데이터문제해결및실습1

4주차-2
Chapter_05_다시살펴보는머신러닝주요개념

세종대학교
인공지능데이터사이언스학과
박동현 교수

- 본 강의는 골든래빗 출판사의 머신러닝/딥러닝 문제해결전략이 제공하는 강의 교안에 기반함.

Chapter

05

다시살펴보는 머신러닝주요개념

□ 학습 목표

□ 다루는 내용

2부의 경진대회를 푸는 데 필요한 주요 머신러닝 개념들을 요약·정리했다. 경진대회 문제를 풀다가 언뜻 떠오르지 않는 개념이 있을 때 이번 장을 참고하자.


주요 머신러닝 모델
선형 회귀
로지스틱 회귀
결정 트리
앙상블
랜덤 포레스트
XGBoost
LightGBM
하이퍼파라미터 최적화
그리드서치
랜덤서치
베이지안 최적화

5.1 분류와 회귀

5.1.1 분류

- **분류**(classification) : 어떤 대상을 정해진 범주에 구분해 넣는 작업
- **이진분류**(binary classification) : 타깃값이 두 개인 분류
- **다중분류**(multiclass classification) : 타깃값이 세 개 이상인 분류

5.1.1 회귀

- **독립변수**(independent variable) : 영향을 미치는 변수
  예) 학습 시간, 수면의 질, 공장의 재고 등
- **종속변수**(dependent variable) : 영향을 받는 변수
  예) 시험 성적, 건강, 회사 이익 등
- **회귀**(regression) : 독립변수와 종속변수 간 관계를 모델링하는 방법
- **단순선형회귀**(simple linear regression): 독립변수 하나(x)와 종속변수 하나(Y) 사이의 관계를 나타낸 모델링 기법
- **다중선형회귀**(multiple linear regression): 독립변수 여러 개와 종속변수 하나 사이의 관계를 나타낸 모델링 기법
- 회귀 문제에서는 주어진 독립변수(피처)와 종속변수(타깃값) 사이의 관계를 기반으로 최적의 회귀계수를 찾아야 한다.

5.1 분류와 회귀

5.1.1 회귀

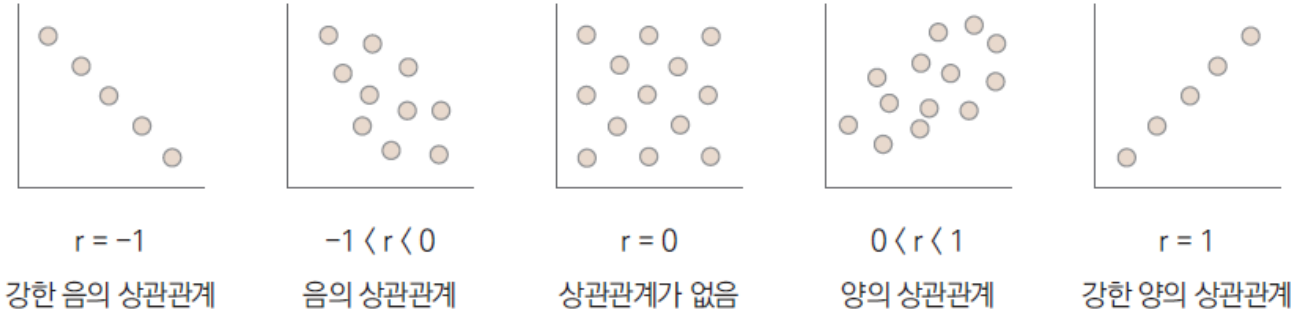
- **회귀 평가지표**

회귀 평가지표	수식	설명
MAE	$\frac{1}{N}\sum_{i=1}^{N}\lvert y_i-\hat{y}\rvert$	평균 절대 오차Mean Absolute Error. 실제 타깃값과 예측 타깃값 차의 절댓값 평균
MSE	$\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y})^2$	평균 제곱 오차Mean Squared Error. 실제 타깃값과 예측 타깃값 차의 제곱의 평균
RMSE	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y})^2}$	평균 제곱근 오차Root Mean Squared Error. MSE에 제곱근을 취한 값
MSLE	$\frac{1}{N}\sum_{i=1}^{N}(\log(y_i+1)-\log(\hat{y}_i+1))^2$	Mean Squared Log Error. MSE에서 타깃값에 로그를 취한 값
RMSLE	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log(y_i+1)-\log(\hat{y}_i+1))^2}$	Root Mean Squared Log Error. MSLE에 제곱근을 취한 값
R^2	$\frac{\hat{\sigma}^2}{\sigma^2}$	결정계수. 예측 타깃값의 분산 / 실제 타깃값의 분산 * 다른 지표와 다르게 1에 가까울수록 모델 성능이 좋습니다.

