3주차 예습과제

4.1 분류의 개요

분류 : 기존 데이터가 어떤 레이블에 속하는지 알고리즘으로 인지 → 새롭게 관측된 데이터에 대한 레이블 판별

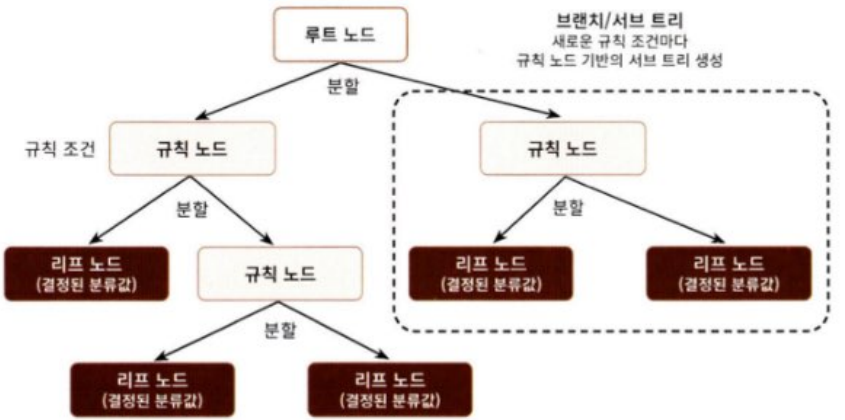
분류 알고리즘

- 나이브 베이즈
- 로지스틱 회귀
- 결정 트리
- 서포트 벡터 머신
- 최소 근접 알고리즘
- 신경망
- 앙상블
  - 배깅 - 랜덤 포레스트
  - 부스팅 - 그래디언트 부스팅

4.2 결정트리

결정트리 : 스무고개 게임같음. if, else를 자동으로 찾아내 예측을 위한 규칙을 만드는 알고리즘

> 트리의 깊이가 깊어질수록 결정트리의 예측 성능 저하 가능성 high



규칙 조건을 만드는 방식

- 정보 균일도가 높은 데이터 세트를 먼저 선택할 수 있도록 규칙 조건을 만든다
- 균일 데이터가 높은 서브데이터 세트를 계속해서 쪼개며 내려가기

정보이득지수 : 정보의 균일도를 측정하는데에 쓰임. 1 - 엔트로피 지수

지니 계수 : 결정 트리 알고리즘에서 데이터 세트를 분할 할 때 사용되는 기준.

- 지니계수가 낮을 수록 데이터 균일도가 높음

결정트리 장점 🙂

- 쉽다, 직관적이다
- 피처의 스케일링이나 정규화 등의 사전 가공 영향도가 크지 않다.

결정트리 단점

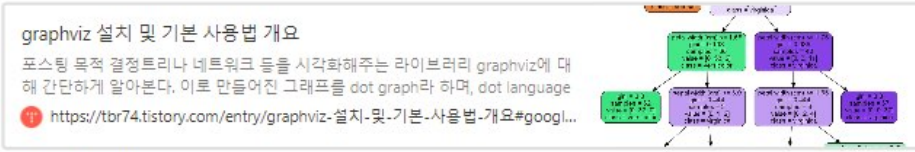
- P : 과적합으로 알고리즘 성능이 떨어진다.
- S : 트리의 크기를 사전에 제한하는 튜닝

결정트리 파라미터

파라미터 명	설명
min_samples_split	• 노드를 분할하기 위한 최소한의 샘플 데이터 수로 과적합을 제어하는 데 사용됨. • 디폴트는 2이고 작게 설정할수록 분할되는 노드가 많아져서 과적합 가능성 증가 • 과적합을 제어. 1로 설정할 경우 분할되는 노드가 많아져서 과적합 가능성 증가
min_samples_leaf	• 말단 노드(Leaf)가 되기 위한 최소한의 샘플 데이터 수 • Min_samples_split와 유사하게 과적합 제어 용도. 그러나 비대칭적(imbalanced) 데이터의 경우 특정 클래스의 데이터가 극도로 작을 수 있으므로 이 경우는 작게 설정 필요.

