

생성형AI

Day 9

머신러닝 I



목차

1. 머신러닝 소개
2. 학습방법과 머신러닝
3. 지도학습 맛보기
4. 선형회귀
5. 나이브 베이즈
6. 모델의 학습과 평가
7. 실습과제



머신러닝 소개

머신러닝이란?

- 인공지능의 한 분야로, 데이터에서 패턴을 학습하고, 예측하거나 의사 결정을 내리는 알고리즘
- 인간의 개입 없이도 스스로 성능을 개선할 수 있는 능력을 가진 시스템
- 데이터를 통해 학습하면서 지속적으로 성능을 향상시키는 모델

머신러닝이 없다면? (스팸메일)

- 스팸으로 의심되는 단어에 대한 블랙리스트 구축
- 새로운 단어가 있을때마다 추가
- 스팸으로 의심되는 단어를 정하기 위해 전문가 필요

머신러닝이 있다면?(스팸메일)

- 스팸메일의 단어들에서 패턴을 분석
- 스팸메일에는 자주 등장하고 일반메일에는 자주 등장하지 않는 단어 추출
- 전문가 필요 X



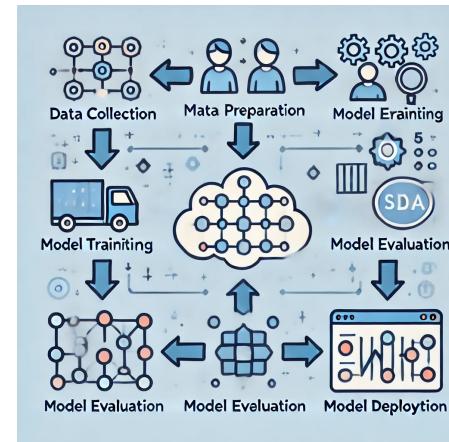
머신러닝 소개

머신러닝, 언제 써야하지?

- 데이터 패턴 분석: 사람이 직접 패턴을 찾기 힘든 대규모 데이터 패턴을 분석하기 위해
- 예측과 분류: 기존 데이터로부터 미래를 예측하거나 특정 범주로 분류해야 할 때
- 자동화된 의사결정 프로세스: 주어진 입력을 바탕으로 기대하는 출력을 반환할 때

머신러닝 프로세스를 진행하기 전 확인해보기

1. 어떤 질문(문제)에 대한 어떤 답(해결)을 원하는가?
2. 내가 갖고 있는 데이터에 답이 있는가?
3. 데이터가 충분한가?
4. 어떤 방법이 내 질문을 가장 잘 해결해 줄 수 있는가?
5. 내가 만든 머신러닝 프로세스의 결과를 어떻게 평가 할 수 있는가?



머신러닝 소개

주로 사용하는 라이브러리 – Numpy

- 과학 계산을 위한 핵심 파이썬 라이브러리로, 고성능 다차원 배열 객체와 다양한 수학 함수들을 제공
- 머신러닝과 데이터 분석의 기초 라이브러리로 널리 사용

주요 기능

- 다차원 배열 객체: 효율적인 다차원 배열(ndarray) 제공
- 수학 함수: 선형 대수, 통계, 푸리에 변환 등의 수학 함수 제공
- 배열 조작: 배열 생성, 변환, 인덱싱, 슬라이싱 등의 기능
- 빠른 계산: C로 구현되어 있어 빠른 계산 속도 제공

장점

- 성능: 대규모 데이터 처리를 위한 고성능 연산
- 호환성: 다른 과학 계산 및 머신러닝 라이브러리와의 호환성
- 사용의 용이성: 간단한 배열 연산으로 복잡한 수학 계산을 쉽게 수행



머신러닝 소개

주로 사용하는 라이브러리 – Pandas

- 데이터 조작과 분석을 위한 고성능 라이브러리
- 사용이 간편한 데이터 구조와 데이터 분석 도구를 제공
- 주로 데이터 전처리와 탐색적 데이터 분석(EDA)에 사용

