Supervised Learning

분류와 회귀

- 지도학습의 두 종류.

- 분류
  - » 미리 정의된 가능성 있는 여러 클래스 레이블 중 하나를 예측하는 것
  - » 두 개의 클래스로 분류하는 이진 분류와 셋 이상의 클래스로 분류하는 다중 분류

- 회귀
  - » 연속적인 숫자 또는 부동소수점(실수) 데이터를 예측하는 것

- 출력 값의 연속성 여부가 두 기법을 구분하는 중요한 기준
  - » 일반적으로 연속성이 있으면 회귀, 없으면 분류
  - » 양적 데이터는 회귀, 범주형 데이터는 분류

일반화, 과대적합, 과소적합

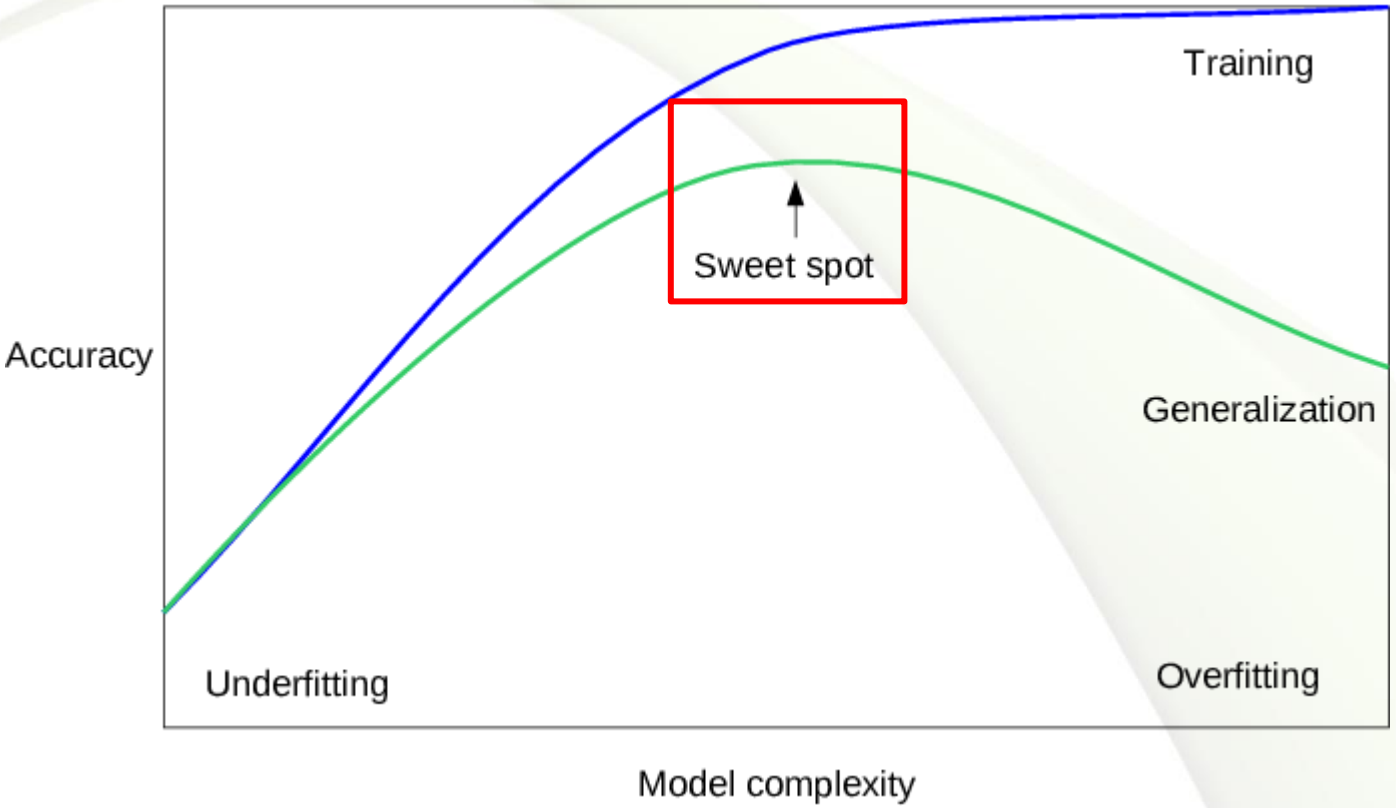
- 훈련 세트에서 테스트 세트로 일반화
  - » 모델이 처음 보는 데이터에 대해 정확하게 예측할 수 있게 되는 것
  - » 모델을 만들 때 가능한 정확하게 일반화하도록 구현해야 함

