9주차

나이브 베이즈 모델

- 베이즈 정리를 적용한 조건부 확률 기반의 분류 모델
  - 스팸 필터링을 위한 대표적인 모델
  - 딥러닝보다 간단한 방법으로 자연어 처리를 원할 때

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

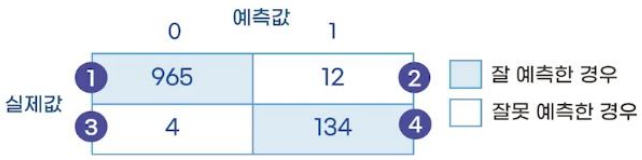
P(A|B)	• 사후확률 • B가 발생했을 때, A가 발생할 확률 스팸문자의 예로 B라는 특정 단어가 등장했을 때 A가 스팸일 확률
P(A)	• 사전확률 • B의 발생유무와 관련 없이 기본적으로 A가 발생할 확률 전체 문자 중 스팸문자의 비율

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

P(B|A)	• 우도 Likelihood 혹은 가능도라고도 부름 • A가 발생했을 때, B가 발생할 확률 스팸 메일인 경우 B라는 특정 단어가 들어 있을 확률
P(B)	• 전체에서 B가 발생할 확률 전체 문자에서 B라는 특정 단어가 들어 있을 확률

- 나이브 베이즈 수식은 사전확률로 사후확률을 예측하는 조건부 확률을 기반으로 함
  - 사후확률: 사건 A와 B가 있을 때, 사건 A가 발생한 상황에서 사건 B가 발생할 확률
  - 사전확률: 사건 A와 상관없이 사건 B가 발생할 확률
  - 베이즈 정리: 두 확률 변수의 사전 확률과 사후 확률 사이의 관계를 나타내는 정리, 사후확률을 구할 때 쓰임

- 장점
  - 비교적 간단한 알고리즘이다.
  - 속도가 빠르다.
  - 작은 훈련셋으로도 잘 예측한다.
- 단점
  - 모든 독립변수가 각각 독립적임을 전제로 하는데 이는 장점이 될 수도 있으나 단점이 되기도 한다. → 실제 그렇지 X
  - 실제로 모든 변수들이 독립적이라면 다른 알고리즘보다 우수할 수 있으나, 실제 데이터에서는 그런 경우가 많지 않기 때문에 단점이기도 하다.
- 실습단계
  - 1단계: 문제 정의
  - 2단계: 라이브러리 및 데이터 불러오기, 데이터 확인하기
  - 전처리 단계
    - 3단계: 특수기호 제거
    - 4단계: 불용어 제거
    - 5단계: 목표 컬럼 형태 변경
    - 6단계: 카운트 기반으로 벡터화
  - 7단계: 모델링 예측



- 이 매트릭스의 값들을 단순 산술해도 정확도를 구할 수 있음

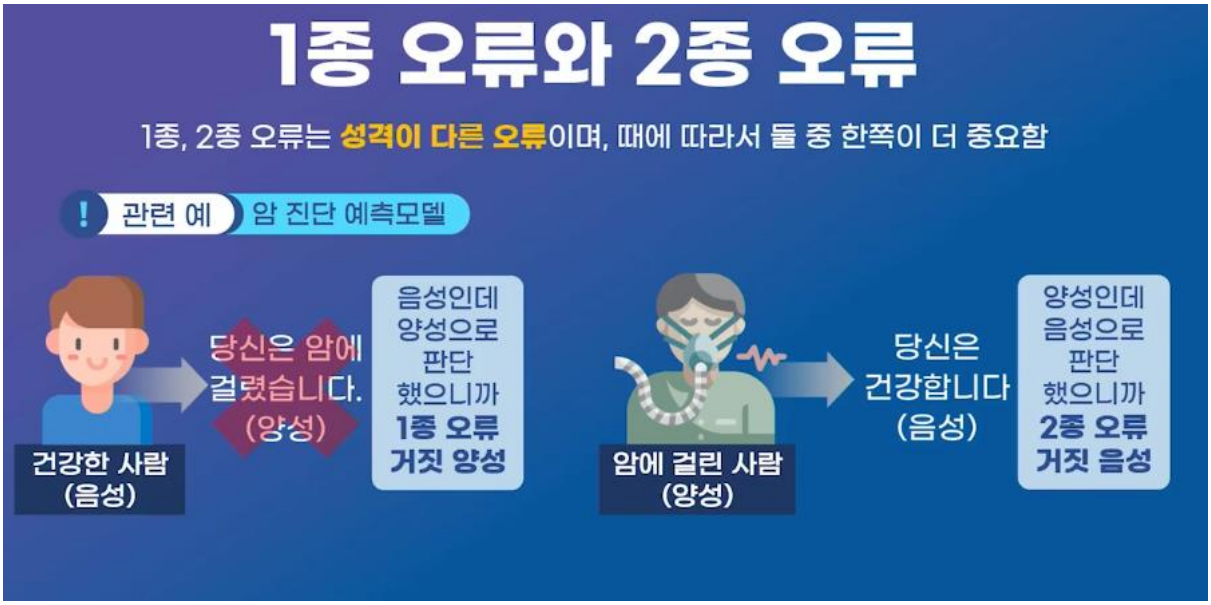
$$\frac{\text{정확한 예측 건수}}{\text{전체 경우의 수}} \Rightarrow \frac{965+134}{965+134+12+4} \fallingdotseq 98.9\%$$

		예측값	
		0	1
실제값	0	True Negative (TN) 음성을 음성으로 판단	False Positive (FP) 음성을 양성으로 판단
	1	False Negative (FN) 양성을 음성으로 판단	True Positive (TP) 양성을 양성으로 판단

1종오류

2종오류

		예측값	
		음성	양성
실제값	음성	정확함 (Correct)	1종 오류 (Type 1 Error)
	양성	2종 오류 (Type 1 Error)	정확함 (Correct)

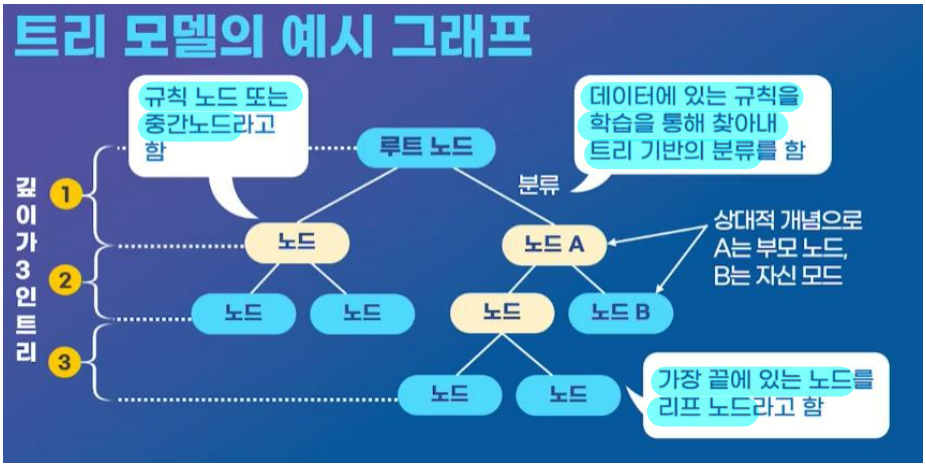


나이브 베이즈.