주요 기능

- 데이터 프레임: 행과 열로 구성된 2차원 데이터 구조를 제공
- 데이터 조작: 데이터 정제, 결측값 처리, 데이터 병합 및 분할 등 다양한 기능
- 데이터 변환: 데이터 정렬, 필터링, 집계 등의 기능
- 시간 시계열 분석: 날짜 및 시간 데이터를 다루는 기능

장점

- 다양한 기능: 데이터 분석과 전처리를 위한 풍부한 기능 제공
- 호환성: 다른 파이썬 라이브러리(Numpy, Scikit-Learn 등)와의 호환성 우수



머신러닝 소개

주로 사용하는 라이브러리 – Scikit-Learn

- 파이썬에서 가장 널리 사용되는 머신러닝 라이브러리
- 간단하고 일관된 인터페이스를 제공하며, 다양한 알고리즘을 포함

주요 기능

- 데이터 전처리: 결측값 처리, 데이터 스케일링, 차원 축소 등을 포함
- 지도학습 알고리즘: 선형회귀, 로지스틱 회귀, 서포트 벡터 머신, 의사결정나무, 랜덤 포레스트, k-NN 등
- 비지도학습 알고리즘: k-means, PCA, DBSCAN 등
- 모델 평가 및 선택: 교차 검증, 하이퍼파라미터 튜닝, 다양한 평가 지표

장점

- 풍부한 문서화: 광범위한 문서와 예제 코드가 제공되어 학습과 적용이 용이함
- 확장성: 다른 파이썬 라이브러리(Numpy, Pandas 등)와의 호환성이 좋음



학습방법과 머신러닝

지도학습

- 레이블(정답)이 있는 데이터로 모델을 학습하여 새로운 데이터에 대한 예측을 수행
- 입력과 출력 간의 관계를 학습하여 새로운 입력 데이터에 대한 예측 가능

예시

- 이메일 스팸 필터링: 수천 개의 이메일 중 스팸과 일반 이메일로 라벨링된 데이터를 사용하여 학습된 모델은 새로운 이메일이 스팸인지 아닌지 분류
- 주식 가격 예측: 과거 주식 가격 데이터와 실제 주식 가격을 이용하여 학습한 모델은 미래의 주식 가격을 예측

주요 알고리즘

- 선형회귀: 두 변수 간의 선형 관계를 모델링
- 로지스틱 회귀: 이진 분류 문제를 해결하는 데 주로 사용
- 의사결정나무: 데이터의 특성에 따라 의사결정을 트리 구조로 모델링
- k-NN: 가장 가까운 k개의 이웃 데이터를 기반으로 예측

학습방법과 머신러닝

장점

- 명확한 목표: 정답(레이블)이 명확히 존재하므로 모델의 학습 목표가 분명함
- 다양한 응용 분야: 분류, 회귀 등 다양한 문제에 적용 가능
- 높은 성능: 충분한 양질의 데이터가 제공될 경우 높은 예측 성능을 발휘

단점

- 데이터 의존성: 고품질의 라벨링된 데이터가 많이 필요
- 과적합(overfitting): 훈련 데이터에 너무 치우쳐 학습할 경우, 새로운 데이터에 대한 일반화 능력이 떨어질 수 있음
- 복잡성 증가: 고차원의 데이터나 복잡한 문제에서는 모델의 복잡도가 급격히 증가할 수 있음

지도학습의 주요 도전 과제

- 데이터 라벨링: 대량의 데이터를 라벨링하는 것은 많은 시간과 비용이 소요되는 작업
- 데이터 편향: 라벨링된 데이터가 특정 패턴에 치우칠 경우, 모델이 편향된 예측을 할 가능성성이 높음
- 모델 해석 가능성: 복잡한 모델일수록 예측 결과를 해석하고 설명하기 어려움

학습방법과 머신러닝

비지도학습

- 레이블이 없는 데이터에서 패턴이나 구조를 발견하는 학습 방법
- 목표는 데이터 내의 숨겨진 구조나 관계를 이해하고, 이를 기반으로 데이터의 분포나 특징을 파악하는 것

예시

- 고객 세분화: 고객 데이터를 군집화하여 마케팅 전략 수립
- 이상 탐지: 정상 패턴에서 벗어난 이상 데이터를 식별
- 차원 축소: 고차원 데이터를 저차원으로 축소하여 데이터 시각화나 효율적인 데이터 처리에 활용