5.1 분류와 회귀

5.1.1 회귀

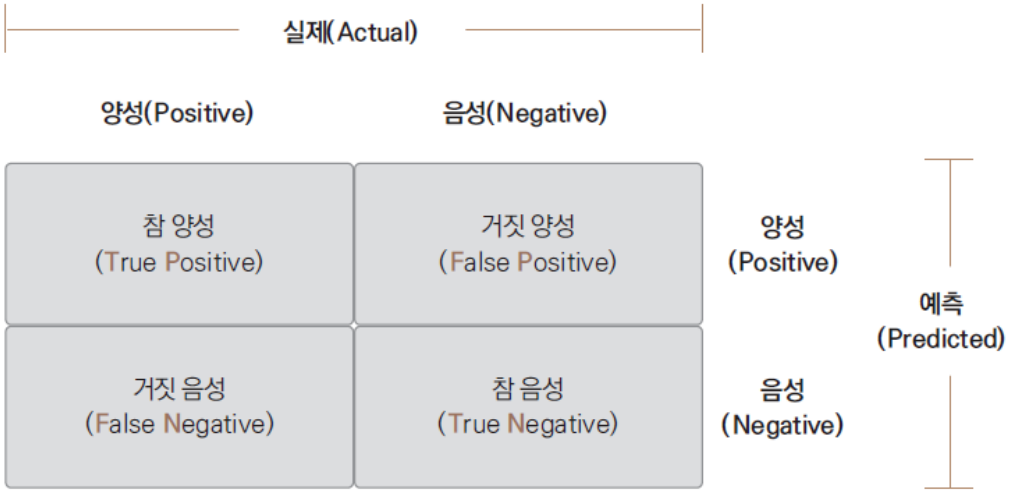
- **상관계수**(correlation coefficient) : 두 변수 사이의 상관관계 정도를 수치로 나타낸 값
- **피어슨 상관계수**(pearson correlation coefficient) : 선형 상관관계의 강도(strength)와 방향(direction)을 나타내며, -1부터 1 사이의 값을 갖는다. 상관계수가 음수면 음의 상관관계, 양수면 양의 상관관계가 있다 한다.



5.1 분류 평가지표

5.2.1 오차행렬

- **오차 행렬**(confusion matrix) : 실제 타깃값과 예측 타깃값이 어떻게 매칭되는지를 보여주는 표



5.1 분류 평가지표

5.2.1 오차행렬

- **정확도**(accuracy) : 실젯값과 예측값이 얼마나 일치되는지를 비율로 나타낸 평가지표

$$\frac{TP+TN}{TP+FP+FN+TN}$$

- **정밀도**(precision) : 양성 예측의 정확도

$$\frac{TP}{TP+FP}$$

- **재현율**(recall) : 실제 양성 값(TP + FN) 중 양성으로 잘 예측한 값(TP)의 비율. 재현율은 민감도(sensitivity) 또는 참 양성 비율(true positive rate, TPR)라고도 한다.

$$\frac{TP}{TP+FN}$$

- **F1 점수**(F1 score): 정밀도와 재현율을 조합한 평가지표

$$F1 = \frac{2}{\frac{1}{precision}+\frac{1}{recall}} = 2 \times \frac{precision \times recall}{precision + recall}$$

5.1 분류 평가지표

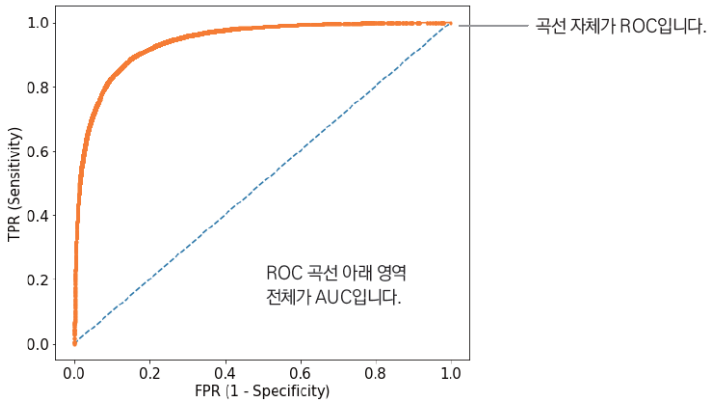
5.2.2 로그손실

- **로그 손실**(logloss) : 분류 문제에서 타깃값을 확률로 예측할 때 기본적으로 사용하는 평가지표. 값이 작을수록 좋은 지표다.

