머신 러닝 - 분류

분류

1. Decision Tree
    - 트리 기반의 알고리즘은 하이퍼 파라미터가 너무 많고, 그로 인한 튜닝 시간이 많이 소모되는 데에 비해 예측 성능이 크게 향상되는 경우가 많지 않다는 단점이 있음
    - 엔트로피(Entropy), 정보 이득(Information Gain)을 이용한 분류
        - 엔트로피 : 주어진 데이터 집합의 혼잡도, 서로 다른 값이 섞여 있으면 엔트로피가 높고 같은 값이 주로 있으면 엔트로피가 낮음
        - 정보 이득 지수 : '1-엔트로피' 로 정보 이득이 높을 수록 해당 항목을 분류하면 정보의 분류성이 높다는 것을 뜻하므로 결정 트리는 이 정보 이득 지수로 분할 기준을 정함
        - 지니 계수 : 0이 가장 평등 ~ 1이 가장 불평등, 지니 계수가 낮을 수록 데이터 균일도가 높은 것이므로 지니 계수가 낮은 속성을 기준으로 분할
2. 앙상블 학습 (Ensemble Learning)
    - 여러 개의 분류기를 생성하고 예측을 결합하여 보다 정확한 최종 예측 도출 (분류기들의 팀플)

    > Federated Learning 에서의 aggregation 과 비슷한가? 전혀 다름, FL 의 aggregation 과정은 아예 같은 딥러닝 모델의 다른 데이터로 훈련시킨 결과 weights 들을 평균 짓는 것이고 (모델 훈련 과정의 영역), 앙상블은 여러 분류기가 예측한 결과를 평균 짓는 것 (추론 과정의 영역)

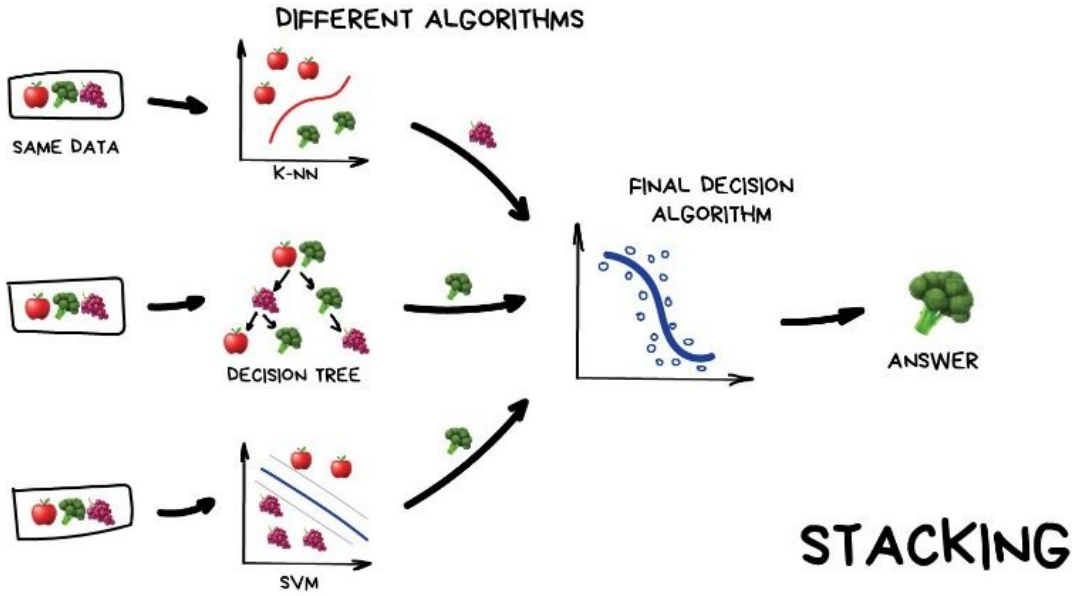
3. 랜덤 포레스트 (Random Forest)
    - 같은 여러 개의 결정 트리 분류기 (Decision Tree Classifier) 가 전체 데이터에서 배깅 (Bagging) 방식으로 각자의 데이터를 샘플링해 개별적으로 학습을 수행한 뒤 최종적으로 모든 분류기가 보팅 (Voting) 을 통해 예측을 결정 (앙상블 학습의 일종)
4. GBM (Gradient Boosting Machine)
    - Boosting 알고리즘은 여러 개의 약한 학습기 (weak learner) 를 순차적으로 학습-예측 하면서 잘못 예측한 데이터에 가중치 부여를 통해 오류를 개선해 나가면서 학습하는 방식
    - 대표적인 Boosting 방식으로 AdaBoost (Adaptive boosting), GBM 가 있음
        - AdaBoost : 개별 학습을 순차적으로 진행 후 가중치 부여해 결합
        - GBM : 가중치 업데이트를 경사 하강법을 이용
            - 일반적으로 GBM 은 Random Forest 보다는 예측 성능이 조금 뛰어난 경우가 많지만 수행 시간이 오래 걸리고, hyper parameter tuning (ex. learning rate, n_estimators etc) 노력이 더 필요함
            - 사이킷런의 GradientBoostingClassifier는 약한 학습기의 순차적인 예측 오류 보정을 통해 학습을 수행해 멀티 CPU 코어 시스템을 사용하더라도 병렬 처리가 지원되지 않아 대용량 데이터의 경우 학습에 많은 시간이 필요
5. XGBoost (eXtra Gradient Boost)
    - GRM 을 기반으로 만들어진 트리 기반의 앙상블 학습에서 각광받고 있는 알고리즘 중 하나
    - GBM 의 단점인 느린 수행 시간 및 과적합 규제 부재 등의 문제를 해결 (병렬 CPU에서 병렬 학습이 가능, 과적합에 좀 더 강한 내구성)
6. LightGBM