주요 알고리즘

- k-means: 데이터를 k개의 클러스터로 분할하는 군집
- PCA: 데이터의 차원을 축소하여 주요 특징을 추출
- DBSCAN: 밀도 기반 군집화
- 계층적 군집화: 데이터의 계층적 구조를 탐색하는 군집화 알고리즘

학습방법과 머신러닝

장점

- 라벨이 필요 없음: 라벨링된 데이터가 필요하지 않음
- 데이터 탐색: 데이터의 숨겨진 구조나 패턴을 발견 가능
- 다양한 응용 분야: 군집화, 이상 탐지, 차원 축소 등 다양한 문제에 적용

단점

- 해석의 어려움: 비지도학습의 결과는 해석하기 어려운 경우가 많음
- 모델 평가의 어려움: 지도학습처럼 명확한 정답이 없기 때문에, 모델의 성능을 평가하기가 어려움
- 초기 설정 민감도: 초기 파라미터 설정에 따라 결과가 크게 달라질 수 있음

비지도학습의 주요 도전 과제

- 최적의 파라미터 설정: 군집화 알고리즘의 경우, 최적의 클러스터 개수나 밀도 파라미터를 설정하는 것이 어려움
- 대규모 데이터 처리: 대규모 데이터셋에서는 계산 비용이 높아질 수 있음
- 노이즈 데이터 처리: 노이즈나 이상치가 많을 경우, 모델의 성능에 부정적인 영향을 미칠 수 있음

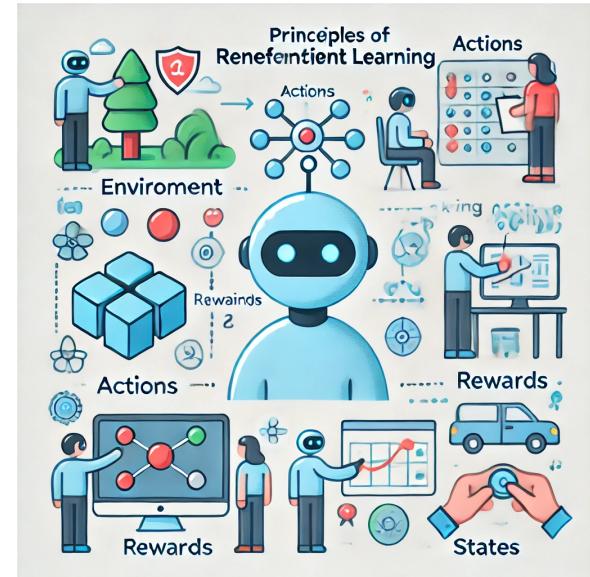
학습방법과 머신러닝

강화학습

- 에이전트(agent)가 환경(environment)과 상호작용하면서 보상을 최대화하는 행동(policy)을 학습하는 방법
- 시퀀스 기반의 의사결정 문제에서 최적의 행동을 찾는 것이 목표

예시

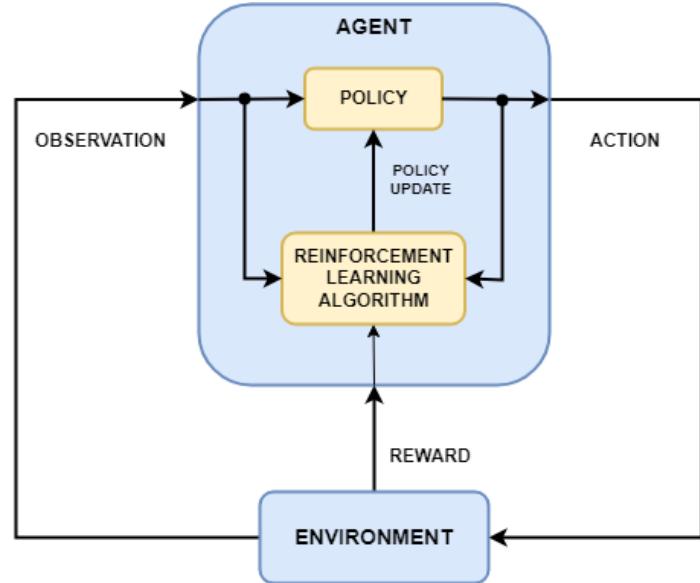
- 게임 인공지능: 스스로 학습하여 최적의 전략을 찾음
- 로봇 제어: 특정 작업을 수행하면서 환경과 상호작용하여 최적의 동작을 학습
- 자율 주행: 자율 주행 자동차가 도로 환경에서 최적의 주행 경로를 학습