10주차

결정트리

- 관측값과 목표값을 연결시켜주는 예측 모델
  - 결정트리
    - 선형모델: 각 변수에 대한 기울기 값들을 최적화하여 모델을 만들어 나감
    - 트리모델: 각 변수의 특정 지점을 기준으로 데이터를 분류해가면서 예측 모델을 만듦
      - 예시: 남자/여자로 나눠 각 목표값 평균치를 나눈다.
      - 나이를 30세 이상/미만으로 나눠 평균치를 계산하는 방식으로 데이터를 무수히 쪼개 나가고, 각 그룹에 대한 예측치를 만들어낸다.



- 트리 모델의 장점
  - Non-parametric Model (데이터에 대한 가정이 없는 모델)
    - 예시
      - 선형 모델: 정규분포에 대한 가정이나 독립변수와 종속변수의 선형 관계 등을 가정으로 하는 모델
      - 결정 트리: 데이터에 대한 가정이 없으므로 어디에나 자유롭게 적용할 수 있다.
  - 아웃라이어에 거의 영향을 받지 않음
  - 시각화에 굉장히 탁월하다.
    - 트리 그래프를 통해 직관적으로 이해하고 설명 가능

- 트리 모델의 단점
  - 오버 피팅
    - 트리가 무한정 깊어지면 오버피팅 문제를 야기할 수 있음
  - 예측력이 떨어짐
    - 발전된 트리 기반 모델들에 비하면 예측력이 상당히 떨어진다
- 결정트리: 특정 변수에 대한 특정 기준값으로 데이터를 계속 분류해가면서 유사한 그룹으로 묶어내어 예측값을 만드는 알고리즘
- 결정트리 이해
  - 분류 과정의 포인트
    - 첫 번째 분류 과정: 사용할 변수 선정과 기준점을 정하는 것이 매우 중요
    - 두 번째 분류 과정: 각 상황에서 최적의 변수와 기준점을 찾아내야 함
  - 분류 결정 트리
    - Decision Tree Classifier는 각 노드의 순도가 가장 높은 방향으로 분류한다.
      - 노드: 결정트리 다이어그램에서 보이는 각 박스
      - 순도: 한 노드 안에 여러 종류가 아닌 한 종류의 목표값만 있는 상태에 대한 지표
    - 지니 인덱스와 교차 엔트로피를 평가하여 분류한다

지니 인덱스	$1-\sum_{i=1}^{n}(pi)^2$ 낮을수록 순도↑

      - 각 노드에 대해 지니 인덱스가 계산된다.
        - P: 노드 안에 특정 아이템의 비율

사과 2, 복숭아 2인 경우 : 지니 인덱스	$1 - [0.5^2 + 0.5^2] = 0.5$
사과 1, 복숭아 3인 경우 : 지니 인덱스	$1 - [0.25^2 + 0.75^2] = 0.375$
사과 0, 복숭아 4인 경우 : 지니 인덱스	$1 - [0^2 + 1^2] = 0$

➡ 노드가 한쪽 아이템으로 완전히 분류된 경우 지니 인덱스가 0이며, 순도가 높다고 평가함

지니 인덱스

예시	비율	$\sum_{i=1}^{n}(pi)^2$	지니 인덱스 $1-\sum_{i=1}^{n}(pi)^2$
사과 2개, 복숭아 2개	각각 50%	$0.5^2 + 0.5^2 = 0.5$	1 - 0.50이므로 0.5
사과 1개, 복숭아 3개	25%, 75%	$0.25^2 + 0.75^2 = 0.625$	1 - 0.6250이므로 0.375
사과 0개, 복숭아 4개	0%, 100%	$0^2 + 1^2 = 1$	1 - 1이므로 0

➡ 지니 인덱스의 최댓값은 0.5, 최솟값은 0이 나올 수 있음

➡ 사이킷런의 결정 트리에서 분류를 위한 기본 지수가 지니 인덱스임

교차 엔트로피

예시	비율	중간 계산 $\sum_{i=1}^{n}P_i \times log_2(Pi)$	교차 엔트로피 $-\sum_{i=1}^{n}P_i \times log_2(Pi)$
사과 2개, 복숭아 2개	각각 50%	• 사과 50% : 0.5 x $Log_2(0.5)$ = 0.5 x -1 = -0.5 • 복숭아 50% : 0.5 x $Log_2(0.5)$ = 0.5 x -1 = -0.5 • 시그마 위 두값 대한 합 : -0.5 + -0.5 = -1	1
사과 1개, 복숭아 3개	25%, 75%	• 사과 25% : 0.25 x $Log_2(0.25)$ = 0.25 x -2 = -0.5 • 복숭아 75% : 0.75 x $Log_2(0.75)$ = 0.75 x -0.415037⋯≈ -0.31 • 시그마 위 두값 대한 합 : -0.5 + -0.5 = -1	약 0.81
사과 0개, 복숭아 4개	0%, 100%	• 사과 0% : 0 x $Log_2(0)$ = 0 • 복숭아 100% : 1 x $Log_2(1)$ = 1 x 0 = 0 • 시그마 위 두값 대한 합 : 0 + 0 = 0	0

➔ 교차 엔트로피의 최대값은 1, 최소값은 0이 나올 수 있다.

➔ 교차 엔트로피를 지니 인덱스 대신 활용할 수 있다.

■ 결정 트리의 활용 분야

◆ 종속 변수가 연속형 데이터와 범주형 데이터 모두에 사용할 수 있다.

◆ 모델링 결과를 시각화 할 목적으로 가장 유용하다.

◆ 아웃라이어가 문제될 정도로 많을 때 선형 모델보다 좋은 대안이 될 수 있다.

- 회귀 결정 트리
  - 회귀는 연속형 변수를 대상으로 MSE를 평가 기준으로 이용함
  - MSE를 구하며, 결정 트리 회귀는 가장 낮은 MSE값이 나오도록 노드를 분류해 나감
  - MSE는 사이킷런의 결정 트리 모델에서 기본값으로 설정된 평가 기준임.
  - 필요에 따라 매개변수를 이용하여 MSE 대신 MAE나 Poisson 등으로 설정할 수도 있음
  - MSE: 실제 값과 예측 값의 차이에 대한 계산

- 결정 트리 특징
  - 예측력과 설명력 (반비례 관계)
    - 예측력: 모델 학습을 통해 얼마나 좋은 예측치를 보여주는 지를 의미
    - 설명력: 학습된 모델을 얼마나 쉽게 해석할 수 있는지를 의미
    - 단순한 알고리즘일수록 예측력이 상대적으로 떨어질 수 있으나 해석이 용이함
    - 복잡한 알고리즘일수록 예측력이 뛰어난만큼 해석은 어려움
      - 예시: 결정트리/회귀분석 → 상대적으로 해석이 쉬워 설명력 높음