- 과대적합 (Overfitting)
  - » 모델이 훈련 세트에 너무 가깝게 맞춰져서 새로운 데이터에 일반화되기 어려운 경우
  - » 훈련 데이터는 잘 설명하지만 새로운 데이터에 대한 예측 정확도가 낮음

- 과소적합 (Underfitting)
  - » 모델을 지나치게 단순화해서 훈련 데이터와 테스트 데이터 모두에서 예측 정확도가 낮음

일반화, 과대적합, 과소적합

- 일반화 성능을 최대화 하는 모델을 찾는 것이 데이터 분석의 목표



- 일반적으로 데이터가 많으면 다양성을 강화하기 때문에 큰 데이터 셋을 사용하면 과대적합 없이 복잡한 모델을 만드는 것 가능

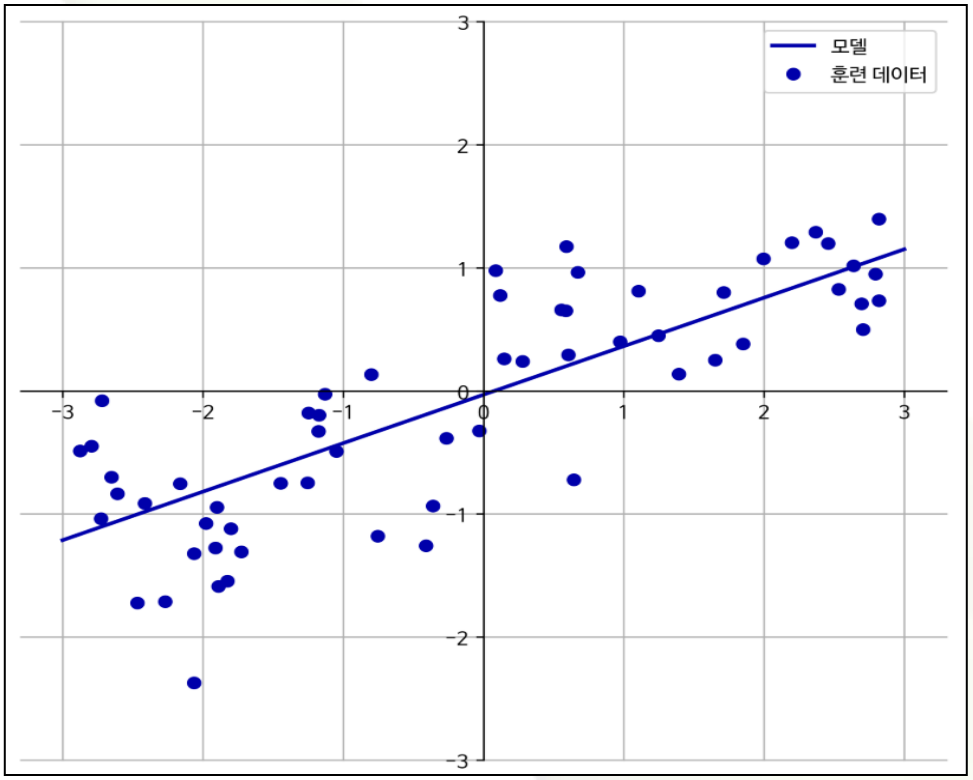
Regression

선형 모델

- 입력 특성에 대한 선형 함수를 만들어 예측 수행
  - » 여러 개의 독립변수와 한 개의 종속변수 사이의 상관관계를 모델링하는 기법

- 선형 회귀 모델의 일반화된 예측 함수

$$y_i = \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n,$$



- 구분
  - » 독립 변수의 개수에 따라
    - › 단일 회귀 / 다중 회귀
  - » 회귀 계수의 결합에 따라
    - › 선형 회귀 / 비선형 회귀