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log\left(\hat{y}_i\right) + \left(1-y_i\right)\log\left(1-\hat{y}_i\right)\right)$$

5.2.3 ROC 곡선과 AUC

- **ROC**(Receiver Operating Characteristic) : 참 양성 비율(TPR)에 대한 거짓 양성 비율(False Positive Rate, FPR)곡선
- **AUC**(Area Under the Curve) : ROC 곡선 아래 면적



5.3 데이터 인코딩

5.3.1 레이블 인코딩(label encoding)

- 범주형 데이터를 숫자로 일대일 매핑해주는 인코딩 방식. 범주형 데이터를 숫자로 치환하는 것.
- https://www.kaggle.com/werooring/ch5-categorical-data-encoding

5.3 데이터 인코딩

5.3.2 원-핫 인코딩(one-hot encoding)

- 여러 값 중 하나(one)만 활성화(hot)하는 인코딩

원본

과일
사과
블루베리
바나나
귤
블루베리
바나나
바나나
사과

→

원-핫 인코딩 적용 후

과일_귤	과일_바나나	과일_블루베리	과일_사과
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0
0	0	1	0
0	1	0	0
0	1	0	0
0	0	0	1

5.4 피처 스케일링

피처 스케일링(feature scaling) : 서로 다른 피처 값의 범위(최댓값 – 최솟값)가 일치하도록 조정하는 작업. 값의 범위가 데이터마다 다르면 모델 훈련이 제대로 안 될 수도 있다.

5.4.1 min-max 정규화

- 피처 값의 범위를 0~1로 조정하는 기법. 조정 후 최솟값은 0, 최댓값은 1이 된다.

$$x_{scaled} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

5.4.2 표준화(standardization)

- 평균이 0, 분산이 1이 되도록 피처 값을 조정하는 기법. min-max 정규화와 다르게 표준화는 상한과 하한이 없다.