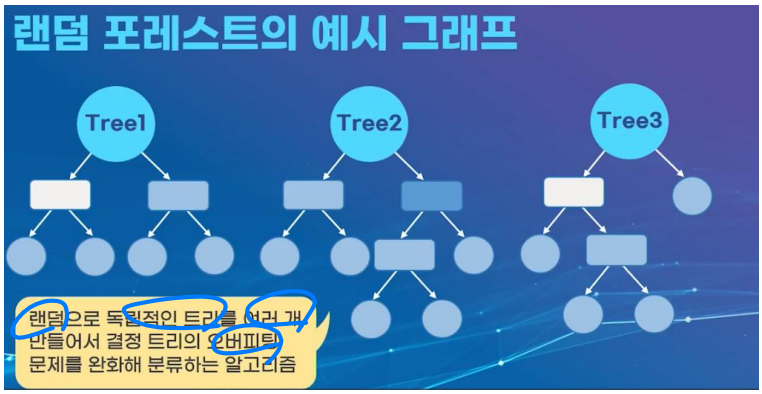
        앙상블 기법/인공신경망 → 예측력은 높지만 해석이 어려움
  - 상황에 따라 예측력/설명력 중 선택 문제가 발생
    - 예시 1: 특정 질병의 발병률에 대한 예측모델
      - 발병률을 높이거나 억제하는 중요한 요인을 밝히는 설명력이 좋은 알고리즘이 적합
    - 사기거래 예측모델
      - 요인보다는 더 정확하게 사기거래를 잡아낼 수 있어야 하므로 예측력이 높은 알고리즘이 더 적합

나비→결론/전로.

11주차

랜덤 포레스트

- 결정 트리의 단점인 오버피팅 문제를 완화시켜주는 발전된 형태의 트리 모델
  - 무수히 많은 트리를 이용해 예측에 활용한다.
  - 여러 모델을 활용해서 하나의 모델을 이루는 기법으로 앙상블이라 한다.



  - 장점
    - 아웃라이어에 거의 영향을 받지 않음
      - 결정 트리와 마찬가지임
    - 별다른 가정 없이 잘 작동함
      - 선형/비선형 데이터에 상관없이 잘 작동함
    - 다양한 트리의 의견을 반영해 오버피팅의 위험을 낮춤
      - 랜덤 포레스트가 여러 개의 트리를 만들 때는 데이터 전체를 사용하지 않고, 매번 다른 일부의 데이터를 사용하여 다른 트리를 만들어 냄.
      - 각 트리에서 주어진 모든 독립변수를 사용하지 않고 일부 변수들만을 매번 다르게 추출하여 사용함
    - 특정 변수의 강력한 힘을 제어하고 오버피팅을 피할 수 있다.

    - 단점
        - 학습 속도가 상대적으로 느림
        - 모델에 대한 해석이 어려움
            - 수많은 트리를 동원하기 때문에 모델에 대한 해석이 어려움
    - 주의 사항
        - 앙상블 기법을 사용한 트리 기반 모델 중 가장 보편적이나, 이후에 다루게 될 부스팅 모델에 비하면 예측력이나 속도에서 부족한 부분이 존재.
- 실습단계
    - 1단계: 문제 정의
    - 2단계: 라이브러리 및 데이터 불러오기/데이터 확인하기
    - 전처리 단계:
        - 3단계: 텍스트 데이터
        - 4단계: 결측치 처리와 더미 변수 변환
    - 5단계: 모델링 예측하기
    - 이해 단계:
        - 6단계: K겹 교차검증
        - 7단계: 랜덤 포레스트
    - 8단계: 하이퍼파라미터 튜닝
- 랜덤 포레스트 해석
    - K겹 교차검증
        - 교차 검증의 목적: 교차 타당성
            - 모델의 예측력을 더 안정적으로 평가하기 위함.
            - 새로운 데이터를 얼마나 잘 예측하는지 확인하고자 시험셋을 나누어 평가
            - 훈련셋과 시험셋의 데이터를 랜덤하게 분할하여 어느정도 안정성을 보장

■ K겹 교차검증의 아이디어

■ 하이퍼파라미터 튜닝 (주 사용 매개변수)

◆ n_estimators

- 랜덤 포레스트를 구성하는 결정트리 개수
- 기본값은 100으로 설정되어 있음
- 너무 많거나 적은 수를 입력하면 성능이 떨어지므로 적정 수준의 값을 찾아 넣어야함

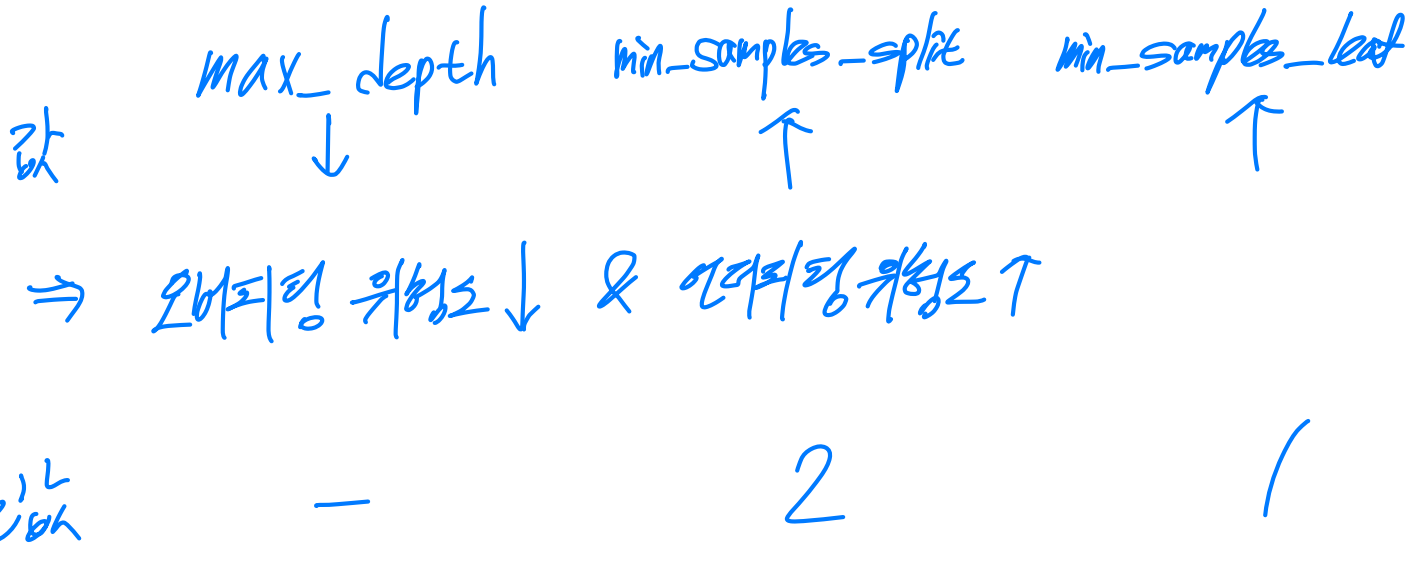
◆ max_depth

- 결정트리와 동일하게, 각 트리의 최대 깊이를 제한함
- 숫자가 낮을수록 오버피팅을 피할 수 있으며, 언더피팅의 위험은 올라감

◆ min_samples_split

- 해당 노드를 나눌 것인지 말 것인지를 노드 데이터 수로 판단
- 해당 매개변수에 지정된 숫자보다 적은 데이터 수가 노드에 있으면 더는 분류하지 않음
- 숫자가 높을수록 분리되는 노드가 적어질 것이므로 오버피팅을 피하는 방법이자 언더피팅의 위험도 있음
- 기본값은 2임