선형 회귀

- 가장 간단하고 오래된 회귀용 선형 알고리즘으로 최소 제곱법으로도 불림

- 예측과 훈련 세트에 있는 목적 변수 y 사이의 평균제곱오차(MSE)를 최소화하는 파라미터 w와 b를 추적
  - » 평균제곱오차 → (예측 값과 목적 변수 값의 차이)^2 / 데이터 개수

$$MSE = 1/n \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2$$

- 매개변수가 없는 것이 장점이지만 이로 인해 모델의 복잡도를 제어할 방법도 없음

- 모델이 과대 적합된 경우 복잡도를 제어할 수 있는 모델 필요
  - » 기본 선형 회귀 대신 릿지 회귀와 라소 회귀 모델을 널리 사용

회귀 평가 지표

평가 지표	설명
MAE	Mean Absolute Error 실제 값과 예측 값 차이의 절대값 평균
MSE	Mean Squared Error 실제 값과 예측 값 차이의 제곱값 평균
RMSE	Root Mean Squared Error MSE의 제곱근
R^2	0 ~ 1 사이의 값 전체 편차 제곱합 중에서 회귀 제곱합이 설명하는 비중

$$\sum(y_i - \bar{y})^2 = \sum(\hat{y}_i - \bar{y})^2 + \sum(y_i - \hat{y}_i)^2$$

$$\text{SST} = \text{SSR} + \text{SSE}$$

비용함수 최소화

- 회귀계수(weight, 가중치)로 구성되는 손실(RMSE)의 함수 정의

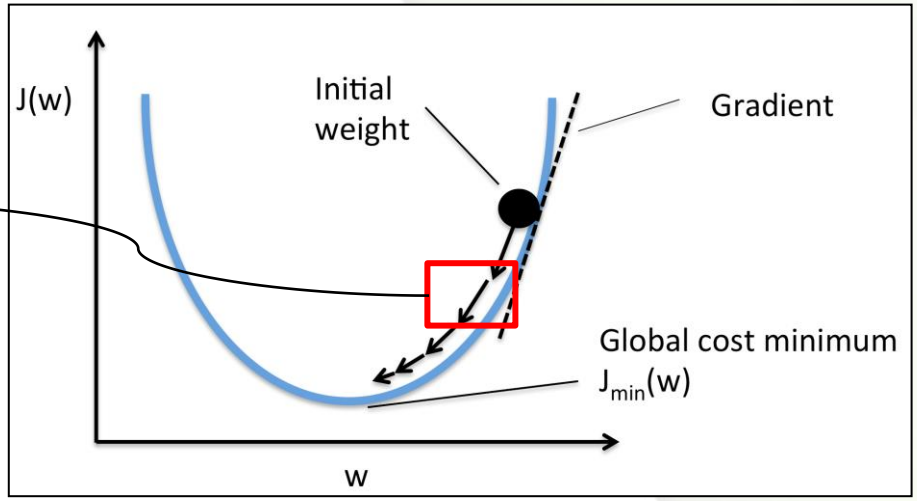
$$J(\boldsymbol{w}) = \sum_{i=1}^{N} \left(g\left(\boldsymbol{w}^T \boldsymbol{x}^{(i)}\right) - y^{(i)} \right)^2$$

- 회귀 모델 최적화 → 손실을 최소화하는 회귀계수를 찾는 문제

- 경사하강법(Gradient Descent) 최적화 알고리즘 적용
  - » 국지적 비용 최소화 또는 전역 비용 최소화에 다다를 때까지 점진적으로 가중치 갱신