학습방법과 머신러닝

주요 개념

- 에이전트(Agent)
 - 환경과 상호작용하며 학습을 수행하는 주체
 - 현재 상태를 기반으로 행동을 선택하고, 그 행동의 결과로 보상을 받으며, 이를 통해 학습을 진행
- 환경(Environment)
 - 에이전트가 상호작용하는 외부 시스템
 - 에이전트의 행동에 따라 상태를 변화시키고 보상을 제공
- 상태(State)
 - 환경의 현재 상태를 나타내는 정보.
 - 에이전트가 현재 상황을 이해하고, 적절한 행동을 선택하는 데 사용
- 행동(Action)
 - 에이전트가 현재 상태에서 선택할 수 있는 행동.
 - 에이전트는 주어진 상태에서 행동을 선택하고, 이 행동은 환경의 상태를 변화시킴
- 보상(Reward)
 - 에이전트의 행동 결과로 제공하는 피드백
 - 에이전트의 행동이 얼마나 좋은지에 대한 지표이며, 에이전트는 보상을 최대화하는 방향으로 학습
- 정책(Policy)
 - 주어진 상태에서 행동을 선택하는 전략
 - 정책은 상태를 입력받아 행동을 출력하며, 최적의 정책은 에이전트가 장기적으로 최대 보상을 얻도록 함
- 가치 함수(Value Function)
 - 특정 상태나 상태-행동 쌍의 가치를 평가하는 함수
 - 가치 함수는 장기적으로 얻을 수 있는 누적 보상의 기대값을 나타내며, 최적의 행동을 선택하는 데 도움을 줌



학습방법과 머신러닝

장점

- 순차적 의사결정 문제 해결: 시퀀스 기반의 문제에서 최적의 행동을 찾는 데 유리
- 적응성: 변화하는 환경에 대해 지속적으로 학습하고 적응할 수 있음

단점

- 복잡성: 학습 과정이 복잡하며, 많은 계산 자원이 필요
- 보상 설계: 적절한 보상 함수를 설계하는 것이 어려울 수 있음

강화학습의 주요 도전 과제

- 보상의 희소성: 특정 행동에 대한 보상이 드물 경우, 학습이 어려울 수 있음
- 고차원 상태 공간: 상태 공간이 매우 크거나 복잡할 경우, 학습이 어려울 수 있음
- 장기 의존성 문제: 긴 시퀀스의 의사결정에서 발생하는 의존성을 처리하는 것이 어려움
- 안정성: 학습 과정의 안정성을 확보하는 것이 중요하며, 불안정한 학습은 잘못된 정책으로 이어질 수 있음

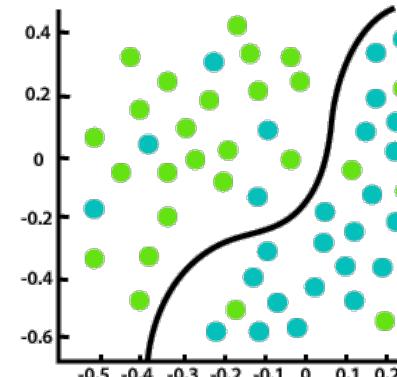
지도학습 맛보기

회귀

- 연속적인 값을 예측하는 문제
- 예시
 - 집값 예측
 - 기온 예측
- 기본 알고리즘 소개
 - 단순 선형 회귀: 하나의 독립 변수와 종속 변수 간의 관계를 직선으로 모델링
 - 다중 선형 회귀: 여러 독립 변수와 종속 변수 간의 관계를 모델링

