

1. 머신러닝에 대한 이해

1.1 머신러닝의 의미

머신러닝은 데이터의 훈련 샘플들을 학습해 데이터의 규칙과 패턴을 학습하여 결과를 예측하는 분야입니다. 머신러닝에서는 데이터에 대한 규칙을 프로그래밍으로 구현하지 않아도 파이썬 API 기반의 라이브러리로 손쉽게 모델을 구축할 수 있으며, 이를 통해 자동으로 데이터의 패턴을 학습할 수 있습니다.

1.2. 머신러닝 분석 프로세스

머신러닝 분석 프로세스 개요

Step	설명	실전팁
Step 1 데이터 확인	데이터 불러오기 데이터 속성 확인	- 독립변수, 종속변수 확인 - 연속형 vs 범주형 확인 - 적용 가능한 분석모델 확인 (회귀/분류/비지도 학습)
Step 2 EDA	데이터 시각화를 통한 인사이트 도출 분석 방향 수립	- 변수에 적합한 데이터 시각화 - 향후 스텝 수립
Step 3 전처리	결측치·이상치 처리 데이터 분할 및 범주형 변수처리 스케일링 및 샘플링	- 표준화(평균 0, 표준편차 1) 또는 MinMax 정규화 - 결측치 확인 후 처리 - 이상치 확인 후 처리
Step 4 학습	회귀/분류/비지도/하이퍼파라미터 조절	- 머신러닝 알고리즘 적용 - 회귀/분류/비지도 학습 - 최적모델을 결정하기 위해 하이퍼파라미터 탐색·조절
Step 5 평가	분석 정확도 확인/알고리즘 성능 제시	- 평가세트에 최종모델을 적용 - 평가세트에 대한 정확도를 머신러닝 분석에 대한 성능으로 제시

1.2.2 머신러닝 분석 프로세스 상세

1. 데이터 확인

먼저 분석할 데이터의 특성과 구성 요소를 명확히 파악하는 것이 중요합니다. 여기에서는 데이터의 독립변수와 종속변수를 식별하고, 각 변수의 특성을 이해합니다. 종속변수가 있는 경우, 그것이 연속형인지 범주형인지에 따라 적합한 분석 모델을 선택합니다. 연속형 종속변수는 회귀 분석을, 범주형 종속변수는 분류 분석을 활용합니다. 종속변수가 없다면, 비지도 학습이 적합합니다. 또한, 독립변수 중 범주형 변수가 있는지 확인하고, 이를 효과적으로 분석에 활용하기 위한 방법을 고민해야 합니다.

2. EDA

탐색적 데이터 분석(EDA)은 데이터의 주요 특징을 시각적으로 파악하고, 이를 바탕으로 분석 방향을 설정하는 단계입니다. 다양한 시각화 도구를 사용해 데이터의 분포, 관계, 이상치 등을 파악합니다. 이를 통해 데이터의 구조와 주요 패턴을 이해하고, 어떤 전처리나 분석 기법이 적합할지 결정할 수 있습니다. 또한, 이 과정에서 데이터의 문제점이나 개선해야 할 점을 발견하여 다음 단계의 전처리 전략을 세우는 것이 중요합니다.

3. 전처리

데이터 전처리는 분석의 핵심 단계로, 데이터의 품질을 향상시키고 분석에 적합한 형태로 변환하는 과정입니다. 여기에는 결측치와 이상치의 처리, 범주형 변수의 인코딩, 데이터 스케일링 등이 포함됩니다. 예를 들어, 결측치는 평균 또는 중앙값으로 대체하거나, 삭제하거나, 특수한 값을 이용해 처리할 수 있습니다. 이상치는 데이터의 분포와 관계를 고려하여 제거하거나 수정합니다. 범주형 변수는 원핫인코딩이나 라벨 인코딩을 통해 수치형으로 변환하고, 연속형 변수는 표준화나 정규화를 통해 스케일링합니다.

4. 모델학습

모델 학습 단계에서는 전처리된 데이터를 사용하여 머신러닝 모델을 학습시킵니다. 데이터의 특성과 분석 목표에 따라 적합한 모델을 선택하고, 이를 학습 데이터에 적용합니다. 회귀, 분류, 비지도 학습 등의 방법을 고려하여 모델을 구축합니다. 또한, 모델의 성능을 최적화하기 위해 교차 검증을 통해 하이퍼파라미터 튜닝을 수행합니다. 다양한 하이퍼파라미터 값을 실험하여 최적의 설정을 찾고, 이를 통해 모델의 예측력을 극대화합니다.