- min_samples_leaf
  - 분리된 노드의 데이터에 최소 몇 개의 데이터가 있어야 할 지를 결정하는 매개변수임
  - 여기에 지정된 숫자보다 적은 수의 데이터가 분류된다면, 해당 분리는 이루어지지 않음
  - 숫자가 클수록 오버피팅을 피할 수 있고, 언더피팅의 위험도는 높아짐
  - 기본값은 1임
- n_jobs
  - 병렬 처리에 사용되는 cpu 코어 수임
  - 많은 코어를 사용할수록 속도가 빨라지며 -1을 입력 시 지원하는 모든 코어를 사용함
  - 기본값은 None, 실제는 1개의 코어를 사용함
  - 랜덤 포레스트의 속도가 다소 느린만큼 충분한 코어를 사용하는게 좋다

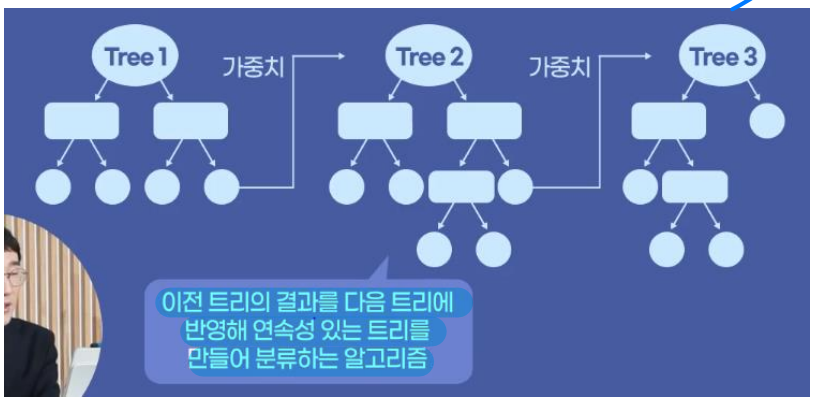


나이/경트/2주/XGBoost

12주차

XGBoost

- 부스팅: 순차적으로 트리를 만들어 이전 트리로부터 더 나은 트리를 만드는 알고리즘
  - 트리모델을 기반으로 한 최신 알고리즘
  - 빠른 속도와 높은 예측력으로 랜덤 포레스트를 능가함
  - 우수한 예측 능력과 빠른 처리 속도로 실제 산업 현장에서는 굉장히 중요한 가치
  - 대표적인 알고리즘으로 XG부스트, 라이트 GBM, 캣부스트가 있음
  - XG부스트 (eXtreme Gradient Boost)
    - 가장 먼저 개발되었다.
    - 가장 널리 활용된다.
    - 손실함수뿐만 아니라 모형 복잡도까지 고려한다.
    - 2차 도함수 활용과 정규화 하이퍼파라미터 지원이라는 특징이 있다
    - 각 트리가 독립적인 랜덤 포레스트와 달리 이전 트리를 기반으로 생성.



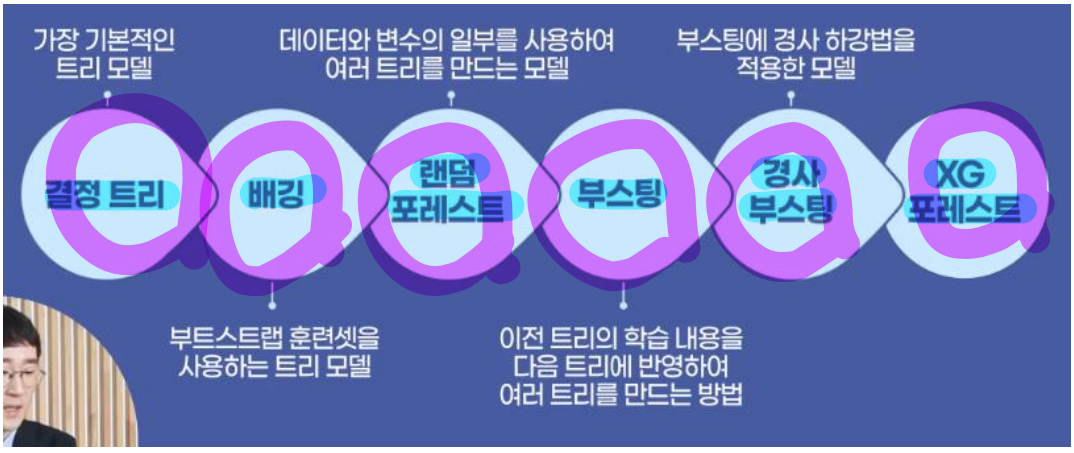
    - 장점
      - 예측 속도가 빠름
      - 예측력 높음
      - 변수 종류가 많고 데이터가 클수록 상대적으로 뛰어난 성능을 보임
    - 단점
      - 복잡한 모델인만큼, 해석에 어려움이 있다
      - 더 나은 성능을 위한 하이퍼파라미터 튜닝이 까다롭다

◆ 유용한 곳

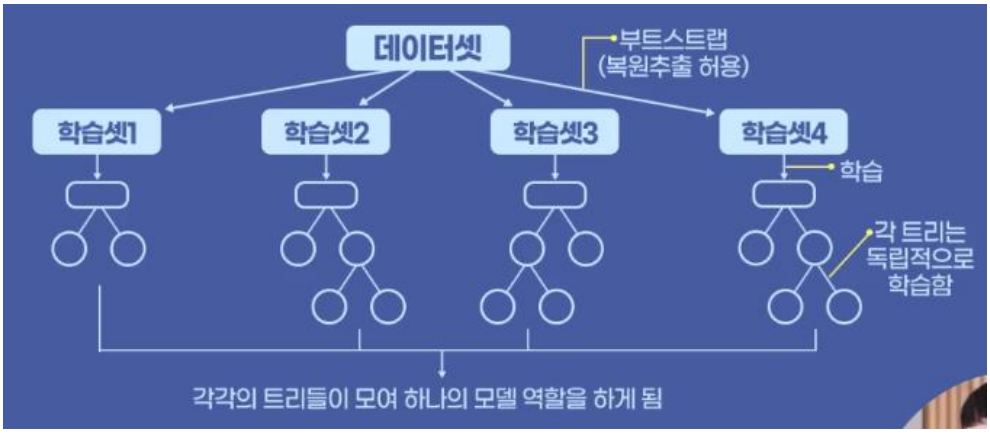
- 다양한 데이터 모두 사용 가능
  - 종속 변수가 연속형 데이터, 범주형 데이터인 경우 모두 사용 가능
- 거의 모든 상황에 사용 가능
  - 이미지나 자연어가 아닌 표로 정리된 정형 데이터의 경우, 거의 모든 상황에 활용할 수 있음.