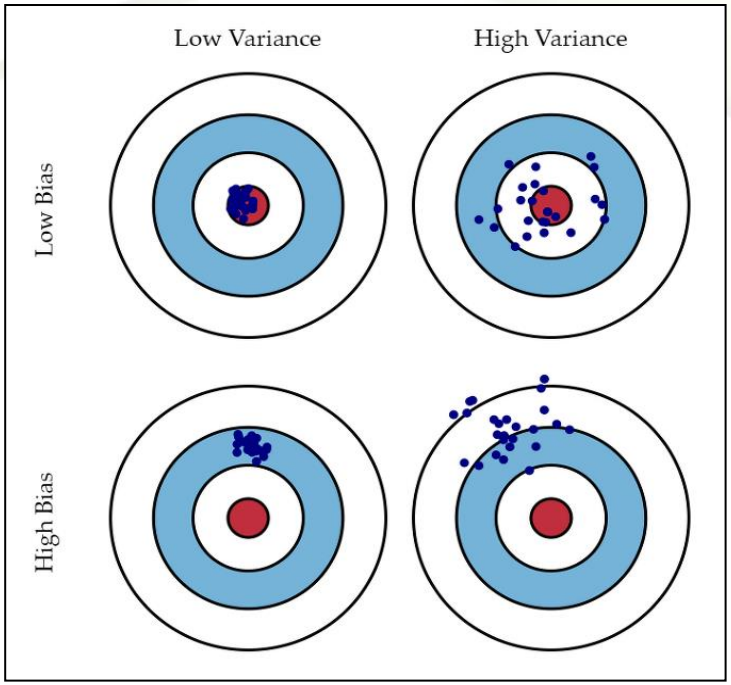
$$w_{ij} = w_{ij} + \Delta w_{ij}$$

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$$

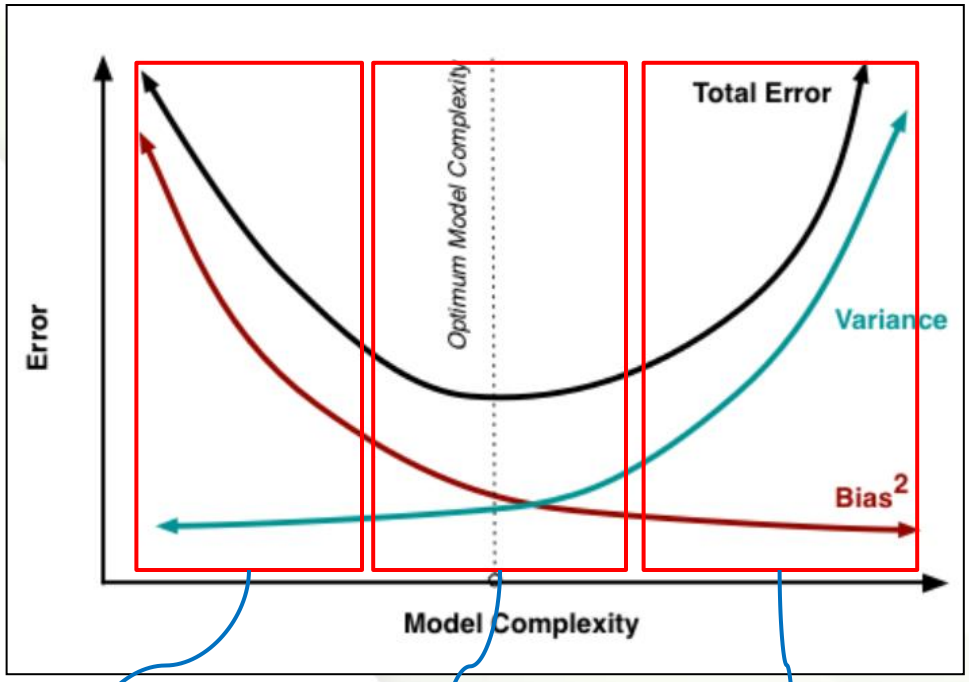


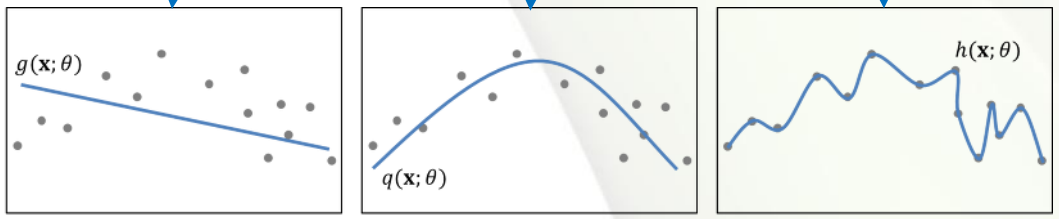
편향 - 분산

편향과 분산



편향 - 분산 트레이드오프





(a) Underfit (b) Ideal fit (c) Overfit

릿지(Ridge) 회귀

- 최소 제곱법에서 사용한 예측 함수를 사용하지만 릿지 회귀에서 가중치 선택은 훈련 데이터를 잘 예측하는 것뿐만 아니라 추가 제약 조건을 만족시키기 위한 목적도 포함
  - » 가중치의 절대 값을 최대한 작게 만드는 것 → w의 모든 원소를 0에 가깝게 만드는 것 (기울기를 작게 만드는 것) → 모든 특성이 출력에 주는 영향을 최소화
  - » 이런 제약을 규제(regularization)라고 하며 릿지 회귀에 사용되는 규제를 L2 규제라고 함

$$J(\theta) = \text{MSE}(\theta) + \alpha \frac{1}{2} \sum_{i=1}^{n} \theta_i^2$$

- scikit-learn의 Ridge 사용