$$x_{scaled} = \frac{x - \bar{x}}{\sigma}$$

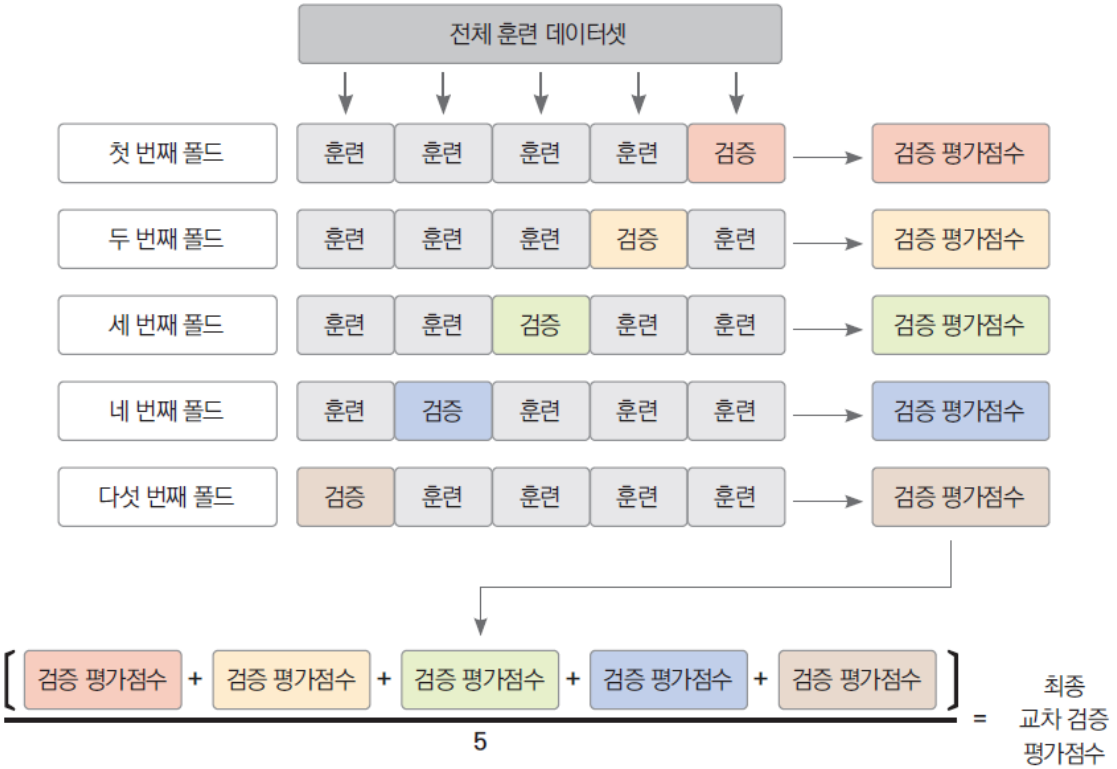
5.5 교차 검증

5.5.1 K 폴드 교차 검증

1. 전체 훈련 데이터를 K개 그룹으로 나눈다.
2. 그룹 하나는 검증 데이터로, 나머지 K-1개는 훈련 데이터로 지정한다.
3. 훈련 데이터로 모델을 훈련하고, 검증 데이터로 평가한다.
4. 평가점수를 기록한다.
5. 검증 데이터를 다른 그룹으로 바꿔가며 2~4 절차를 K번 반복한다.
6. K개 검증 평가점수의 평균을 구한다.

5.5 교차 검증

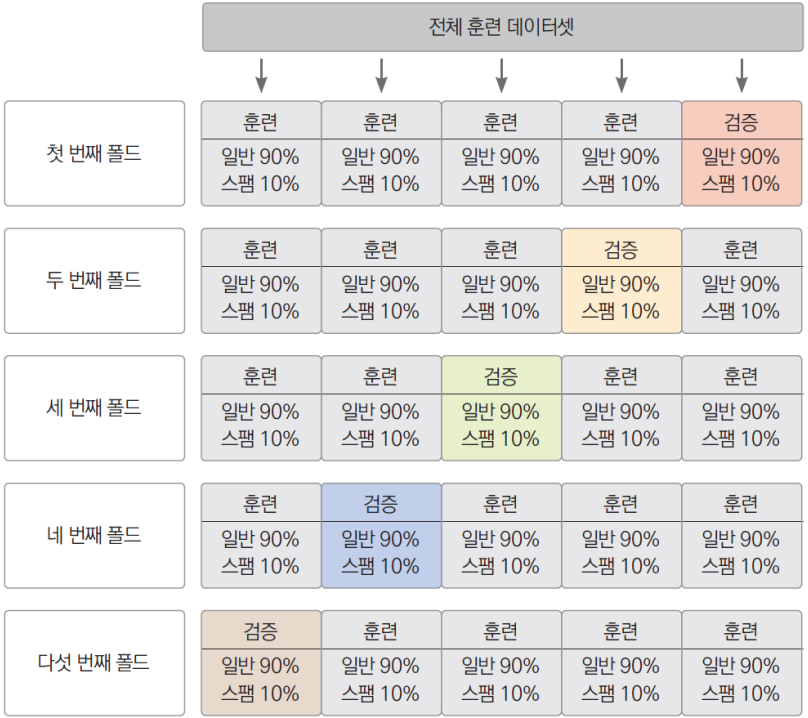
5.5.1 K 폴드 교차 검증



5.5 교차 검증

5.5.2 층화 K 폴드 교차 검증(Stratified K-Fold Cross Validation)