- XG부스트 실습
  - 1단계: 문제정의
  - 2단계: 라이브러리 및 데이터 불러오기
  - 전처리 단계:
    - 3단계: 데이터 클리닝
    - 4단계: 피처 엔지니어링
  - 5단계: 모델링 및 예측하기
  - 6단계: 이해하기(경사 하강법)
  - 7단계: 하이퍼파라미터 튜닝 – 그리드 서치
  - 8단계: 중요변수 확인

- 트리 모델의 진화 과정

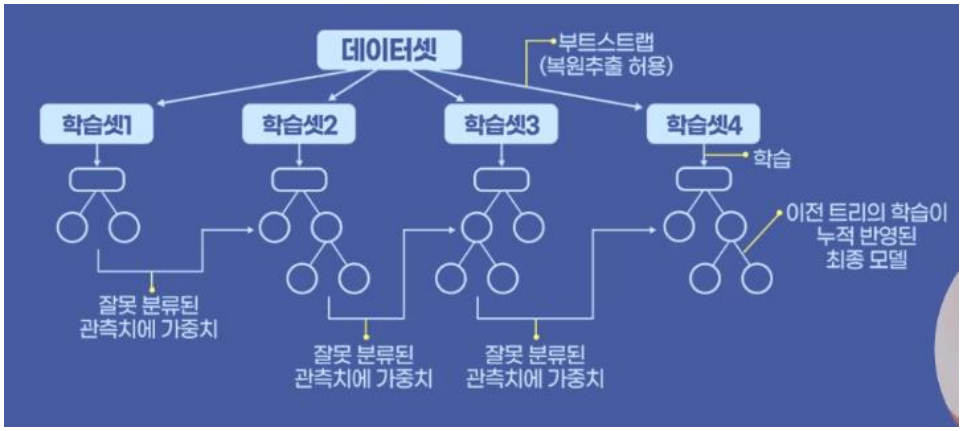


- 배깅 학습 방법



  - 배깅: 부트스트랩 훈련셋을 사용하는 트리 모델
    - 부트스트랩: 데이터의 일부분을 무작위로 반복 추출하는 방법
    - 추출한 데이터의 여러 부분집합을 사용해 여러 트리를 만들어 오버피팅을 방지함.
    - 랜덤 포레스트는 배깅에서 한단계 더 발전된 모델임

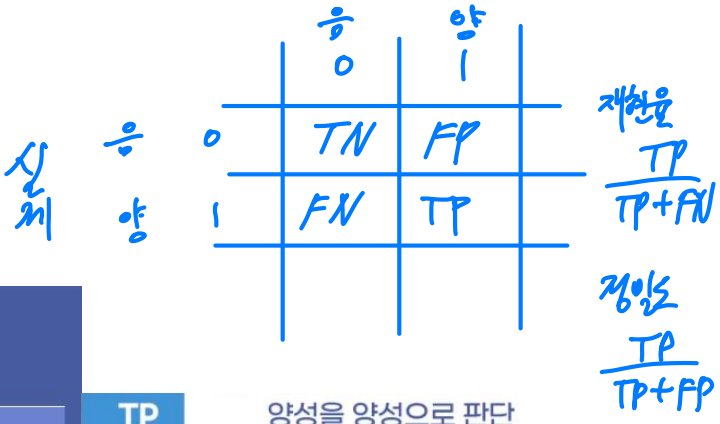
- 에이다부스트(Adaptive Boosting)
  - 단계적으로 트리를 만들 때 이전 단계에서 분류 결과에 따라 각 데이터에 가중치를 부여/수정함
  - 예시
    - 이전 트리에서 가중치가 덜 부여되고 잘못 분류된 데이터들에 더 높은 가중치를 부여함
    - 후속 트리에서는 가중치가 높은 데이터를 분류하는데 우선 순위를 둠



- 경사부스팅
  - 경사하강법을 사용하여 이전 모델의 에러를 기반으로 다음 트리를 만든다.
    - 구현 알고리즘: XGBoost, LightGBM, CatBoost
- XGBoost의 핵심 용어
  - 부스팅
    - 랜덤 포레스트에서 그 다음 세대로 진화하게 되는 중요한 개념임
    - 랜덤 포레스트에서는 각각의 트리를 독립적으로, 서로 관련 없이 만듦
  - 부스팅 알고리즘
    - 부스팅 알고리즘에서는 트리를 순차적으로 만들면서 이전 트리에서 학습한 내용이 다음 트리를 만들 때 반영됨
  - 경사하강법
    - 경사 부스팅의 핵심 개념, 모델이 어떻게 최소 오차가 되는 매개 변수들을 학습하는지에 대한 방법
    - 오차식에 대한 미분계수를 통해 매개변수의 이동 방향과 보폭을 결정한다.
    - 보폭은 learning rate라는 하이퍼파라미터로 조절할 수 있음

- 모델링 및 평가

■ 혼동 행렬



혼동행렬

		예측값 0	예측값 1
실제값	0	1303 (TN)	62 (FP)
	1	150 (FN)	111 (TP)

TP	양성을 양성으로 판단
FN	양성을 음성으로 판단
FP	음성을 양성으로 판단
TN	음성을 음성으로 판단

■ 정밀도

양성을 양성 / 양성을 양성 + 음성을 양성

- 1로 예측한 경우 중, 얼마만큼이 실제로 1인지를 나타냄

$$\frac{TP}{TP+FP} = \frac{\text{양성을 양성으로 판단}}{\text{양성을 양성으로 판단} + \text{1종 오류}} = \frac{\text{양성을 양성으로 판단}}{\text{양성으로 판단한 수}}$$

- FP가 커질수록 분모가 커지므로, 정밀도는 낮아짐
- 1종 오류와 관련됨

■ 재현율

양성을 양성 / 양성을 양성 + 양성을 음성

- 실제로 1 중에, 얼마만큼을 1로 예측했는지를 나타냄

$$\frac{TP}{TP+FN} = \frac{\text{양성을 양성으로 판단}}{\text{양성을 양성으로 판단} + \text{2종 오류}} = \frac{\text{양성을 양성으로 판단}}{\text{실제로 양성인 수}}$$

- FN이 커질수록 recall 값이 작아짐
- 2종 오류와 관련됨

■ F1-Score

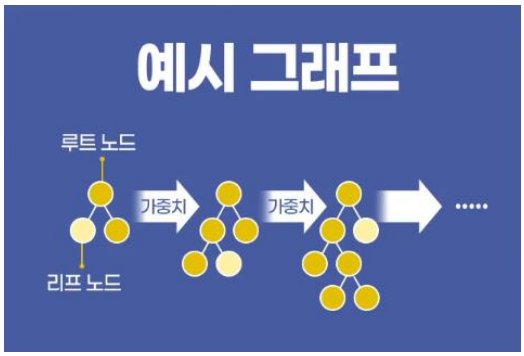
- 정밀도와 재현율의 조화평균

$$2 \times \frac{precision \times recall}{precision + recall} = 2 \times \frac{\text{정밀도} \times \text{재현율}}{\text{정밀도} + \text{재현율}}$$

