머신러닝

머신러닝

명시적으로 프로그래밍을 하지 않고도 컴퓨터가 **학습할 수 있는 능력**을 갖게 하는 것

종류

1. 지도학습
2. 비지도학습
3. 강화학습

머신러닝을 위한 데이터 전처리

범주형 자료

범주형 데이터는 몇 개의 범주로 나누어진 자료

범주의 크기가 의미가 없다 → 명목형 자료

범주의 크기가 의미가 있다 → 순서형 자료

명목형 자료

1) 수치 맵핑 방식

일반적으로 범주를 0,1로 맵핑

3개 이상인 경우, 수치의 크기 간격을 같게 하여 수치 맵핑 Ex)(0,1,2,3,...)

2) 더미 기법

각 범주를 0 or 1로 변환

순서형자료

1) 수치 맵핑 방식

수치에 맵핑하여 변환하지만, 수치 간 크기 차이는 커스텀 가능

크기 차이가 머신러닝 결과에 영향을 끼칠 수 있음

Ex) 매우 많음:10 없음:0 조금 많음: 4

수치형 자료

크기를 갖는 수치형 값으로 이루어진 데이터

머신러닝의 입력으로 바로 사용할 수 있으나, 모델의 성능을 높이기 위해서 데이터 변환이 필요

대표적인 수치형 자료 변환 방식

1) 스케일링 - 정규화 - 표준화

2) 범주화

스케일링

변수 값의 범위 및 크기를 변환하는 방식

변수간의 범위가 차이가 나면 사용

1) 정규화(Nomalization)

변수 X 를 정규화한 값 X`

X` = $\frac{X - X_{min}}{X_{max} - X_{min}}$

2) 표준화(Standardization)

변수 X를 표준화한 값 X`

X` = $\frac{X - \mu}{\sigma}$

범주화

변수의 값보다 범주가 중요한 경우 사용

Ex) 시험점수를 평균 이상 →1 평균 이하→1

회귀

데이터를 **가장 잘 설명하는 모델을** 찾아 입력값에 따른 미래 결과값을 예측하는 알고리즘

완벽한 예측은 불가능하기에 최대한 잘 근사해야 한다

각 데이터의 실제 값과 모델이 예측하는 값의 차이를 최소한으로 하는 선을 찾자

단순 선형 회귀

데이터를 설명하는 모델을 직선 형태로 가정

직선을 구성하는 β_0(y절편)와β_1(기울기)를 구해야함

Loss 함수

실제 값과 예측 값 차이의 제곱의 합

→Loss함수가 작을 수록 좋은 모델이다.

Loss 함수: $\frac{1}{N}\sum_i^N \left(y^{(i)} - \left(\beta_0 + \beta_1 x^{(i)}\right)\right)^2$

Loss함수에서 주어진 값은 입력 값과 실제 값이다.

→ y절편, 기울기 값을 조절하여 Loss함수의 크기를 작게 한다.

Loss 함수 줄이기

1) 경사 하강법

Loss함수 값이 제일 작게 하는 β_0, β_1를 계산 한번으로 구하는 것이 아니라 초기값에서 점진적으로 구하는 방식

1) β_0, β_1값을 랜덤하게 초기화

2) β_0, β_1값으로 Loss 값 계산

3) 현재 β_0, β_1 값을 어떻게 변화해야 Loss 값을 줄일 수 있는지 알 수 있는 Gradient 값 계산

4) Gradient 값을 활용하여 β_0, β_1 업데이터

5) Loss 값의 차이가 거의 없어질 때까지 반복

단순 선형 회귀 특징

- 가장 기초적이나 여전히 많이 사용됨
- 입력값이 1개인 경우에만 적용 가능
- 입력값과 결과값의 관계를 알아보는데 용이함
- 입력값이 결과값에 얼마나 영향을 미치는지 알 수 있음
- 두 변수 간의 관계를 직관적으로 해석하고자 하는 경우 활용

다중 선형 회귀

입력값 X가 여러개인 경우 활용할 수 있는 회귀 알고리즘 각 개별X_i에 해당하는 최적의 β_1을 찾아야함

다중 선형 회귀 모델

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + +\beta_M X_M$$

다중 선형 회귀 모델의 Loss함수

단순 선형 회귀와 마찬가지로 Loss함수는 입력값과 실제 차이의 제곱의 합으로 정의

Loss 함수: $\frac{1}{N}\sum_i^N \left(y^{(i)} - \left(\beta_0 + \beta_1 {x_1}^{(i)} + \beta_2 {x_2}^{(i)} + \cdots + \beta_M {x_M}^{(i)}\right)\right)^2$

다중 선형 회귀 특징

- 여러 개의 입력값과 결과값 간의 관계 확인 가능
- 어떤 입력값이 결과값에 어떠한 영향을 미치는지 알 수 있음
- 여러 개의 입력값 사이 간의 상관관계가 높을 경우 결과에 대한 신뢰성을 잃을 가능성이 있음

회귀 평가 지표

목표를 얼마나 잘 달성했는지 정도를 평가해야 함

실제값과 모델이 예측하는 값의 차이에 기반한 평가 방법 사용

RSS - 단순오차

1. 실제 값과 예측 값의 단순 오차 제곱함
2. 값이 작을수록 모델의 성능이 높음
3. 전체 데이터에 대한 실제 값과 예측하는 값의 오차 제곱의 총합