파라미터 명	설명
max_features	• 최적의 분할을 위해 고려할 최대 피처 개수. 디폴트는 None으로 데이터 세트의 모든 피처를 사용해 분할 수행. • int 형으로 지정하면 대상 피처의 개수, float 형으로 지정하면 전체 피처 중 대상 피처의 퍼센트임 • 'sqrt'는 전체 피처 중 sqrt(전체 피처 개수), 즉 $\sqrt{\text{전체 피처 개수}}$만큼 선정 • 'auto'로 지정하면 sqrt와 동일 • 'log'는 전체 피처 중 log2(전체 피처 개수) 선정 • 'None'은 전체 피처 선정
max_depth	• 트리의 최대 깊이를 규정. • 디폴트는 None. None으로 설정하면 완벽하게 클래스 결정 값이 될 때까지 깊이를 계속 키우며 분할하거나 노드가 가지는 데이터 개수가 min_samples_split보다 작아질 때까지 계속 깊이를 증가시킴. • 깊이가 깊어지면 min_samples_split 설정대로 최대 분할하여 과적합할 수 있으므로 적절한 값으로 제어 필요.
max_leaf_nodes	• 말단 노드(Leaf)의 최대 개수

결정 트리 모델의 시각화 : `Graphviz`

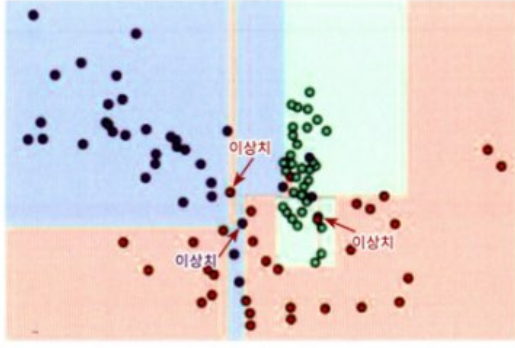


- 설치 방법

`export_graphviz()` : graphvix가 읽어 들여서 그래프 형태로 시각화할 수 있는 출력파일을 생성

- 인자 : 학습이 완료된 estimator, output 파일명, 결정클래스의 명칭, 피처의 명칭

결정 트리 과적합(Overfitting)



- 일부 이상치 데이터까지 분류하기 위해 분할이 자주 일어남 → 결정 기준 경계가 매우 많아짐
- 약간만 다른 형태의 데이터 세트를 예측하면 예측 정확도 떨어짐. 😲

4.3 앙상블 학습

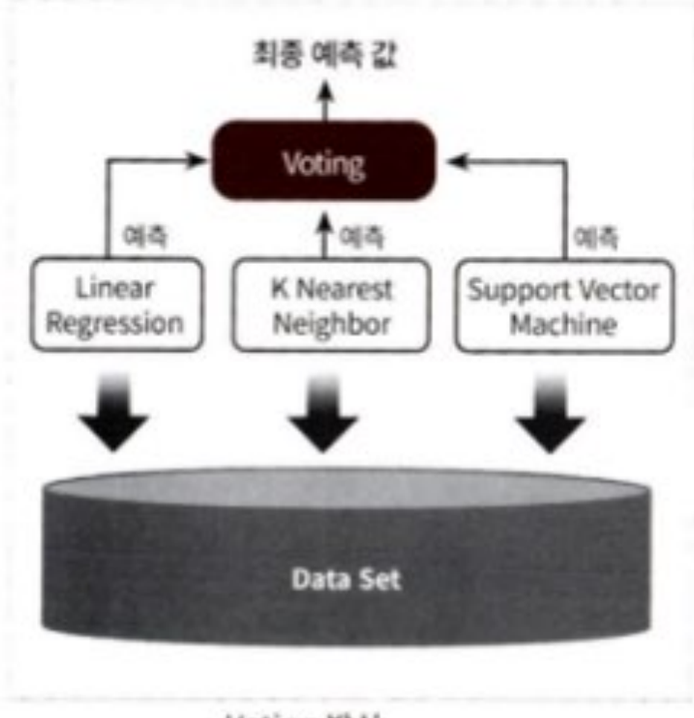
앙상블 학습 : 여러 개의 분류기 생성 → 예측들을 결합 → 보다 정확한 최종 예측 도출