- 정밀도와 재현율이 높을 때 함께 높아지며, 둘의 값이 비슷할수록 더 높은 값을 보임
- 1종 오류가 중요하다면 정밀도에, 2종 오류가 중요하다면 재현율에 더욱 신경써야 함
- 특별히 더 중요한 오류 유형이 없다면 F1-점수를 검토하는 것이 무난함

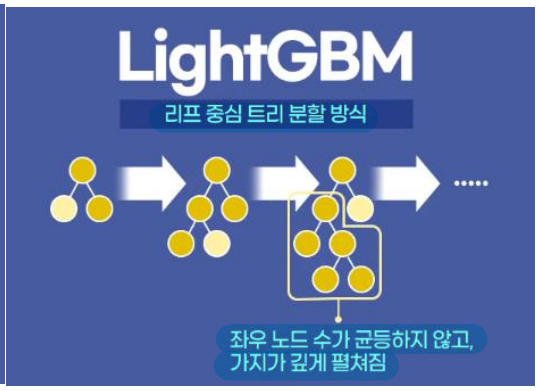
13주차

- LightGBM
  - XGBoost 이후로 나온 최신 부스팅 모델로 출시 전에는 XGBoost가 강세였으나, 출시 후는 라이트 GBM이 강세를 보임



  - 장점
    - XGBoost보다도 빠르고 높은 정확도를 보여주는 경우가 많다
    - 예측에 영향을 미친 변수의 중요도를 확인할 수 있다.
    - 변수 종류가 많고 데이터가 클수록 상대적으로 뛰어난 성능을 보인다.
  - 단점
    - 복잡한 모델인 만큼, 해석에 어려움이 있다
    - 더 나은 성능을 위한 하이퍼파라미터 튜닝이 까다롭다.
  - 유용한 곳
    - 정형 데이터 – 캣부스트, LightGBM, XGBoost
  - XGBoost와 비교
    - 빠른 학습 및 예측
    - 더 적은 메모리 사용
    - 데이터 셋 자동 변환 및 최적 분할

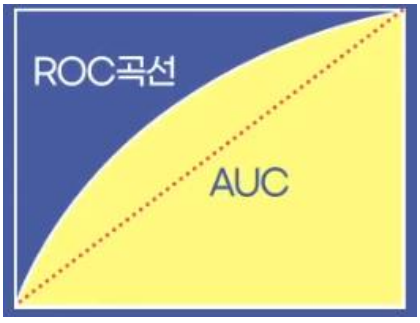
L, 경희
→ 인식처리
⇒ 피처선택



- LightGBM 핵심
  - XGBoost보다 한단계 더 진화한 형태로 리프 중심 트리 분할을 사용하여 더 빠르고 정확한 예측을 보여준다
  - 리프 중심 트리 분할
    - XGBoost와 LightGBM의 중요한 차이점이다.
    - 동일한 레벨로 노드를 확장하지 않고 불규칙적으로 노드를 뻗어 나가기 때문에 더 빠르고 높은 예측율을 보이나 오버피팅에 주의해야한다.

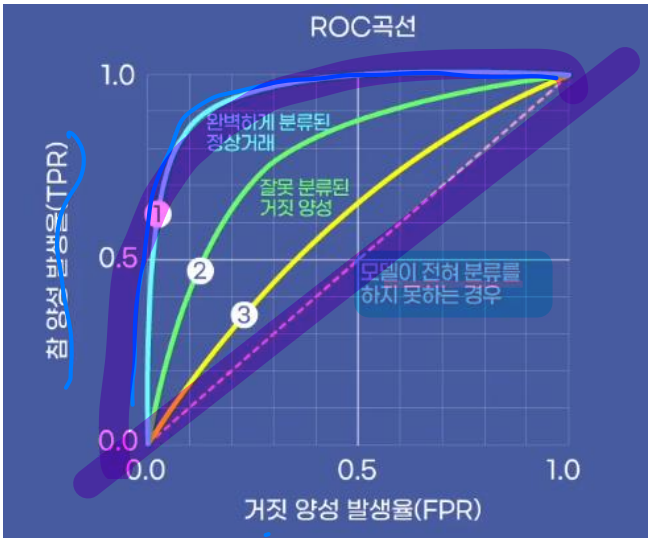
- 실습단계
  - 1단계: 문제정의
  - 2단계: 라이브러리 및 데이터 불러오기
  - 전처리 단계:
    - 3단계: 데이터 클리닝
    - 4단계: 피처 엔지니어링
  - 5단계: 모델링 및 평가하기

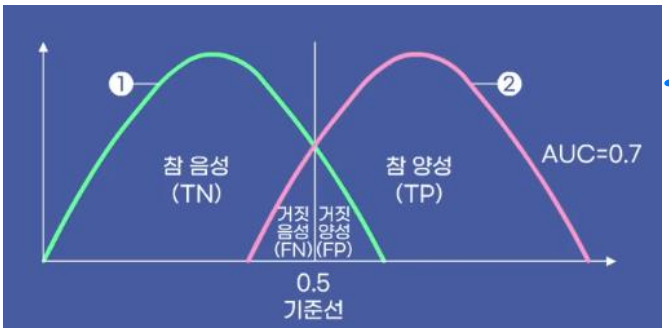
- AUC (Area Under the Curve)
  - ROC 곡선: ROC곡선의 ROC 커브 아래쪽 면적

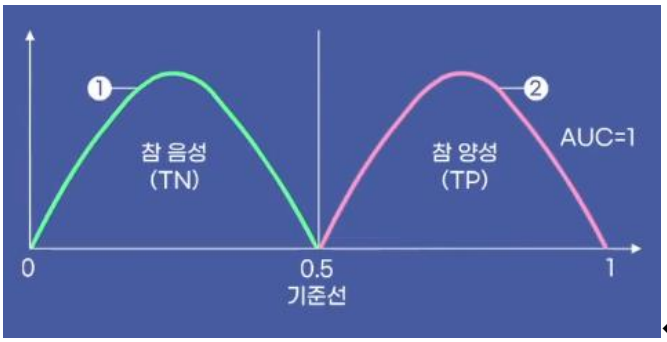


■ ROC 곡선은 민감도와 특이도 개념을 통해 만들어진다.