5. 성능평가

마지막으로, 평가 데이터셋을 사용하여 학습된 모델의 성능을 검증합니다. 모델의 예측 성능을 정확히 측정하기 위해, 평가 데이터는 학습과 검증에 사용하지 않은 새로운 데이터여야 합니다. 주요 평가 지표로는 정확도, 정밀도, 재현율, F1 점수 등을 사용하여 모델의 성능을 종합적으로 평가합니다. 평가 결과를 바탕으로 모델의 장단점을 분석하고, 필요시 추가적인 튜닝이나 개선을 고려할 수 있습니다. 이 과정에서, 학습 데이터나 검증 데이터를 평가에 사용하지 않도록 주의해야 합니다.

1.3 성능평가 기법

앞서 살펴보았듯이 머신러닝 분석 과정은 데이터를 가공 과정을 거친 후 모델이 데이터를 학습한 뒤 알고리즘을 평가하는 프로세스로 구성됩니다. 모델 예측 성능을 평가하는 것은 결국 학습모델의 실질값을 얼마나 정확하게 맞추었는지를 나타내는 것입니다. 머신러닝 모델은 여러 가지 방법으로 예측 성능을 평가할 수 있습니다. 이를 성능평가지표(Evaluation Metric)라 하며, 일반적으로 분석 알고리즘이 회귀분석인지 분류분석인지에 따라 여러 종류로 나뉩니다.

1.3.1 회귀분석

회귀분석에서는 실질값과 예측값의 차이를 기반으로 한 지표들을 중심으로 성능평가지표가 발전해왔습니다. 실질값과 예측값의 차이를 구해서 이 값들을 단순히 합하게 되면 +와 -가 섞여 오류가 상쇄될 수 있습니다. 극단적으로 두 데이터 값의 차이가 하나는 -2, 다른 하나는 +2라면 최종 오류는 0이 됩니다. 이러한 문제를 해결하기 위해 오류의 절댓값을 구하거나 제곱한 뒤 평균값을 구합니다. 회귀분석에서 사용하는 성능평가지표는 다음과 같습니다.

📌 MAE(Mean Absolute Error)

실제값과 예측값의 차이를 절대값으로 변환해 평균을 낸 것

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i|$$

[image mae]

📌 MSE(Mean Squared Error)

실제값과 예측값의 차이를 제곱해 평균을 낸 것

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

[image mse2]

1.3.2 분류분석

분류분석은 실제분류와 예측분류가 얼마나 일치했는가를 기반으로 알고리즘의 성능을 평가합니다. 단, 이진분류에서는 단순히 정확도로만 모델을 평가했을 때 잘못된 평가 결과를 초래할 수 있습니다. 예를 들어, 클래스가 매우 불균형한 데이터셋에서 높은 정확도가 실제로 좋은 성능을 의미하지 않을 수 있습니다. 이런 경우에는 정확도 이외의 다른 성능 평가지표를 고려해야 합니다. 분류분석에서 사용되는 다양한 성능 평가지표를 살펴보겠습니다.

📌 정확도(Accuracy)

실제 데이터와 예측 데이터가 얼마나 같은지 판단하는 지표

$$Accuracy = \frac{\text{예측결과가 동일한 데이터 건수}}{\text{전체 예측 데이터 건수}}$$

📌 정확도의 한계

데이터 구성에 따라 머신러닝 모델의 성능을 왜곡할 가능성이 있습니다. 예를 들어, 클래스가 매우 불균형한 데이터셋에서는 대부분의 데이터를 한 클래스만으로 예측해도 높은 정확도를 얻을 수 있습니다. 이러한 경우, 정확도는 모델의 성능을 제대로 평가하지 못합니다.

정확도 지표가 어떻게 머신러닝 모델의 성능을 왜곡할 수 있는지 예를 들어 보겠습니다. 예를 들어 3%의 확률로 발생하는 질병을 예측하는 분류모델을 만들어야 하는 사람이 있다고 가정해봅시다. 만약 이 사람이 만든 모델이 모든 데이터에 대해 '질병이 발생하지 않는다'고 예측한다면, 예측결과가 동일한 데이터의 비율은 97%가 되므로, 정확도는 97%가 됩니다. 그러나 이는 질병 예측의 관점에서 매우 부적절한 모델입니다.

정확도는 이러한 예시처럼 불균형한 클래스 분포에서 머신러닝 모델의 성능을 평가할 경우 적합한 지표가 아닐 수 있습니다. 따라서, 분류분석의 성능지표로서 정확도(Accuracy)는 한계가 있으며, 보완을 위해 여러 가지 분류 지표를 함께 고려해야 합니다.

