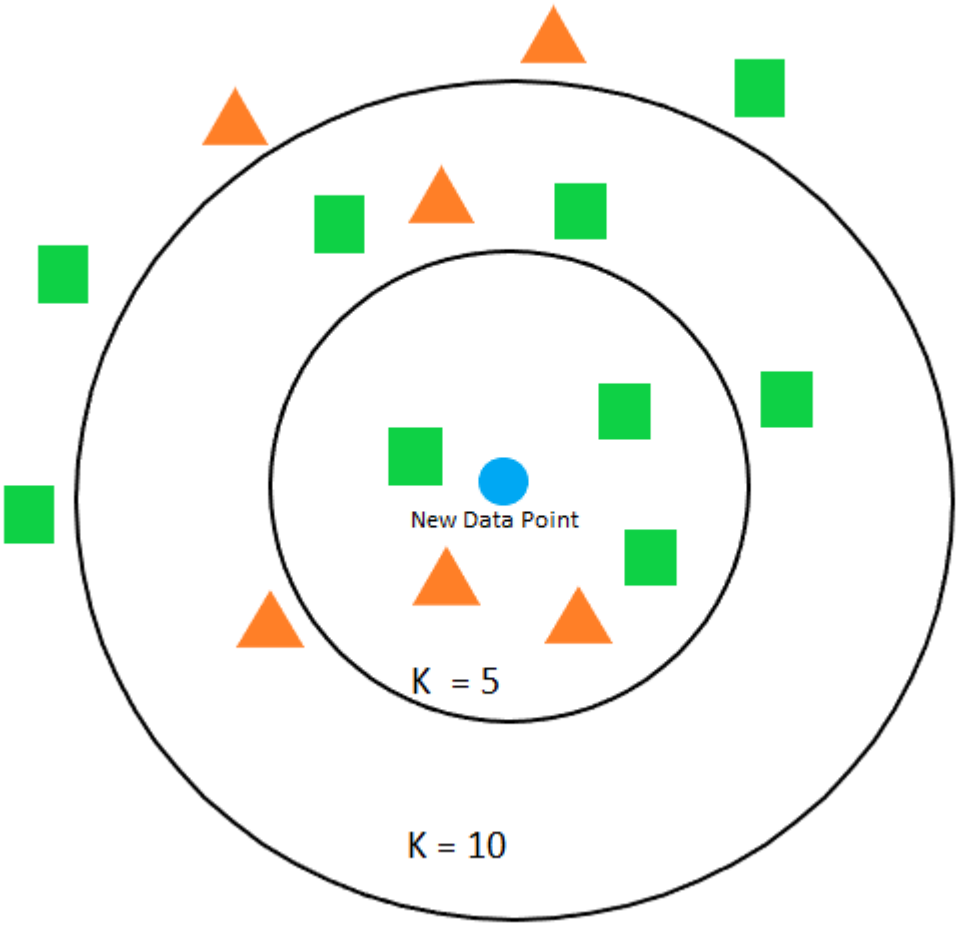
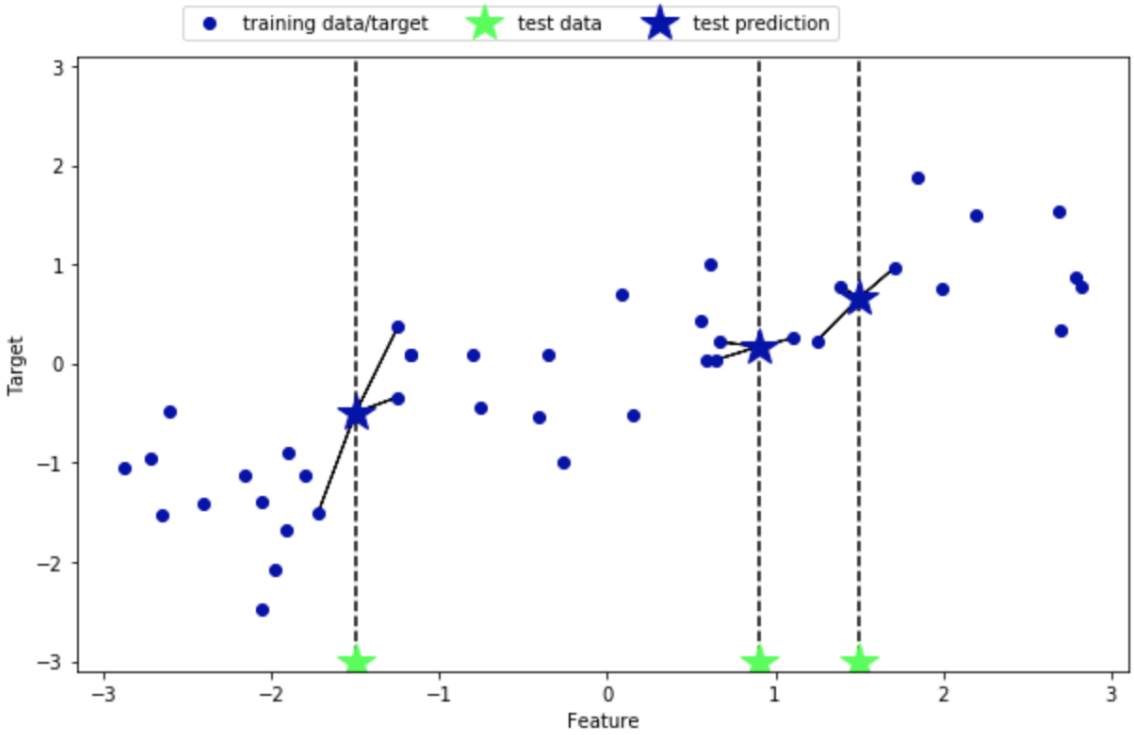
KNN (최근접 이웃)

