CH5. 회귀
회귀 알고리즘
데이터를 계속 학습하면서 이 비용 함수가 반환하는 값(즉, 오류값)을 지속해서 감소시키고 최종적으로는 더 이상 감소하지 않는 최소의 오류값을 구하는 것
비용함수 = 손실함수(loss function)

경사하강법
비용 함수 RSS를 최소화하는 방법
반복적으로 비용 함수의 반환 값, 즉 예측값과 실제 값의 차이가 작아지는 방향성을 가지고 W 파라미터를 지속해서 보정
더이상 오류값이 작아지지 않으면 그 오류 값을 최소 비용으로 판단, 그 때의 W값을 최적 파라미터로 반환
핵심 : "어떻게 하면 오류가 작아지는 방향으로 W값을 보정할 수 있을까?" -> 미분값이 계속 감소하는 방향으로 순차적으로 W를 업데이트(W0, W1 각 변수에 편미분)

경사하강법

확률적 경사 하강법
전체 입력 데이터로 W가 업데이트되는 값을 계산하는 것이 아니라 일부 데이터만 이용해 W가 업데이트되는 값을 계산하므로 경사 하강법에 비해서 빠른 속도 보장

Ordinary Least Squares 기반의 회귀 계수 계산은 입력 피처의 독립성에 많은 영향을 받음
다중공선성(Multi-collinearity)
피처간의 상관관계가 매우 높은 경우 분산이 매우 커져서 오류에 매우 민감

회귀 평가 지표
실제값과 회귀예측값의 차이 갑승ㄹ 기반으로 한 지표가 중심
MAE
MSE
RMSE
R^2

scoring함수에 'neg_mean_absolute_error'를 적용해 음수값을 반환하는 이유
사이킷런의 scoring함수가 score값이 클수록 좋은 평가 결과로 자동평가하기 때문
그러나 실제값과 예측값의 오류 차이를 기반으로하는 회귀 평가 지표의 경우 값이 커지면 오히려 나쁜 모델이라는 의미이므로 이를 사이킷헌의 scoring함수에 일반적으로 반영하려면 보정이 필요 -> -1을 원래의 평가지표 값에 곱해 음수를 만들어 작은 오류 값이 더 큰 숫자로 인식

다항회귀
회귀가 독립변수의 단항식이 아닌 2차, 3차 방정식과 같은 다항식으로 표현되는 것

다항회귀도 선형회귀!
회귀에서 선형 회귀/비선형회귀를 나누는 기준은 회귀 계수가 선형/비선형인지에 따른 것이지 독립변수의 선형/비선형 여부와는 무관하다

비선형 함수를 선형 모델에 적용시키는 방법을 사용해 구현
사이킷런은 PolynomialFeatures로 피처를 변환한 후에 linearRegrssiong 클래스로 다항 회귀를 구현

다항 회귀를 이용한 과소적합 및 과적합 이해
차수가 높아질수록 과적합의 문제가 크게 발생

편향-분산 트레이드오프(Bias-Variance Trade off)
고편향성 : 지나치게 한 방샹성으로 치우친 경향
고분산성 : 지나치게 높은 변동성
일반적으로 편향과 분산은 한쪽이 높으면 한쪽이 낮아지는 경향
'골디락스 지점' : 편향을 낮추고 분산을 높이면서 전체 오류가 가장 낮아지는 지점
=> 편향과 분산이 서로 트레이드오프를 이루면서 오류 Cost값이 최대로 낮아지는 모델을 구축하는 것이 가장 효율적인 머신러닝 예측 모델을 만드는 방법

비용 함수
학습 데이터의 잔차 오류 값을 최소로 하는 RSS 최소화 방법과 과적합을 방지하기 위해 회귀 계수 값이 커지지 않도록 하는 방법이 서로 균형

규제
비용 함수에 알파 값으로 페널티를 부여해 회귀 계수 값의 크기를 감소시켜 과적합을 개선하는 방식
릿지 회귀 : L2 규제를 적용한 회귀
라쏘 회귀 : L1 규제를 적용한 회귀

엘라스틱넷 회귀
L2규제와 L1규제를 결합한 회귀
L2 규제를 라쏘 회귀에 추가한 것

선형 회귀의 경우 최적의 하이퍼 파라미터를 찾아내는 것 못지 않게 먼저 데이터 분포도의 정규화와 인코딩 방법이 매우 중요
타깃값을 일반적으로 로그 변환을 적용

로지스틱 회귀
선형 회귀 방식을 분류에 적용한 알고리즘
회귀가 선형인가 비선형인가는 독립변수가 아닌 가중치 변수가 선형인지 아닌지를 따름
선형 함수의 회귀 최적선을 찾는 것이 아니라 시그모이드 함수 최적선을 찾고 이 시그모이드 함수의 반환값을 확률로 간주해 확률에 따라 분류를 결정

회귀 트리
트리를 기반으로 하는 회귀 방식
리프 노드에 속한 데이터 값의 평균값을 구해 회귀
예측값을 계산