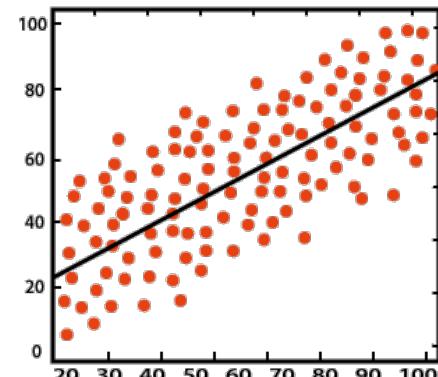
분류

- 범주형 값을 예측하는 문제
- 예시
 - 스팸 메일 분류
 - 이미지 분류
- 기본 알고리즘 소개
 - 로지스틱 회귀: 이진 분류 문제에서 확률을 예측
 - 서포트 벡터 머신: 두 클래스 간의 최대 마진을 찾는 방법
 - k-NN: 가장 가까운 k개의 이웃 데이터를 기반으로 분류



Classification

<https://mvje.tistory.com/77>



Regression

선형회귀

목표는 독립 변수와 종속 변수 간의 관계를 나타내는 선형 방정식을 찾는 것

모델이 비교적 간단하여 해석이 용이

- 수학적 배경
 - 회귀 방정식: $y = \beta_0 + \beta_1 x + \epsilon$
 - y : 종속변수(예측값), x : 독립변수, β_0 : 절편, β_1 : 기울기, ϵ : 오차 항 (모델의 예측과 실제 값 간의 차이)
 - 주어진 데이터를 잘 설명하는 β_0, β_1 의 값을 찾는 것
- 잔차 제곱합 최소화(Ordinary Least Squares, OLS): 최적의 직선을 찾기 위해 사용하는 방법으로 잔차(오차)의 제곱합을 최소화
 - $$RSS = \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_i))^2$$
 - RSS : Residual Sum of Squares(잔차 제곱합), n : 데이터 포인트 수, y_i : 실제 값, $\beta_0 + \beta_1 x_i$: 예측값
 - OLS는 이 잔차 제곱합을 최소화하는 β_0, β_1 의 값을 찾는 것

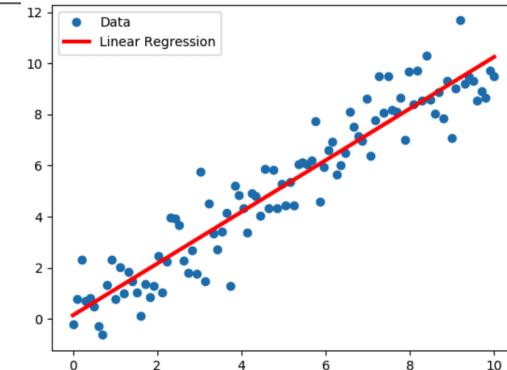
선형회귀

단순 선형회귀(Simple Linear Regression)

- 하나의 독립 변수와 하나의 종속 변수를 다루는 선형회귀
 - 예: 집 크기(x)에 따른 집 가격(y)의 관계를 모델링

다중 선형회귀(Multiple Linear Regression)

- 다중 선형회귀는 여러 개의 독립 변수를 사용하는 선형회귀
 - $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \epsilon$
- 예: 집 가격(y)을 예측할 때, 집 크기(x_1), 방 개수(x_2), 위치(x_3) 등의 여러 독립 변수를 고려



모델 적합성

- R-제곱(R-squared): 모델이 데이터를 얼마나 잘 설명하는지 나타내는 지표로 0과 1 사이의 값을 가지며, 1에 가까울수록 모델이 데이터를 잘 설명함
- $$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$
- \hat{y}_i : 예측값 \bar{y}_i : 실제값의 평균

잔차 분석

- 잔차(Residuals): 실제 값과 예측 값의 차이로 잔차의 분포를 분석하여 모델의 적합성을 평가할 수 있음
- 잔차 플롯(Residual Plot): 예측 값에 대한 잔차를 그래프로 표현, 잔차가 무작위로 분포하면 모델이 적합하다는 신호

나이브 베이즈

조건부 확률을 이용한 분류 알고리즘으로, 독립 변수들이 독립적이라는 가정 하에 작동

- 베이즈 정리

- 조건부 확률을 이용해 사건의 사후 확률을 계산하는 원리

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

- $P(A|B)$: 사건 B가 일어났을 때 사건 A가 일어날 확률 (사후확률)

