코딩 방법 중 하나
  텍스트 데이터(문자) → 수치 데이터(고유의 숫자)

- MEstimateEncoder()
  : 주어진 데이터 분포를 기반으로 값을 변환하는 방법
  데이터 분포의 파라미터를 추정할 때 사용
  데이터 포인트를 특정한 확률 분포에 대한 추정치로 변환하는데 사용

- LocalOutlierFactor()
  : 이상치 탐지 알고리즘
  데이터 포인트의 이상치 여부를 판단하기 위해 주변 데이터 포인트들의 밀도와 비교

→ 오버피팅

Overfitting

- 특성 차원
  : 인코딩을 통해 데이터를 변환하면서 새로운 특성 공간이 생성되면서 특성의 차원 증가
  모델이 훈련 데이터에 더 적합해짐
  너무 많은 특성들로 인해 훈련 데이터의 노이즈까지 모델에 학습하게 되어 오버피팅 발생

- 인코딩 복잡성
  : 데이터를 더 복잡한 형태로 변환
  모델이 훈련 데이터를 더 정확하게 모델링할 수 있음
  테스트 데이터에 대한 일반화 능력 감소

- 데이터 부족
  : 인코딩한 후 원본 데이터보다 데이터 양이 부족해지는 경우가 있음
  오버피팅 발생

→ 훈련 데이터에 대한 정확도와 테스트 데이터에 대한 정확도의 차이가 **15%**정도 차이남

Ensemble

- Voting : 다수결 투표를 통해 예측
  - Hard : 다수 모델 중 가장 많은 투표를 받은 class 선택
  - Soft : 모든 모델의 예측 확률을 평균화하여 확률이 가장 높은 class 선택

- Bagging : 동일 모델을 여러번 훈련하고 예측을 평균화하여 안정성 향상
  ex. RandomForest

- Boosting : 약한 모델을 순차적으로 학습시켜 강력한 모델 구축
  ex. AdaBoost, Gradient Boosing, XGBoost, LightGBM

- Stacking : 여러 모델을 조합하여 메타 모델 훈련
  기본 모델들의 예측 결과를 입력으로 사용하여 메타 모델을 학습시켜 최종 예측 수행

→ Voting, Bagging, Boosting을 함께 활용해보았으나 유의미한 결과X
성능이 낮은 모델이 포함되어 있으면 오히려 성능이 낮아지는 경우도 생긴다고 예측함.
또한 각 모델들이 overfitting되어 ensemble 결과도 overfitting되었을 가능성도 있을 듯함.