RSS 특징

- 가장 간단한 평가 방법으로 직관적인 해석이 가능
- 오차를 그대로 이용하기 때문에 입력값의 크기에 의존적

- 절대적인 값과 비교가 불가능

MSE(Mean Squared Error)

평균 제곱 오차, RSS 에서 데이터 수 만큼 나눈 값

작을수록 모델의 성능이 높다고 평가할 수 있음

$$MSE = \frac{1}{N}\sum_{i}^{N}\left(y^{(i)} - \left(\beta_0 + \beta_1 x^{(i)}\right)\right)^2$$

MAE(Mean Absolute Error)

평균 절대값 오차, 실제 값과 예측 값의 오차의 절대값의 평균

작을수록 모델의 성능이 높다고 평가할 수 있음

MSE, MAE특징

- MSE: 이상치에 민감함
- MAE: 변동성이 큰 지표와 낮은 지표를 같이 예측할 시 유용
- 가장 간단한 평가 방법들로 직관적인 해석이 가능
- 평균을 그대로 이용하기 때문에 입력 값의 크기에 의존적
- 절대적인 값고 비교 불가능

R^2(결정계수)

회귀 모델의 설명력을 표현하는 지표

1에 가까울수록 높은 성능의 모델이라고 할 수 있음

$$R^2 = 1 - \frac{RSS}{TSS}$$

TSS는 데이터 평균값과 실제 값 차이의 제곱

$$TSS = \sum_{i}^{N} \left(y^{(i)} - \bar{y}\right)^2 \qquad \bar{y} = \frac{1}{N} \sum_{i}^{N} y^{(i)}$$

R^2의 특징

- 오차가 없을수록 1에 가까운 값을 갖음
- 값이 0인 경우, 데이터의 평균 값을 출력하는 직선 모델을 의미함
- 음수 값이 나온 경우, 평균 값 예측 보다 성능이 좋지 않음

분류

주어진 입력 값이 어떤 클래스에 속할지에 대한 결과 값을 도출하는 알고리즘

다양한 분류 알고리즘이 존재하며, 예측 목표와 데이터 유형에 따라 적용

분류 알고리즘

종류	이름
트리 구조 기반	의사결정나무 랜덤포레스트...
확률 모델 기반	나이브 베이즈 분류기
결정 경계 기반	선형 분류기 로지스틱 회귀 분류기 SVM
신경망	퍼셉트론 딥러닝 모델

의사결정나무

특정 질문들을 통해 정답을 찾아가는 모델

최상단의 뿌리 마디에서 마지막 끝 마디까지 아래 방향으로 진행

불순도

다른 데이터가 섞여 있는 정도

- **지니 계수**

해당 구역 안에서 특정 클래스에 속하는 데이터의 비율을 모두 제외한 값 즉, 다양성을 계산하는 방법

- **지니 불순도**

$$\text{Gini Index} = 1 - (yes\text{의 확률})^2 - (no\text{의 확률})^2$$

$$\text{Gini Impurity} = \frac{n_1}{N} Gini_1 + \frac{n_2}{N} Gini_2$$

n_i: i번째 자식 마디의 데이터 개수
N: 부모 마디의 데이터 개수

의사결정나무의 깊이(중간노드의 개수)의 trade-off

의사결정나무의 깊이가 깊어질 수록 세분화해서 나눌 수 있음

하지만 너무 깊은 모델은 과적합의 문제

의사결정나무의 특징

- 결과가 직관적이며, 해석하기 쉬움
- 나무 깊이가 깊어질수록 과적합 문제 발생 가능성이 높음
- 학습이 끝난 트리의 작업 속도가 매우 빠름

분류 평가 지표

혼동 행렬

분류 모델의 성능을 평가하기 위함

		예측	
		Positive	Negative
실제	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

정확도

전체 데이터 중에서 제대로 분류된 데이터의 비율로 모델이 얼마나 정확하게 분류하는지를 나타냄

일반적으로 분류 모델의 주요 평가 방법으로 사용됨

클래스 비율이 불균형 할 경우 평가 지표의 신뢰성을 잃을 가능성이 있음

$$Accuracy = \frac{TP+TN}{P+N}$$

$$P: TP + FN,$$

$$N: TN + FP$$

정밀도

모델이 Positive라고 분류한 데이터 중에서 실제로 Positive인 데이터의 비율

Negative가 중요한 경우

실제로 Negative인 데이터를 Positive라고 판단하면 안되는 경우 사용되는 지표

스팸 메일 판결을 위한 분류

스팸 → Positive

일반 → Negative

일반 메일을 스팸 메일로 잘못 예측했을 경우 중요한 메일을 전달 받지 못하는 상황이 발생할 수 있음

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

재현율

실제로 Positive인 데이터 중에서 모델이 Positive로 분류한 데이터의 비율

Positive가 중요한 경우

실제로 Positive인 데이터를 Negative라고 판단하면 안되는 경우 사용

악성 종양 여부 판결을 위한 검사

악성 종양 → Positive

양성 종양 → Negative

악성 종양을 양성 종양으로 잘못 예측했을 경우 제 때 치료를 받지 못하게 되어 생명이 위험

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

다양한 분류 지표의 활용

- 분류 결과를 전체적으로 보고 싶다 → 혼동 행렬
- 정답을 얼마나 잘 맞췄나 → 정확도
- FP , FN의 중요도가 높다 → 정밀도, 재현율