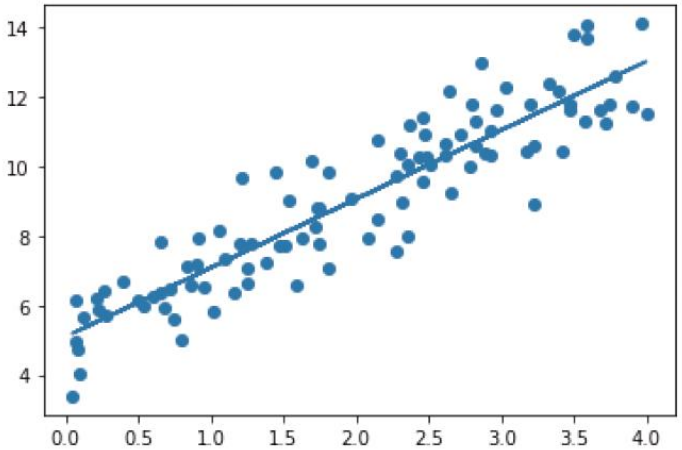
- 타깃값이 골고루 분포되게 폴드를 나누는 K 폴드 교차 검증 방법. 타깃값이 불균형하게 분포되어 있을 때 층화 K 폴드를 사용하면 좋다.



5.6 주요 머신러닝 모델

5.6.1 선형 회귀 모델

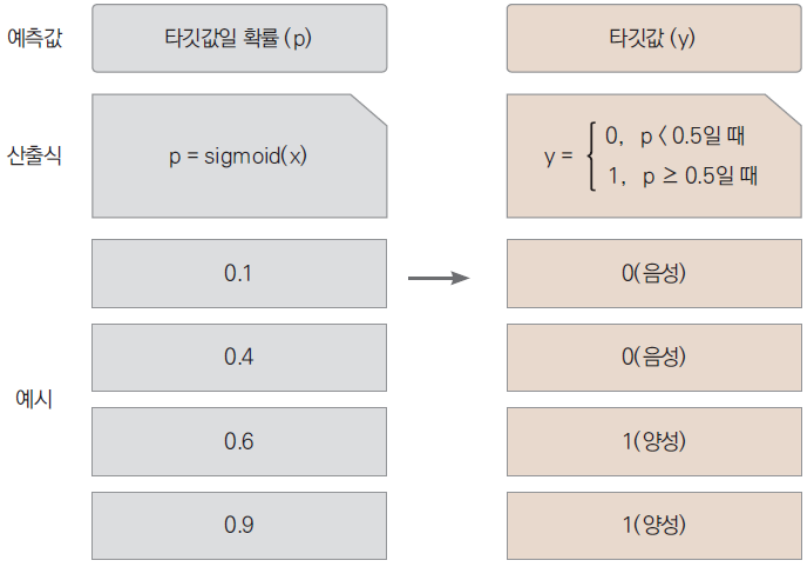
- **선형 회귀**(Linear Regression) : 선형 회귀식을 활용한 모델. 선형 회귀 모델을 훈련한다는 것은 훈련 데이터에 잘 맞는 모델 파라미터, 즉 회귀계수를 찾는 것이다.



5.6 주요 머신러닝 모델

5.6.2 로지스틱 회귀 모델

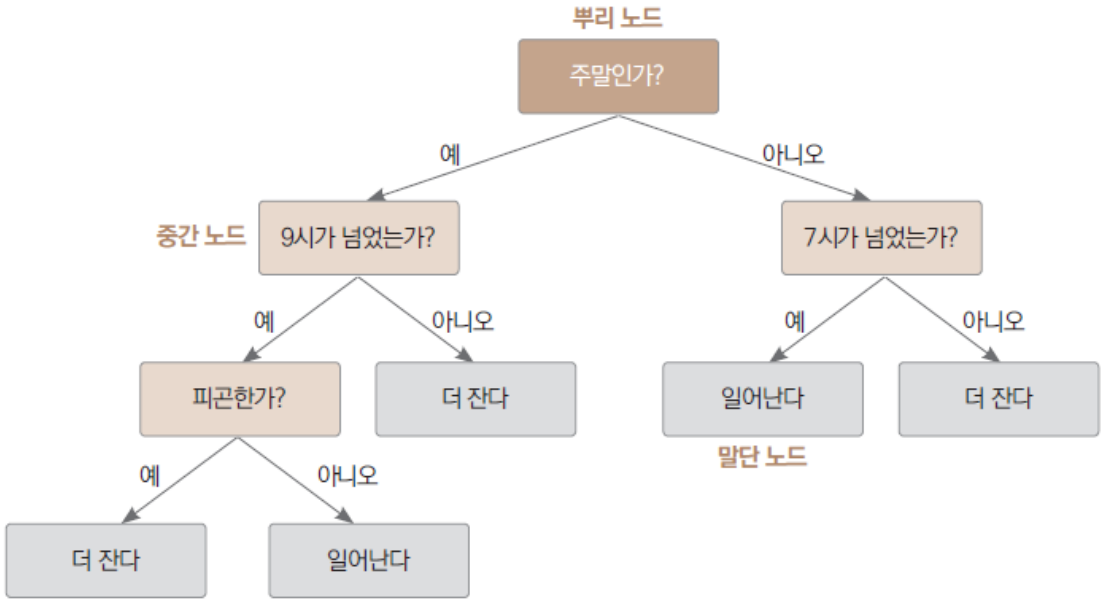
- **로지스틱 회귀**(Logistic Regression) : 선형 회귀 방식을 응용해 분류에 적용한 모델. 스팸 메일일 확률을 구하는 이진 분류 문제에 로지스틱 회귀를 사용할 수 있다.



