CHAPTER2. 머신러닝 프로젝트 처음부터 끝까지

- 큰그림 보기
- 데이터 가져오기
- 데이터 탐색 및 시각화
- 데이터 전처리
- 모델 선택과 훈련
- 모델 세부 튜닝

2.1 큰 그림 보기

2.1.1 문제 정의

- 비즈니스의 목적이 정확히 무엇인가요?
- 현재 솔루션은 어떻게 구성되어 있나요?
- 지도/비지도/강화, 분류/회귀, 배치/온라인 중 어떤걸 써야할까요?

2.1.2 성능측정지표 선택

- 회귀 : RMSE, MAE, ANOVA Table
- 분류 : Confusion Matrix, ROC Curve

2.2 데이터 가져오기

2.2.1 데이터 구조 훑어보기

- head( ) : 처음 다섯행을 보여줍니다.
- info( ) : 데이터에 대한 간략한 설명, 전체 행 수, 특성의 데이터 타입과 널값이 아닌 값의 개수 확인합니다.
- describe( ) : count, mean, std, 4분위수 등 숫자형 특성의 요약정보를 보여줍니다 (수치형)
- value_counts( ) : 카테고리별 개수를 파악합니다. (카테고리형)
- hist( ) : 히스토그램을 통해 분포, 치우침 등을 파악합니다.

2.2.2 테스트 데이터 만들기

- from sklearn.model_selection import train_test_split

2.3 데이터 탐색 & 시각화 (EDA)

- 그래프 : plot( )을 활용해 데이터를 시각화합니다.
- 상관관계 조사 : corr( )를 통해 표준 상관계수를 구합니다.
- 특성조합으로 실험 : 외부데이터를 활용하거나 기존의 데이터를 조합해 새로운 데이터를 만들어 분석합니다.

2.4 데이터 전처리 (Preprocessing)

2.4.1 데이터 정제

- dropna( ) : 해당 구역을 제거합니다.
- drop( ) : 전체 특성을 삭제합니다.
- fillna( ) : 특정 값으로 채웁니다. (0, 평균, 중간값 등)
- Imputer(strategy = " ") : 누락된 값을 손쉽게 다루도록 해줍니다. (sklearn)

2.4.2 텍스트 & 범주형

- 대부분의 머신러닝 알고리즘은 숫자형을 다루므로 이 카테고리를 텍스트에서 숫자로 바꿔야 합니다.
- factorize( ) : 각 카테고리를 다른 정숫값으로 매핑해줍니다. 비슷한 단어를 반영할 수 없습니다.
- OheHotEncoder( ) : 한 특성값을 1로 하고 나머지 값을 0인 벡터를 만듭니다. 비슷한 단어를 반영할 수 있습니다.

2.4.3 스케일링 (Scaling)

: 머신러닝 알고리즘은 입력 숫자 특성들의 스케일이 많이 다르면 잘 작동하지 않습니다. (scikit-learn 활용)

1) min-max 스케일링

- 일반적으로 정규화(normalization)라고 합니다.
- 0 ~ 1사이의 값으로 스케일을 조정합니다.
- 조정하는 방법은 데이터에서 최솟값을 뺀 후 최댓값과 최솟값의 차이로 나누면 됩니다.

2) 표준화 (Standardization)

- 먼저 데이터에 평균을 빼서 평균을 0으로 만듭니다.
- 표준편차로 나눠서 결과 분포의 분산이 1이 되도록 합니다.
- min-max보다 이상치에 영향을 덜받습니다.

2.5 모델 선택과 훈련

2.5.1 훈련세트에서 훈련하고 평가

- 회귀 : RMSE, MAE, ANOVA Table
- 분류 : Confusion Matrix, ROC Curve

2.5.2 K-겹 교차검증 (K-Fold Cross Validation)

- 훈련세트를 폴드(Fold)라 불리는 10개의 서브셋으로 무작위로 분할합니다.
- 모델을 K번 훈련하고 평가합니다.
- 매번 다른 폴드를 선택해 평가에 사용하고 나머지 K-1개 폴드는 훈련에 사용합니다.
- K개의 평가 점수가 담긴 배열이 결과가 됩니다.

2.6 모델 튜닝

2.6.1 그리드 탐색 (Grid Search)

- 만족할만한 하이퍼파라미터 조합을 찾을 때까지 수동으로 하이퍼파라미터를 조정하는 것입니다.
- 비교적 적은 수의 조합을 탐구할 때 괜찮습니다.
- sklearn의 GridSearchCV를 사용하는 것이 좋습니다.

2.6.2 랜덤 탐색 (Random Search)

- 각 반복마다 하이퍼파라미터에 임의의 수를 대입해 지정한 횟수만큼 평가합니다.
- 랜덤 탐색을 1,000회 반복하도록 실행하면 하이퍼파라미터마다 각기 다른 1,000개의 값을 탐색합니다.
- 단순 반복 횟수를 조절하는 것만으로 하이퍼파라미터 탐색에 투입할 컴퓨팅 자원을 제어할 수 있습니다.

2.6.3 앙상블 방법 (Emsemble)

- 모델의 그룹이 최상의 단일 모델보다 더 나은 성능을 발휘할 때가 많습니다.

2.6.4 최상의 모델과 오차분석

- 각 변수별 중요도를 파악할 수 있는데 이를 바탕으로 덜 중요한 특성을 제외할 수 있습니다.
- 시스템이 특정 오차를 만들었다면 왜 그런 문제가 생겼는지 이해하고 문제를 해결하는 방법이 무엇인지 찾아야 합니다.

2.6.5 Test 데이터로 시스템 평가하기