목차

특징

- KNN 알고리즘: 새로운 데이터가 주어졌을 때, 기존 데이터 중 가장 가까운 k 개의 데이터를 정보로 새로운 데이터를 예측하는 방법론.
  classification 의 경우, k 개의 훈련 표본 사이에서 가장 많이 나온 분류명을 할당하고 regression 의 경우(보통) k 개의 훈련 표본들의 평균이 예측값이 됨





- Instance base: 내부에서 모델을 생성하여 학습하지 않음. 데이터 수집->계산->예측
- Memory base: 모든 데이터를 메모리에 저장해서 이를 기준으로 다음 예측을 실행
- Lazy 한 모델: 미리 학습하지 않고 새로운 데이터의 task 요청이 올 때마다 계산하고 분류 수행
- 위의 특징들로 인해 메모리를 많이 사용하고, 관측치가 많아질수록 연산량이 많아짐(관측치-학습데이터간 거리를 모두 계산해야함)

Hyperparameter

- K: 사용자 정의 상수 / 인접한 데이터를 몇 개까지 탐색할 것인가?
  >이진분류의 경우 홀수의 적은 k 개 or 부트스트랩방식
  1. 최선의 K 는 데이터 의존적임
  2. K 값이 커지면 노이즈의 영향 감소 but 경계 불분명(underfitting)
  3. K 값이 작아지면 training error 은 낮아지지만 overfitting
  >k 값을 선정하기 위해서 분류/회귀 모델은 아래와 같은 방식으로 평가함(그치만 결국
  노가다)

$$MisclassError_k = \frac{1}{k}\sum_{i=1}^{k} I(c_i \neq \hat{c}_i) \;\; for \;\; k = 1,2,\ldots,k^*$$