  - » 알파 값을 크게 해서 규제의 강도를 높이면 일반화 성능이 향상됨

라쏘(Lasso) 회귀

- 릿지 회귀와 같이 가중치 계수를 0에 가깝게 만드는 작업을 하지만 릿지 회귀와 다른 방식으로 처리 → L1 규제

- L1 규제의 결과로 어떤 가중치 계수는 실제 0이 되기도 함
  - » 모델에서 완전히 제외되는 특성이 발생
  - » 특성 선택이 자동으로 이루어지는 것으로 해석할 수 있음
  - » 일부 계수를 0으로 만들면 모델을 이해하기 쉬워지고 모델의 중요한 특성을 구분할 수 있음

$$J(\theta) = \text{MSE}(\theta) + \alpha \sum_{i=1}^{n} | \theta_i |$$

- scikit-learn의 Lasso 사용

Elastic Net

- L1 규제와 L2 규제를 결합한 회귀
  - » L1 규제로 인해 규제 강도의 조정 값에 따라 회귀 계수가 급격히 변경하는 특성을 보완하기 위해 L2 규제 결합

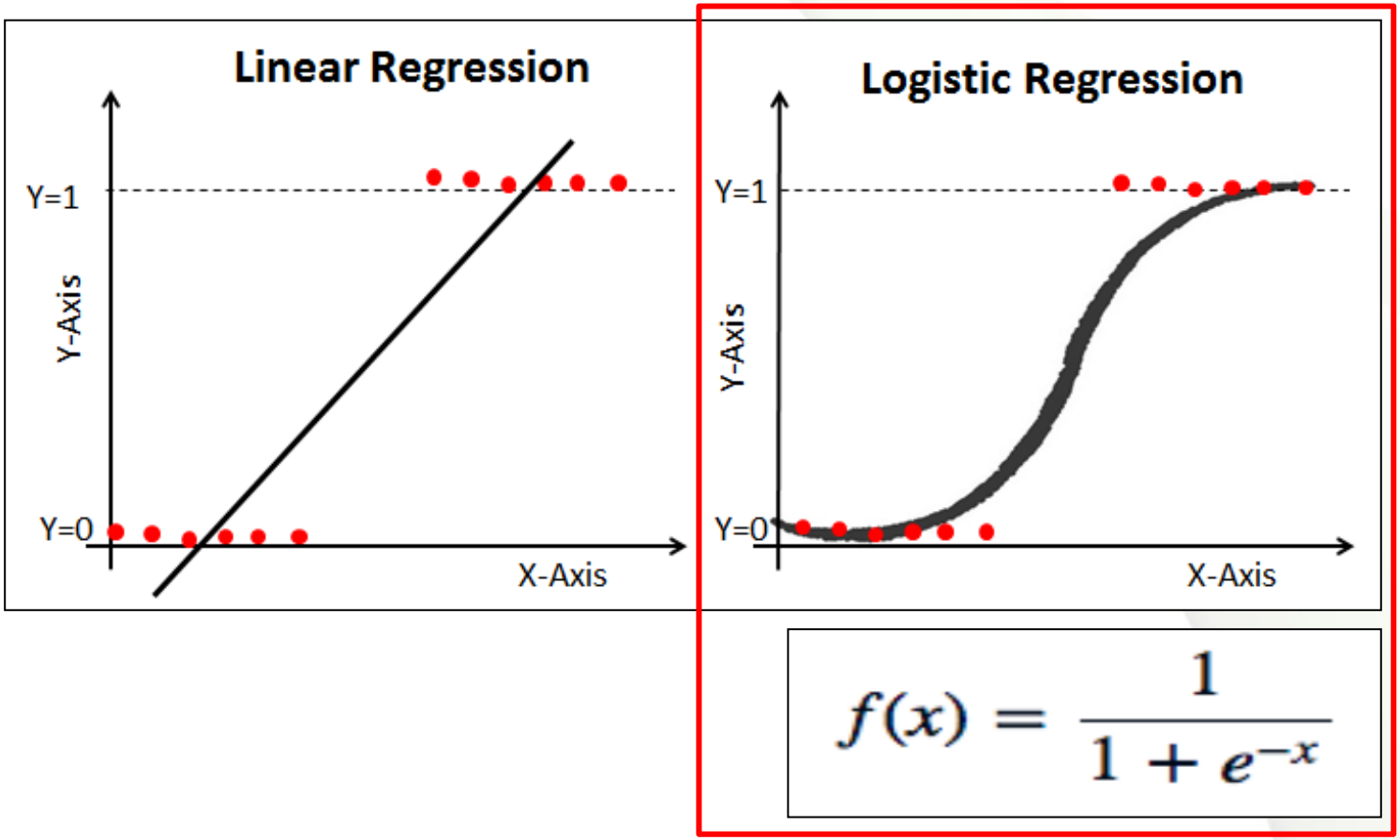
$$J(\theta) = \mathrm{MSE}(\theta) + r\alpha \sum_{i=1}^{n} |\theta_i| + \frac{1-r}{2}\alpha \sum_{i=1}^{n} \theta_i^2$$