민감도	$TPR = \frac{TP(참\ 양성)}{TP(참\ 양성) + FN(거짓\ 음성)}$
특이도	$FPR = \frac{FP(거짓\ 양성)}{FP(거짓\ 양성) + TN(참\ 음성)}$







← 아주 완벽한 예측 모델

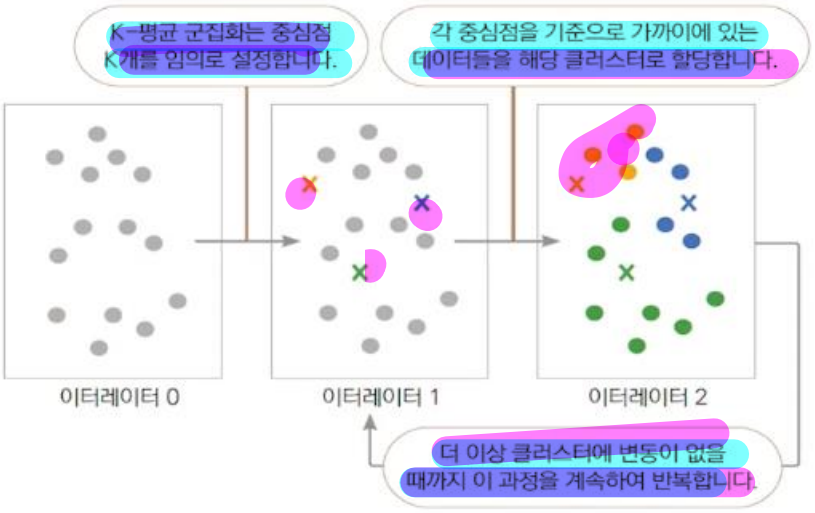
■ AUC

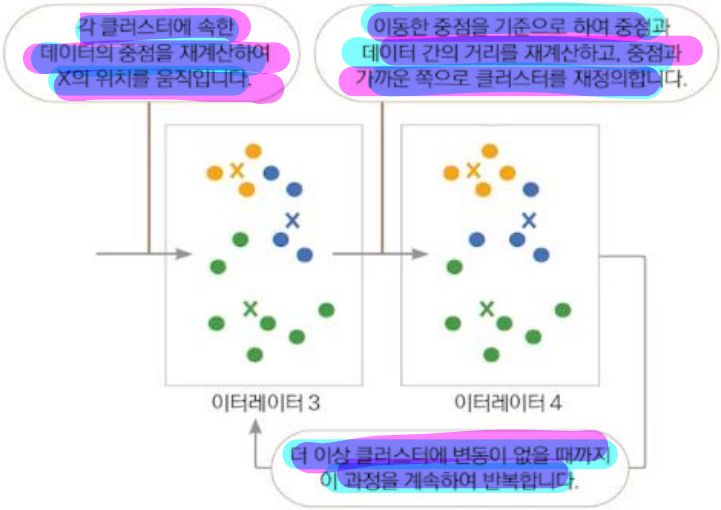
◆ 여러 모델 비교 → 적합한 객관적 지표

14주차

K 평균 군집화

- K-평균 군집화
  - ■ 비지도 학습의 대표적인 알고리즘
    - ◆ 목표 변수가 없는 상태에서 데이터를 비슷한 유형끼리 묶는 머신러닝 기법
    - ◆ K 최근접 이웃 알고리즘과 비슷하게 거리 기반으로 작동하며 적절한 K 값을 사용자가 지정해야한다
    - ◆ 거리 기반으로 작동하기 때문에 데이터 위치가 가까운 데이터끼리 한 그룹으로 묶음
    - ◆ 이때 전체 그룹의 수는 사용자가 지정한 K이다.

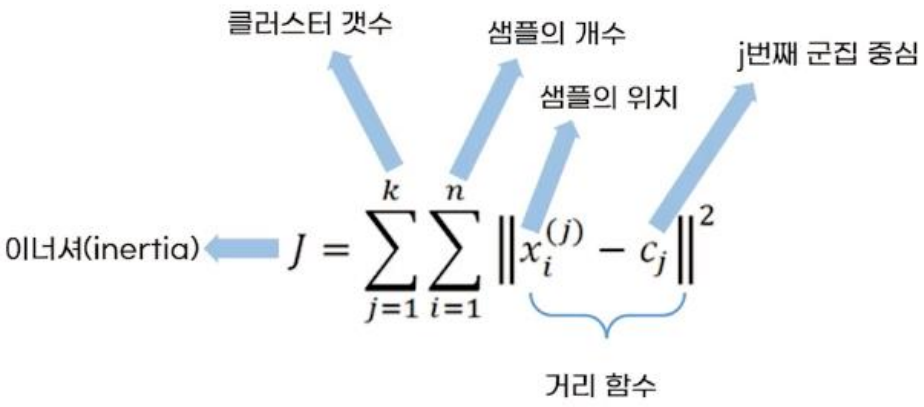




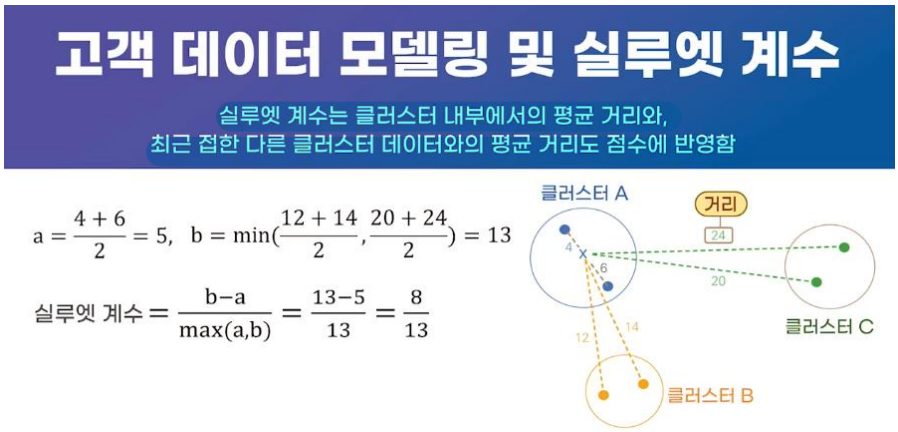
- 장점
  - 구현이 비교적 간단하다
  - 클러스터링 결과를 쉽게 해석할 수 있다.
- 단점
  - 사용자가 직접 선택해야 한다.
    - 최적 K값을 자동으로 찾지 못해 사용자가 직접 선택해야 함
  - 거리 기반 알고리즘
    - 변수의 스케일에 따라 다른 결과를 나타낼 수 있음

- 실습단계
  - 1단계: 문제정의
  - 2단계: K 평균 군집화 맛보기(라이브러리 및 데이터 불러오기, 연습용 데이터 모델링 및 평가, 엘보우 기법으로 최적 K값 구하기)
  - 3단계: 데이터 불러오기 및 데이터 확인
  - 4단계: 전처리: 피처 엔지니어링
  - 5단계: 고객 데이터 모델링 및 실루엣 계수
  - 6단계: 최종 예측 모델 및 결과 해석

- 이너셔
  - 각 샘플과 가장 가까운 군집 중심 사이의 평균 제곱 거리를 측정한 수치



- K평균 군집화 핵심용어
  - 이너셔
    - 각 클러스터의 중점과 그에 속한 데이터 간의 거리. 값이 작을수록 잘 뭉쳐진 클러스터임
  - 실루엣 계수
    - 엘보우 기법과 같이 최적의 클러스터 수를 찾는 방법으로, 엘보우 기법에서 적절한 클러스터 수를 찾지 못했을 때 대안으로 사용할 수 있음
    - 엘보우 기법보다 계산시간이 오래 걸리는 단점이 있음
  - 엘보우 기법
    - 최적의 클러스터 개수를 확인하는 방법
    - 클러스터의 중점과 각 데이터 간의 거리를 기반으로 계산함



15주차

주성분 분석(PCA)

