1. About lgbm

Light gradient boosting machine

Gradient boosting : Gradient Descent + Boosting

부스팅(boosting)이란 단순한 학습기를 결합해서 보다 정확하고 강력한 학습기를 만드는 방식을 의미한다. 정확도가 낮더라도 일단 모델을 만들고, 드러난 예측 오류는 두번째 모델이 보완한다. 이 둘을 합치면 처음보다는 정확한 모델이 만들어지고, 그럼에도 여전히 남아 있는 문제는 다음 모델에서 보완하여 계속 더하는 과정을 반복하는 원리이다. Gradient Boosting을 알기에 앞서 손실함수는 예측 모델의 오류를 정량화해준다. 학습이란 손실함수를 최소화하는 파라미터를 찾는 일이라고 할 수있다. Gradient Descent는 손실함수를 파라미터로 미분해서 기울기를 구하고, 값이 작아지는 방향으로 파라미터를 움직이다 보면 손실함수가 최소화 되는 지점에 도달한다. Gradient Boosting은 손실함수를 파라미터가 아니라 현재까지 학습된 모델 함수로 미분한다. 정리하면 Gradient Boosting에서는 Gradient가 현재까지 학습된 모델의 약점을 드러내는 역할을 하고, 다른 모델이 그걸 중점적으로 보완해서 성능을 Boosting한다.

1. 사용한 패키지

numpy: A library thatplays a key role when used in computational science. Provides high-

performance multidimensional array objects and tools to handle them

pandas: **pandas** is a [Python](http://www.python.org/) package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, **real world** data analysis in Python.

lightgbm: lightgb을 사용하기 위한 패키지

1. 데이터의 전처리
   1. registration\_init\_time, expiration\_date를 각각 연월일로 세분화하여 feature에 추가 한다.
   2. 노래정보의 해당하는 isrc를 통해 해당 노래의 연도를 추출하여 feature에 추가한다.
   3. song\_length 중 에서 na인 원소를 평균값으로 채워준다.
   4. lyricist, composer, artist 중 에서 na인 원소를 각각 no\_lyricist, no\_composer,no\_artist로 바꾸어 준다.
   5. 각 노래당 작곡, 작사, 아티스트가 몇 명인지 세고 그 각각의 수를 feature에 추가한다.
   6. 해당 노래가 피쳐링되었는 지 안되었는 지도 새로운 feature로 추가한다.
   7. 아티스트와 작곡가가 같은 지도 artist\_composer라는 feature에 추가한다.
   8. 아티스트, 작사가, 작곡가가 같은 지도 artist\_composer\_lyricist라는 feature에 추가한다.
   9. 노래길이가 평균의 절반도 안될 경우 0, 평균 이하일 경우 1, 평균의 1.5배보다 작은 경우 2 나머지는 3으로 값을 같는 feature를 추가한다.
   10. 노래가 재생된 횟수를 feature에 추가한다.
   11. 해당 노래의 가수의 노래 재생 횟수를 feature에 추가한다.
2. Lgbm의 파라미터
   1. Learning\_rate: numeric. The shrinkage rate applied to each iteration.
   2. Boosting: string. Boosting type.
   3. objective : string. Specify the learning task and the corresponding learning objective or a custom objective function to be used (see note below).
   4. max\_depth: limit the max depth for tree model. This is used to deal with over-fitting when #data is small. Tree still grows by leaf-wise
   5. num\_leaves: number of leaves in one tree
   6. verbosity: verbosity, default=1, type=int, alias=verbose<0 = Fatal, =0 = Error (Warn), >0 = Info
   7. metric: metric, default={l2 for regression}, {binary\_logloss for binary classification}, {ndcg for lambdarank}, type=multi-enum, options=l1, l2, ndcg, auc, binary\_logloss, binary\_error
   8. verbose : bool. If True and an evaluation set is used, writes the evaluation progress.
   9. baggig\_fraction: double. this will randomly select part of data without resampling. can be used to speed up training and deal with over-fitting.
   10. bagging\_freq: frequency for bagging, 0 means disable bagging. k means will perform bagging at every k iteration.
   11. bagging\_seed: random seed for bagging
   12. feature\_fraction: LightGBM will randomly select part of features on each iteration if feature\_fraction smaller than 1.0. For example, if set to 0.8, will select 80% features before training each tree.
   13. Feature\_fraction\_seed: random seed for feature\_fraction.
   14. Max\_bin: max number of bins that feature values will be bucketed in. Small number of bins may reduce training accuracy but may increase general power (deal with over-fitting)
3. 결과