- ex) $P(\text{Spam}|\text{words})$, 특정 단어가 있을 때 스팸메일일 확률

- $P(B|A)$: 사건 A가 일어났을 때 사건 B가 일어날 확률

- ex) $P(\text{words}|\text{Spam})$, 스팸메일에 특정 단어가 있을 확률 -> 구할 수 있음

- $P(A)$: 사건 A가 일어날 확률 (사전확률)

- ex) $P(\text{Spam})$, 스팸메일 비율 -> 구할 수 있음

- $P(B)$: 사건 B가 일어날 확률

- ex) $P(\text{words})$, 특정 단어가 있을 확률 -> 구할 수 있음

나이브 베이즈

장점

- 간단하고 빠름: 알고리즘이 단순하며, 학습과 예측 속도가 빠름
- 작은 데이터셋에 적합: 비교적 작은 데이터셋에서도 잘 작동
- 다양한 응용 가능: 텍스트 분류, 스팸 필터링, 문서 분류 등 다양한 분야에서 사용 가능
- 확률 해석 가능: 출력이 확률값으로 예측의 불확실성을 해석 가능

단점

- 독립 가정의 한계: 특성들 간의 독립 가정이 현실적으로 맞지 않을 수 있음
- 연속형 데이터 처리의 어려움: 가우시안 나이브 베이즈를 사용하지 않는 한, 연속형 데이터의 처리가 복잡함
- 훈련 데이터의 편향성: 훈련 데이터가 편향되면, 예측 성능이 크게 저하됨

종류

- GaussianNB: 연속 데이터
- BernoulliNB: 이진 데이터
- MultinomialNB: 이산적 카운트 데이터

모델의 학습과 평가

훈련데이터와 테스트 데이터의 분할

- 머신러닝 모델의 성능을 정확히 평가하기 위해 전체 데이터셋을 훈련 데이터와 테스트 데이터로 분할
- 모델의 훈련에서 발생할 수 있는 과적합을 방지하기 위해 테스트 데이터로 일반화 성능을 확인
- 일반적인 비율: 훈련 데이터와 테스트 데이터의 분할 비율은 일반적으로 70:30 또는 80:20

데이터 분할 방법

- 임의 분할(Random Split)
 - 데이터를 무작위로 섞은 후, 지정된 비율에 따라 훈련 데이터와 테스트 데이터로 분할
 - 데이터의 순서가 모델 성능에 영향을 미치지 않음
- 층화 분할(Stratified Split)
 - 데이터의 클래스 분포를 유지하면서 훈련 데이터와 테스트 데이터로 분할
 - 주로 클래스 불균형 문제가 있는 경우에 사용

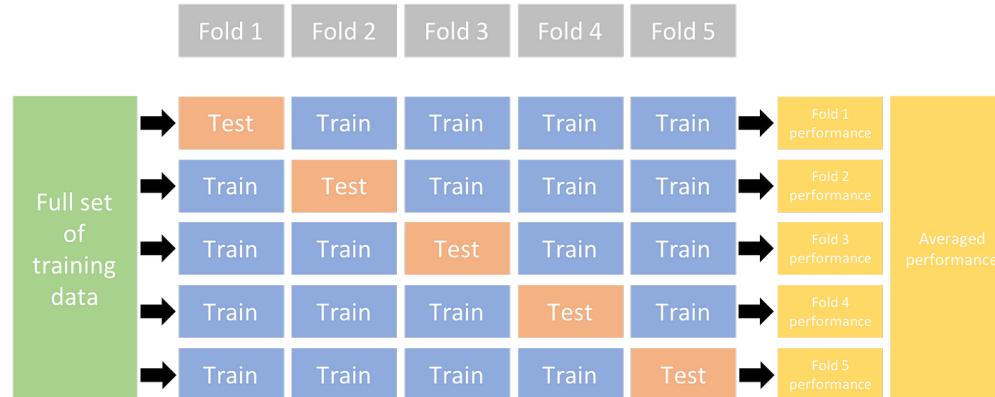
모델의 학습과 평가

교차검증(cross-validation)