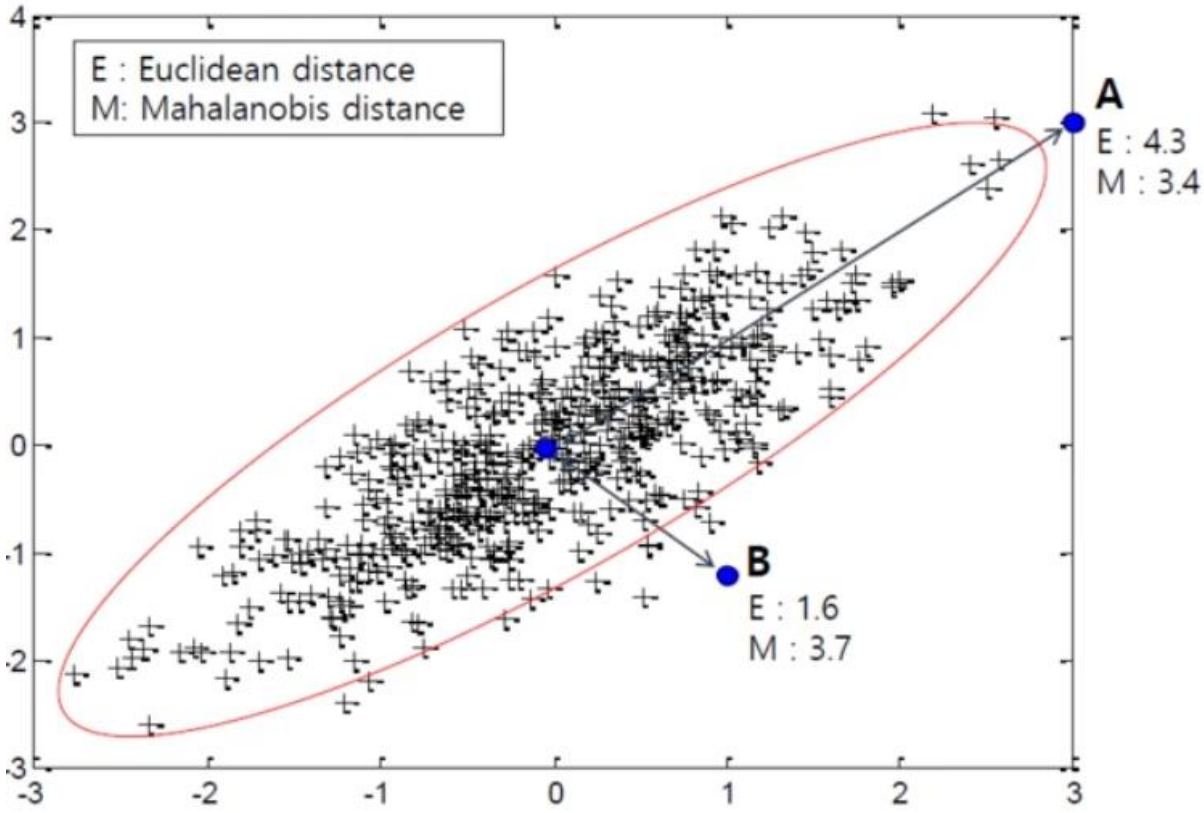
$I(\cdot)$: Indicator Function

$$SSE_k = \sum_{i=1}^{k} (y_i - \hat{y}_i)^2 \;\; for \;\; k = 1,2,\ldots,k^*$$

- distance measures: 차원이 많아도 거리를 구하면 결국 1x1
  1. Euclidean
  Distance

$$d_{(X,Y)} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

  2. Manhattan Distance
  3. Mahalanobis Distance: 변수 내 분산, 변수 간 공분산을 모두 반영 --> 전체 데이터에서 유사도가 높은 데이터는 거리가 가깝게 나오도록 보정해줌



$$d_{Mahalanobis\,(X,Y)} = \sqrt{(X-Y)^T \Sigma^{-1} (X-Y)},$$

Σ^{-1}: inverse of covariance matrix

4. Correlation Distance: 데이터간 correlation 을 거리측도로 사용하여 데이터 패턴의 유사도를 반영할 수 있음

weighted KNN

- KNN 은 빈번한 항목의 데이터가 새로운 데이터 예측에 큰 영향을 미침 --> weighted KNN
- 가중치: 가까운 이웃일수록 더 먼 이웃보다 많이 기여하도록 가중치(일반적으로 1/d)

$$\hat{y}_{new} = \frac{\sum_{i=1}^{k} w_i y_i}{\sum_{i=1}^{k} w_i} \qquad where\ w_i = \frac{1}{d_{(new,x_i)}{}^2}$$

$$\hat{c}_{new} = \max_c \sum_{i=1}^{k} w_i I(w_i \in c)$$

고차원 데이터

- 고차원의 데이터의 경우 차원 축소 수행(pca, lda, cca 를 전처리 과정으로 사용)