📌 혼동행렬(Confusion Matrix)

분류의 예측오류가 얼마나 있고 어떤 유형의 예측 오류가 발생하고 있는지 나타내는 지표

혼동행렬은 4분면 행렬로, 실제 레이블 클래스 값과 예측 레이블 클래스 값이 어떤 유형의 오류를 가지고 매핑되는지를 보여줍니다.

📌 혼동행렬의 구성

cm

혼동행렬의 4분면은 각각 아래와 같은 의미를 지닙니다.

구분	내용
TN	예측값을 Negative(0)로 예측했고, 실제 값도 Negative(0)
FP	예측값을 Positive(1)로 예측했지만, 실제 값은 Negative(0)
FN	예측값을 Negative(0)로 예측했지만, 실제 값은 Positive(1)
TP	예측값을 Positive(1)로 예측했고, 실제 값도 Positive(1)

이 4가지 값을 조합해 분류모델의 성능을 측정하는 주요 지표인 정확도(Accuracy), 정밀도(Precision), 재현율(Recall)을 알 수 있습니다. 먼저 앞서 보았던 정확도(Accuracy)는 아래의 수식으로 계산됩니다.

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

📌 정밀도(Precision)와 재현율(Recall)

정밀도와 재현율은 Positive 데이터 예측에 집중한 성능평가지표입니다. 앞서 예시로 들었던 사례는, A가 만든 머신러닝은 Positive로 예측한 값이 없기 때문에, 정확도는 97%이지만 TP가 하나도 없기 때문에 정밀도와 재현율은 모두 0입니다.

1. 정밀도

Positive로 예측한 것들 중 실제로도 Positive인 것들의 비율

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

- 양성 예측도라 불리며, Positive 예측 성능을 더욱 정확하게 측정하기 위한 평가지표
- 정밀도가 상대적인 중요성을 가지는 경우: 실제 Negative인 데이터를 Positive로 잘못 예측했을 때 업무상 큰 영향이 발생할 때 유용

2. 재현율

실제 Positive인 것들 중 Positive로 예측한 것들의 비율

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

- 민감도(Sensitivity) 또는 TPR(True Positive Rate)라고 불리며, 실제 Positive 데이터를 얼마나 잘 찾아내는지를 나타냄
- 재현율이 상대적인 중요성을 가지는 경우: 실제 Positive인 데이터를 Negative로 잘못 예측했을 때 업무상 큰 영향이 발생할 때 유용

📌 정밀도와 재현율의 상충관계 – 트레이드오프(Trade-off)

분류 결정 임계값(Threshold)을 조정함으로써 정밀도 또는 재현율의 수치를 조절할 수 있습니다. 하지만 둘은 상충관계에 있는 성능 평가지표이기 때문에 한쪽을 강제로 높이면 다른 하나의 수치가 떨어지게 됩니다. 분석 상황에 따라 정밀도와 재현율 중 하나에 집중해야 하는 경우가 있지만, 둘 중 하나만 강조해서는 안 됩니다. 두 평가지표의 수치가 적절한 조화를 이루어야 종합적으로 분류모델의 성능을 평가할 수 있습니다.

📌 F1 스코어

정밀도와 재현율이 어느 한쪽으로 치우치지 않고 적절한 조화를 이룰 때 상대적으로 높은 수치를 나타냄

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

📌 ROC 곡선과 AUC 스코어

1. ROC 곡선

FPR(False Positive Rate)이 변할 때 TPR(True Positive Rate)이 변하는 것을 나타내는 곡선입니다.


$$\text{TNR} = \frac{\text{TN}}{\text{FP} + \text{TN}}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} = 1 - \text{TNR}$$

- TPR을 y축으로, FPR을 x축으로 하는 그래프
- 분류 결정 임계값을 조절하면서 FPR이 0부터 1까지 변할 때 TPR의 변화를 그래프에 나타냄
- 우수한 그래프일수록 곡선이 왼쪽 상단으로 치우쳐짐

2. AUC 스코어

Area Under the ROC Curve (ROC 곡선 아래의 면적)로, ROC 곡선 아래의 면적 값을 분류 성능지표로서 사용할 수 있음

- AUC 값은 0에서 1 사이의 값을 가지며, 1에 가까울수록 모델의 예측 성능이 우수함
- 랜덤 수준의 AUC 값은 0.5  분류를 전혀 하지 못함
- 완벽한 모델은 AUC 값이 1이며, 이는 FPR이 0일 때 TPR이 1임을 의미합니다.