- 보팅
- 배깅
- 부스팅

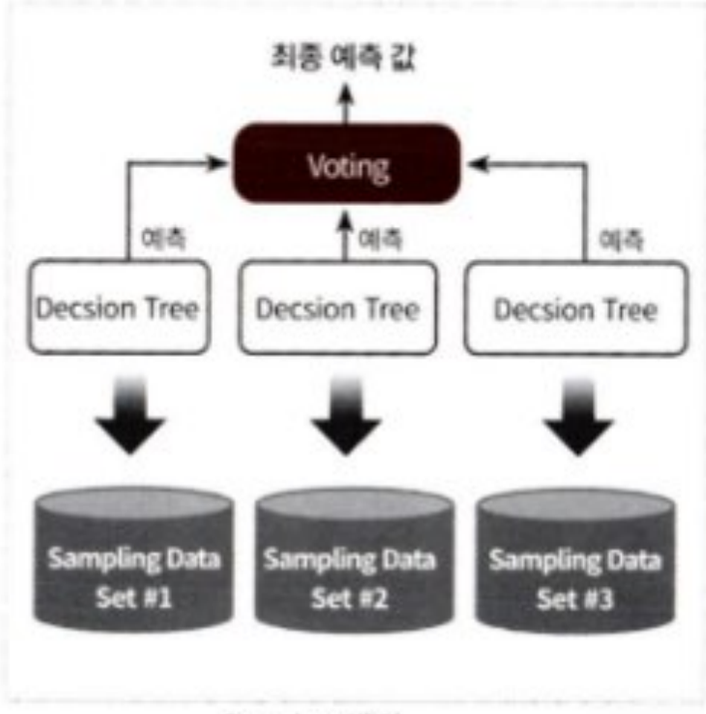
보팅 & 배깅 : 여러 개의 분류기가 투표를 통해 최종 예측 결과를 결정

차이점

- 보팅 : 일반적으로 서로 다른 알고리즘을 가진 분류기를 결합
- 배깅 : 각가의 분류기가 모두 같은 유형의 알고리즘 기반 but, 데이터 샘플링을 서로 다르게 가져가서 학습 → 보팅



Voting 방식



Bagging 방식

보팅 유형 - 하드 보팅 / 소프트 보팅

하드 보팅 : 다수의 분류기가 결정한 예측값을 최종 보팅 결괏값으로 선정

소프트 보팅 : 분류기들의 레이블 값 결정 확률을 모두 더함 → 평균해서 이들 중 확률이 가장 높은 레이블 값을 최종 보팅 결괏값으로 선정

- 일반적으로 소프트 보팅방법이 적용됨.

4.4 랜덤 포레스트

앙상블 - 보팅, 배깅, 부스팅 중에 배깅 에 해당

배깅 : 같은 알고리즘으로 여러 개의 분류기를 만들어서 보팅으로 최종 결정하는 알고리즘

- 대표 알고리즘 : 랜덤 포레스트

랜덤 포레스트

- 빠른 수행 속도
- 높은 예측 성능
- 기반 알고리즘 : 결정 트리 → 쉽고 직관적 :)

기반 알고리즘은 결정트리 BUT 개별 트리가 학습하는 데이터 세트는 전체 데이터에서 일부가 중첩되게 샘플링되어있음. → 부트스트래핑(bootstrapping)

💡 랜덤 포레스트는 중첩된 개별 데이터 세트에 결정 트리 분류기를 각각 적용하는 것이다!