- XGBoost 보다 수행 시간이 적고 메모리 사용량도 적은데 예측 성능의 별다른 차이가 없음, 10,000건 이하의 데이터 세트에는 과적합이 발생하기 쉬움
- 리프 중심 트리 분할 (Leaf Wise) 방식을 사용

> 기존의 대부분 트리 기반 알고리즘은 트리의 깊이를 효과적으로 줄이기 위한 균형 트리 분할(Level Wise) 방식을 사용함 -> 최대한 균형 잡힌 트리를 유지하면서 분할하기에 트리의 깊이가 최소화될 수 있음 -> 오버피팅(과적합)에 더 강한 구조를 가지기 위해
>
> LightGBM 의 리프 중심 트리 분할 방식은 트리의 균형을 맞추지 않고, 최대 손실 값(Max delta loss)을 가지는 리프 노드를 지속적으로 분할하면서 트리의 깊이가 깊어지는 비대칭적인 규칙 트리 생성 -> 학습을 반복할수록 결국 균형 트리 분할 방식보다 예측 오류 손실을 최소화 할 수 있음

7. 스태킹(Stacking) 앙상블
   - 여러 개별적인 알고리즘을 서로 결합하여 예측 결과를 도출하는데, 개별 알고리즘으로 예측한 데이터를 기반으로 다시 예측을 수행

   > 예로, training 데이터를 KNN, 결정트리, SVM 에 훈련하여 생성된 개별 예측 결과들을 한 array로 묶고, 이 array를 final 모델에 training 데이터로 취급하여 모델을 훈련시켜서 결과 예측을 하게 만드는 것



8. CV 스태킹
   - 앞서 말한 스태킹 앙상블의 과적합을 막기 위한 방법으로, final 모델의 훈련 데이터와 테스트 데이터를, 앞서 개별 모델들의 의해 생성시키는 방식

모델 성능 평가

- 정확도 (Accuracy)
  - 정확도는 불균형한(imbalanced) 레이블 값 분포에서 ML 모델 성능 판단에 적합하지 않다.

  > 예로, 100개의 데이터 중 90개가 0이고 10개의 데이터가 1인경우, 무조건 0으로 결과값을 예측해도 정확도가 90%가 된다.

- 오차 행렬 (confusion matrix)

- 예측 오류와 어떠한 유형의 예측 오류가 발생하고 있는 지를 나타낸다.
- TN, FP, FN, TP 형태로 구성 (이진 분류에서 잘 활용)

> TN : 0이라고 분류했는데 예측값이 맞는 경우
>
> FP : 1이라고 분류했는데 예측값이 틀린 경우

- 정밀도(Precision) & 재현율(Recall)
  - 정밀도 = TP / (FP+TP) : 1로 분류한 것 중 맞는 것
  - 재현율 = TP / (FN+TP) : 실제 값이 1인 것 중 맞은 것

> 재현율이 중요한 경우는 실제 Positive 데이터를 Negative로 잘못 판단하면 영향이 더 큰 경우 ex) 금융사기 판단
>
> <-> 정밀도는 반대의 상황

- F1 스코어 (score)
  - 정밀도와 재현율을 결합한 지표
  - `F1=(2*precision*recall)/(precision+recall)`
- ROC(Receiver Operation Characteristic Curve) 곡선과 AUC
  - 이진 분류에 중요하게 사용되는 성능 지표
  - ROC 곡선 - FPR(False Positive Rate)이 X축, TPR(True Positive Rate) 가 Y축인 FPR의 변화에 따른 TPR의 변화 곡선