5.6 주요 머신러닝 모델

5.6.3 결정 트리

- **결정 트리**(decision tree) : 분류와 회귀 문제에 모두 사용 가능한 모델. '의사결정 나무'라고도 한다.



5.6 주요 머신러닝 모델

5.6.3 결정 트리

- **불순도**(impurity) : 한 범주 안에 서로 다른 데이터가 얼마나 섞여 있는지 나타내는 정도
- **엔트로피**(entropy) : ‘불확실한 정도’
- **정보 이득**(information gain) : 1에서 엔트로피를 뺀 수치(1-엔트로피)
- **지니 불순도**(gini impurity) : 엔트로피와 비슷한 개념. 지니 불순도 값이 클수록 불순도가 높고 작을수록 불순도도 낮다.

5.6 주요 머신러닝 모델

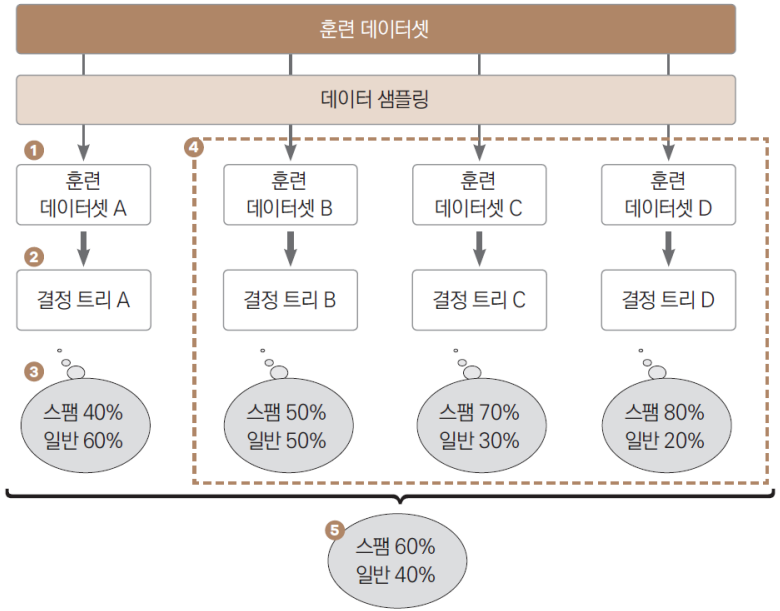
5.6.4 앙상블 학습(ensemble learning)

- 다양한 모델이 내린 예측 결과를 결합하는 기법. 앙상블 학습을 활용하면 대체로 예측 성능이 좋아진다. 과대적합 방지 효과도 있음.
- **보팅**(voting) : 서로 다른 예측 결과가 여러 개 있을 때 개별 결과를 종합해 최종 결과를 결정하는 방식
- **하드 보팅**(hard voting) : '다수결 투표' 방식으로 최종 예측값을 정하는 방식
- **소프트 보팅**(soft voting) : 개별 예측 확률들의 평균을 최종 예측확률로 정하는 방식
- **배깅**(bagging) : 개별 모델로 예측한 결과를 결합해 최종 예측을 정하는 기법. '개별 모델이 서로 다른 샘플링 데이터를 활용'한다는 점이 특징이다.
- **부스팅**(boosting) : 가중치를 활용해 분류 성능이 약한 모델을 강하게 만드는 기법

5.6 주요 머신러닝 모델

5.6.5 랜덤 포레스트(random forest)

- 결정 트리를 배깅 방식으로 결합한 모델. 나무(tree)가 모여 숲(forest)을 이루듯 결정 트리가 모여 랜덤 포레스트를 구성한다. 결정 트리와 마찬가지로 랜덤 포레스트도 분류와 회귀 문제에 모두 적용할 수 있다.



