**<Chapter3. 회귀 알고리즘과 모델 규제>**

**-> Practice05, Practice06, Practice07**

**k-최근접 이웃회귀 선형회귀,다항회귀 다중회귀,규제,릿쏘회귀,릿지회귀**

**K최근접이웃 분류 알고리즘**

1. 예측하려는 샘플에 가장 가까운 샘플 K개 선택

2. 샘플들의 클래스들 확인

3. 다수 클래스를 새로운 샘플의 클래스로 예측

**K최근접이웃 회귀 알고리즘** (KNeighborsRegressor이용)

1. 예측하려는 샘플에 가장 가까운 샘플 K개 선택(클래스가 아닌 데이터)

2. 그 수치들의 평균을 구한다

3. 새로운 샘플의 예측값은 그 평균값임

**단점** => 회귀에서 100%정답은 불가능! 예측값, 타깃 모두 임의의 수치이기때문임 (결정계수사용)

k-최근접이웃알고리즘은 주변 샘플의 평균값을 구한 것이기 때문에 에러발생 가능 !

=> 해결책 : 가장 큰 농어가 포함되도록 훈련세트 다시 만들거나 다른 알고리즘 사용

**과대적합 :** 훈련세트에서 점수가 굉장히 좋은데 테스트 세트에서는 점수가 굉장히 낮음. 훈련 세트에만 잘 맞는 모델이 됨.

=> 모델을 더 단순하게 만든다 (이웃의 개수 K를 늘림)

**과소적합 :** 훈련세트보다 테스트세트의 점수가 높거나 두 점수 모두 낮음.

모델이 너무 단순하여 훈련 세트에 적절히 훈련되지 않은 경우.

=>모델을 더 복잡하게 만든다 (이웃의 개수 K를 줄임)

**선형회귀(Linear Regression) :** 지도학습 알고리즘 중 하나. feature여러개 가능. 농어의 length로 weight를 선형 상관관계로 알아낸다. 선형상관관계 없는데 선형회귀쓰면 정확도가 낮아진다. a(기울기)(종종계수)(coefficient) = lr.coef\_ b(절편)(가중치)(weight) = lr.intercept\_로 확인가능. 여러개의 feature를 보고 무게 예측하는 선형회귀 그릴 수 있음. feature의 개수가 n개면 세타값은 n+1임. feature은 3개 이상이면 시각화 불가. MSE(Meen Square Error)로 error줄여주는 세타값 찾을 수 있음. (m개의 data샘플을 실제 error가 어느정도인지 구해보고 제곱해서 더한후 m으로 나눠준다)

**다항회귀(polynomial Regression) :** 다중상관관계를 찾음. feature가 여러개 혹은 한 개. (feature가 여러개라서 다항회귀가 아니다!) 만약 feature가 2개라면 제곱해서 없던 feature만들어내고(fit) 2차 선형회귀를 적용시키면(transform) 2차 다항회귀와 같아짐. 해당 차수에 맞게 feature 늘어난 상태에서 선형회귀 적용시키면 해당 차수에 맞는 다항회귀 적용된 것!

**다중회귀(multiple regression) : 여러개의 특성을 사용한 선형 회귀. 1개의 특성을 사용했을 때 선형 회귀 모델이 학습하는 것은 직선이지만 2개의 특성을 사용하면 평면을 학습하게 됨!**

**판다스(pandas) : 데이터 분석 라이브러리. csv파일을 읽어들임.**

**사이킷런의 변환기(transformer) : 특성을 만들거나 전처리하기 위한 클래스**

**=>PolynomialFeatures사용.(transform하기전 fit필수)**

**규제(regularization) : 머신러닝 모델이 훈련 세트를 너무 과도하게 학습하지 못하도록 훼방하는 것. 선형회귀모델의 경우 특성에 곱해지는 계수의 크기를 작게 만든다. => 정규화 먼저 하고 규제 적용 해야함 (특성의 스케일이 정규화되지 않으면 그만큼 곱해지는 계수 값도 차이가 나게 되기 때문)**