- PCA 목적
  - 데이터 차원(변수 개수)를 축소함
    - 차원 축소를 통해 변수의 개수를 줄이고 정보를 많이 없앰
  - 변수의 개수를 줄이지만 가능한 그 특성을 최대한 보존
    - 기존 변수들의 정보를 모두 반영하는 새로운 변수를 주로 만든다
      - 새로운 변수를 만들어 차원을 축소하는 방법
- 장점
  - 다차원을 2차원에 적합하도록 차원 축소하여 시각화에 유용하다
  - 변수 간의 높은 상관 관계 문제를 해결해준다.
- 단점
  - 기존 변수가 아닌 새로운 변수를 사용하여 해석하는 데 어려움이 있다.
  - 차원이 축소됨에 따라 정보 손실이 불가피함
- 유용한 곳
  - 다차원 변수들을 2차원 그래프로 표현할 경우
  - 변수가 너무 많아 모델 학습에 시간이 너무 오래 걸릴 경우
    - 차원 축소를 진행하면 연산의 시간을 줄일 수 있다
  - 오버피팅을 방지하는 용도
- 주성분 분석의 이해
  - 선형대수, 역행렬 계산, 주성분 계산
  - 3차원을 2차원으로 자원 축소할 때 투영하는 방향에 따라 간격이 실제보다 가깝거나 겹쳐 보일 수 있으므로 적합한 투영 방향을 결정해야 함
  - 2차원을 1차원으로 자원 축소할 때 또한 투영하는 방향에 따라 점이 겹치거나 간격이 비슷해질 수 있으므로, 점이 완전히 겹치지 않고 거리감도 있는 편이 중요

- 핵심용어
  - 차원 축소
    - 변수 2개면 2차원 그래프, 3개면 3차원 그래프로 나타냄
    - 데이터의 차원 = 변수의 개수
    - 차원 축소는 변수의 수를 줄여 데이터의 차원을 축소함

9주차

미션	스팸 데이터셋을 이용하여 스팸여부를 판단하라.
난이도	★☆☆
알고리즘	나이브 베이즈(Naïve Bayes)
데이터셋 파일명	spam.csv
종속변수	target(스팸 여부)
데이터셋 소개	• 스팸 문자에 대한 데이터로, 독립변수는 text 하나밖에 없음 • 그러나 이 하나의 변수에 긴 문장 형태의 데이터들이 들어 있기 때문에 많은 전처리 작업이 필요함 • 각 문장에 들어간 단어들을 활용하여 문자가 스팸인지 아닌지를 예측하게 됨
문제유형	분류
평가지표	정확도, 혼동 행렬
사용한 모델	MultinomialNB

10주차

미션	학력, 교육, 연수, 혼인 상태, 직업 정보를 담은 연봉 데이터셋을 이용해 연봉을 예측하라.
난이도	★☆☆
알고리즘	결정 트리(Decision Tree)
데이터셋 파일명	salary.csv
종속변수	class(연봉 등급)
데이터셋 소개	• 연봉 데이터를 사용함 • 연봉이 $50,000 이상인지 이하인지를 예측하는 것이 목표임 • 종속변수는 class, 독립변수로는 학력, 교육 연수, 혼인 상태, 직업 등이 있음
문제유형	분류
평가지표	정확도
사용한 모델	DecisionTreeClassifier

11주차

미션	중고차 판매 이력 데이터셋을 이용해 중고차 가격을 예측하라.
난이도	★☆☆
알고리즘	랜덤 포레스트(Random Forest)
데이터셋 파일명	car.csv
종속변수	selling_price(판매 가격)
데이터셋 소개	• 중고차 판매 이력을 다룬 데이터 • 종속변수는 판매 가격임 • 독립변수는 생산년도, 주행거리, 변속기, 마일리지, 배기량 등이 있음
문제유형	회귀
평가지표	RMSE
사용한 모델	RandomForestRegressor

12주차

미션	스피드 데이팅 데이터셋을 이용해서 커플 성사 여부를 예측하라.
난이도	★★★
알고리즘	XG부스트(XGBoost)
데이터셋 파일명	dating.csv
종속변수	match(커플 성사 여부)
데이터셋 소개	• 스피드 데이팅에 대한 데이터임 • 스피드 데이팅은 남녀 수십 쌍이 짧은 시간을 보낸 뒤 서로에 대한 호감도를 표현하여 짝을 매칭하는 이벤트 • 독립변수로는 상대방과 내 정보, 개인의 취향, 상대방에 대한 평가 등이 있으며, 매칭 성사 여부가 종속변수임
문제유형	분류
평가지표	정확도, 혼동 행렬, 분류 리포트
사용한 모델	XGBClassifier

13주차

미션	카드 거래 내역 데이터셋을 이용해 이상거래를 예측하라.
난이도	★★★
알고리즘	라이트GBM(LightGBM)
데이터셋 파일명	fraud.csv
종속변수	Is_fraud(이상거래)
데이터셋 소개	• 이상거래에 관련된 데이터임 • 이상거래라 함은 카드값을 지불하지 않을 의도를 가지고서 결제를 하거나, 도난된 카드를 가지고 결제를 하는 등의 거래를 의미함 • 종속변수는 이상거래 여부임 • 독립변수는 거래 시간, 거래 금액, 고객 성별, 상점 범주 등임
문제유형	분류
평가지표	정확도, 혼동 행렬, 분류 리포트, ROC AUC 점수
사용한 모델	LGBMClassifier, train

14주차

미션	데이터들을 비슷한 속성끼리 분류하라.
난이도	★☆☆
알고리즘	K-평균 군집화(K-Means Clustering)
데이터셋 파일명	example_clustering.csv Customer.csv
종속변수	selling_price(판매 가격)
데이터셋 소개	• 여기에는 2개의 데이터를 사용함 • 첫 번째 데이터는 K-평균 군집화를 학습할 목적으로 인위적으로 만든 데이터로 변수들에는 아무런 의미가 없음 • 두 번째 데이터는 11장에서 사용한 데이터 중 일부 변수와 일부 고객 정보만을 포함함
문제유형	비지도 학습

평가지표	엘보우 기법, 실루엣 점수
사용한 모델	KMeans
데이터셋 소개	• numpy(numpy==1.19.5) • pandas(pandas==1.3.2) • seaborn(seaborn==0.11.2) • matplotlib(matplotlib==3.4.3) • sklearn(scikit-learn==0.23.2) • datetime, calendar

15주차

미션	데이터의 차원을 축소하여 이해하기 쉽게 시각화하라
난이도	★★☆
알고리즘	주성분 분석(Principal Component Analysis, PCA)
데이터셋 파일명	anonymous.csv
데이터셋 소개	• 변수 개수가 천 개가 넘는 데이터셋임 • 변수 이름이 익명 처리되어 있음 • 차원 축소 전후의 모델 학습 속도 및 예측 결과를 비교해보겠음
문제유형	비지도 학습
평가지표	AUC
사용한 모델	PCA
사용 라이브러리	• numpy(numpy==1.19.5) • matplotilib(matplotlib==3.2.2) • Pandas(pandas==1.3.5) • Sklearn(scikit-learn==1.0.2) • Seaborn(seaborn==0.11.2)
예제코드	• 위치 : colab.research.goolgle.com/github/musthave-ML10/notebooks/blob/main • 파일 : 13_PCA.ipynb