- 모델의 일반화 성능을 평가하고, 데이터의 분할로 인한 편향을 줄이기 위한 방법

k-Fold Cross-Validation

- 데이터셋을 k개의 폴드(fold)로 나누어 교차 검증을 수행하는 방법
- 각 폴드는 한 번씩 테스트 데이터로 사용되고, 나머지 $k-1$ 개의 폴드는 훈련 데이터로 사용



모델의 학습과 평가

훈동행렬

- 모델이 예측한 결과와 실제 결과를 비교하여 분류 오류의 유형을 확인
- 다중 클래스 분류 문제에서도 유용하지만, 이진 분류 문제에서 더 자주 사용
- True Positive (TP)
 - 모델이 Positive로 예측한 결과가 실제로도 Positive인 경우
 - 예: 실제로 스팸 이메일이 스팸으로 분류된 경우
- False Negative (FN)
 - 모델이 Negative로 예측한 결과가 실제로는 Positive인 경우
 - 예: 실제로 스팸 이메일이 스팸이 아닌 것으로 분류된 경우 (누락된 스팸)
- False Positive (FP)
 - 모델이 Positive로 예측한 결과가 실제로는 Negative인 경우
 - 예: 실제로 스팸이 아닌 이메일이 스팸으로 분류된 경우 (잘못된 스팸)
- True Negative (TN)
 - 모델이 Negative로 예측한 결과가 실제로도 Negative인 경우
 - 예: 실제로 스팸이 아닌 이메일이 스팸이 아닌 것으로 분류된 경우

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

모델의 학습과 평가

- 정확도: 전체 예측 중 맞춘 비율

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

- 정밀도: 양성 예측 중 실제 양성 비율

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

- 재현율: 실제 양성 중 맞춘 비율

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

- F1-score: 정밀도와 재현율의 조화 평균

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

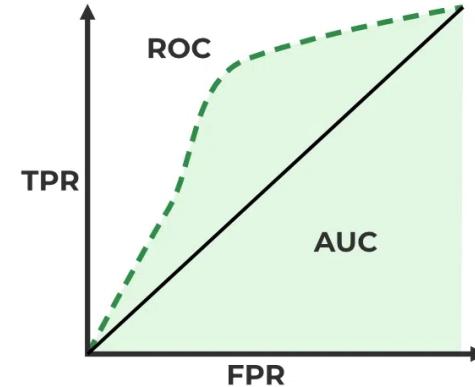
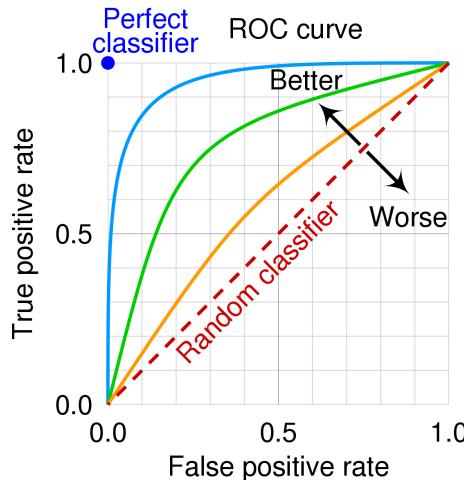
		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
	Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$	

<https://manisha-sirsat.blogspot.com/2019/04/confusion-matrix.html>

모델의 학습과 평가

ROC 곡선

- ROC: 모델의 분류 임계값을 변화시키면서 True Positive Rate(재현율)와 False Positive Rate를 비교하는 곡선
- AUC: ROC 곡선 아래 면적으로, 모델의 전반적인 성능을 나타냄, 1에 가까울수록 좋은 모델
- $TPR = \frac{TP}{TP + FN}$, $FPR = \frac{FP}{FP + TN}$



이론 예제

collar: <https://colab.research.google.com/drive/15J-QV4gaoE-m76jUJRZQywgJLkfbjdJ5?usp=sharing>

실습 과제

collar: <https://colab.research.google.com/drive/1amtUnaZzJW2Ir12jZkP3tGqOoz2KGNpQ?usp=sharing>

실습 진행