- 상대적으로 수행 시간이 오래 걸리는 문제 노출

- scikit-learn의 ElasticNet 사용

로지스틱 회귀

- 선형 회귀 방식을 분류에 적용한 알고리즘
- 정확히 0 또는 1을 예측하는 대신 확률 생성
  - » 스팸 감지의 로지스틱 회귀 모델의 경우 모델이 특정 이메일 메시지에서 추론한 값이 0.932이면 이메일 메시지가 스팸일 확률을 0.932로 해석

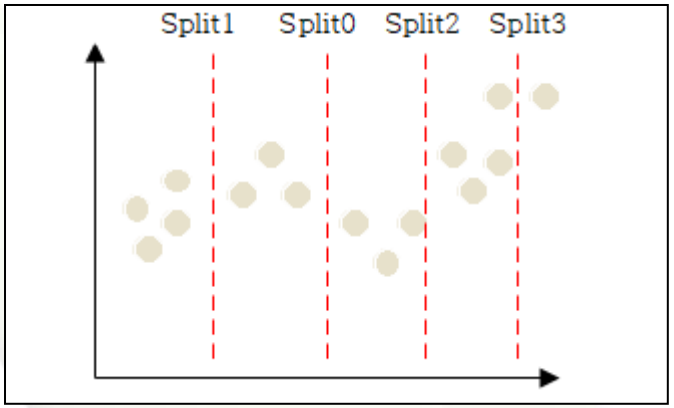


$$f(x) = \frac{1}{1 + e^{-x}}$$

회귀 트리

- 사용법과 분석 방법은 분류 트리와 비슷
  - » 리프 노드에 포함된 훈련 데이터의 평균 값이 출력 값

- 타겟 값의 균일도를 반영한 지니 계수를 고려하여 분할



- 알고리즘과 지원 클래스

알고리즘	회귀 Estimator 클래스	분류 Estimator 클래스
Decision Tree	Decision Tree Regressor	Decision Tree Classifier
Gradient Boosting	Gradient Boosting Regressor	Gradient Boosting Classifier
XGBoost	XGBRegressor	XGBClassifier
LightGBM	LGBMRegressor	LGBMClassifier