2022.01.20, 머신러닝

→ 데이터 분석 종료 이후 모델링 진행시 (머신러닝 - 지도학습 사용)

1) Target의 회귀 / 분류

- 회귀
  - LinearRegression / KNN / 결정트리 / RF → 모델생성
  - MAE, MSE, MAPE, R2Score 등등 → 모델 분석
- 분류
  - KNN, 결정트리, LogisticRegression, RF → 모델생성.
  - accuracy_score, recall_score, precision_score, classification_report, confusion_matrix → 모델분석

★ 데이터를 보는 방법에 따라서

1. 문자 → 숫자 (가변수화)
2. 숫자
   - 연속형으로 생각.
   - 가변수화 진행.

} 성능이 좋아지는 방향을 생각해보기.

ex) Rank = 연속형 → 1~10 / 10~30
상위권 중위권

⇒ 성능의 향상이 될수있다.

✓ LinearRegression , 성능향상을 위함 X

변수와의 관계 → 어떻게 영향?

> 성능이상 : 알고리즘 재선택.

✓ 선형회귀, KNN : 정규화의 필요성↓.

⇒ 성능 알고리즘이 아님 / 주변으로 파악.

cf) 전처리 (Titanic)

⇒ Age의 전처리

기준 : 전체평균 ⇒ Mr, Miss, Ms 카테고리별 Age 평균.

① data['Title'] = data['Name'].str.extract('([A-z a-z]+).', expand=False)

(대괄호안의 A~z, a~z의 문자를 지정
+ : 대괄호 문자가 1개 이상.) → 1개의문자로 인식, 뒤의 '.' 까지포함.

⇒ 추출결과 : 'Mr.' 'Miss.' 'Mrs.'

② array = ['Mr', 'Miss', 'Mrs']

data.loc[data['Title'].isin(array) == False, ['Title']] = 'Others'

→ array에 해당하는 값이 없으면 Title에 others로 설정.

✓ KNN → R2_score와 같이 회귀성능측정 < MAE
R2_score.

⇒ 분류 알고리즘에는 사용 X.

+) 정규화 여부 결정 필요.

✓ Decisiontree (max_depth)

- 가지치기의 개념
- Overfitting과 너무낮은 accuracy_score 사이에서 적절히 균형.

✓ 선형회귀
KNN
결정트리

데이터를 본뒤 → 우리가 하고싶은것이 < 분류를 통해 명확하게 구분
회귀를 통해 해당값을 예측.

⇒ target을 설정할것.

선형회귀 : 성능 < 설명(데이터) → coef_

KNN : 주변 data를 통해 결정. → n_neighbor

결정트리 : max_depth 설정을 통해 grouping. → overfitting 주의