5.6 주요 머신러닝 모델

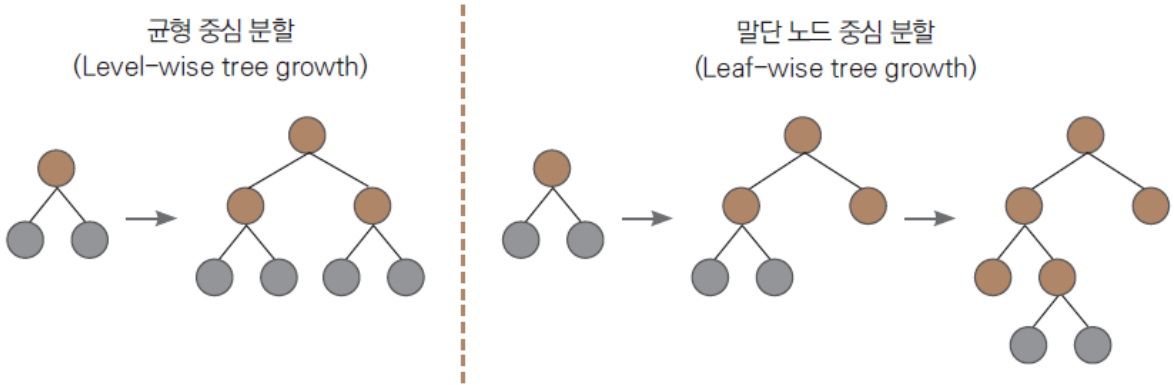
5.6.6 XGBoost

- 성능이 우수한 트리 기반 부스팅 알고리즘. 랜덤 포레스트는 결정 트리를 병렬로 배치하지만, XGBoost는 직렬로 배치해 사용한다. 즉 랜덤 포레스트는 배깅 방식, XGBoost는 부스팅 방식이다. XGBoost는 부스팅 방식이기 때문에 직전 트리가 예측한 값을 다음 트리가 활용해서 예측값을 조금씩 수정할 수 있다.

5.6 주요 머신러닝 모델

5.6.7 LightGBM

- XGBoost와 성능은 비슷하지만 훈련 속도가 더 빨라서 많은 캐글러가 가장 애용하는 머신러닝 모델. 마이크로소프트에서 개발했다.
- 말단 노드 중심으로 예측 오류를 최소화하게끔 분할한다. 말단 노드 중심으로 분할하면 균형을 유지할 필요가 없으니 추가 연산이 필요 없다. 균형 중심 분할보다 빠르다. 하지만 데이터 개수가 적을 때는 과대적합되기 쉽다는 단점이 있다.(과대적합 방지용 하이퍼파라미터를 조정해야 함)



5.7 하이퍼파라미터 최적화

하이퍼파라미터는 사용자가 직접 설정해야 하는 값이다. 모델이 좋은 성능을 내려면 어떤 하이퍼파라미터가 어떤 값을 가지면 좋을지를 찾아야 하며, 이를 하이퍼파라미터 최적화라고 한다.

5.7.1 그리드서치(grid search)

- 가장 기본적인 하이퍼파라미터 최적화 기법. 주어진 하이퍼파라미터를 모두 순회하며 가장 좋은 성능을 내는 값을 찾는다. 모든 경우의 수를 탐색하기 때문에 시간이 오래 걸린다.

5.7.2 랜덤서치(random search)

- 하이퍼파라미터를 무작위로 탐색해 가장 좋은 성능을 내는 값을 찾는 기법. 무작위라는 한계 때문에 그리드서치나 베이지안 최적화에 비해 사용 빈도가 떨어진다.

5.7 하이퍼파라미터 최적화

5.7.3 베이지안 최적화(bayesian optimization)

- 사전 정보를 바탕으로 최적 하이퍼파라미터 값을 확률적으로 추정하며 탐색하는 기법. 그리드 서치나 랜덤서치보다 최적 하이퍼파라미터를 더 빠르고 효율적으로 찾아준다. 코드도 직관적이며 사용하기 편리하다.