D&A
ML Session 4차시
분류 모델(SVM, Naïve Bayes, KNN, Decision Tree)

2022 / 09 / 27
D&A 운영진 이예진

CONTENTS.

01. SVM

SVM이란?

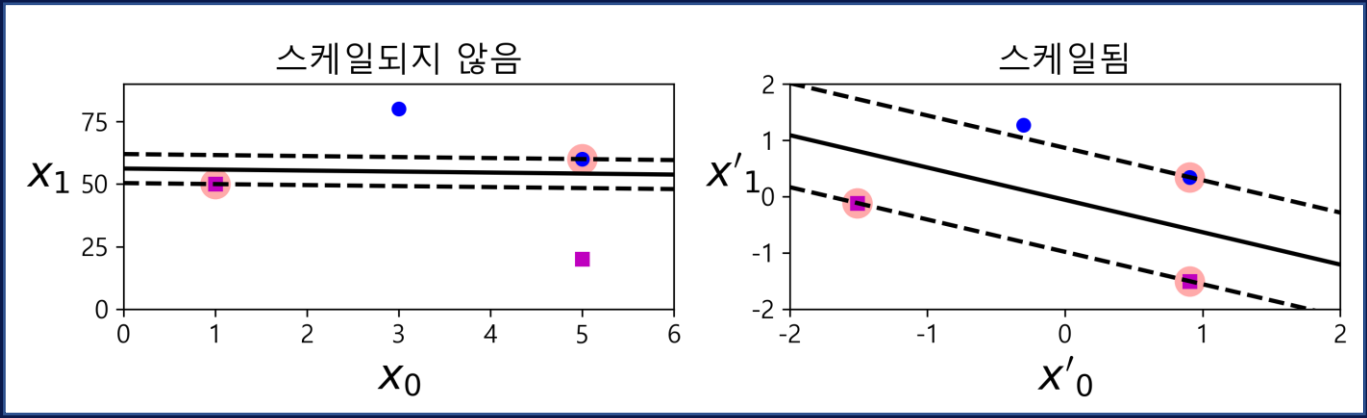
Support Vector Machine

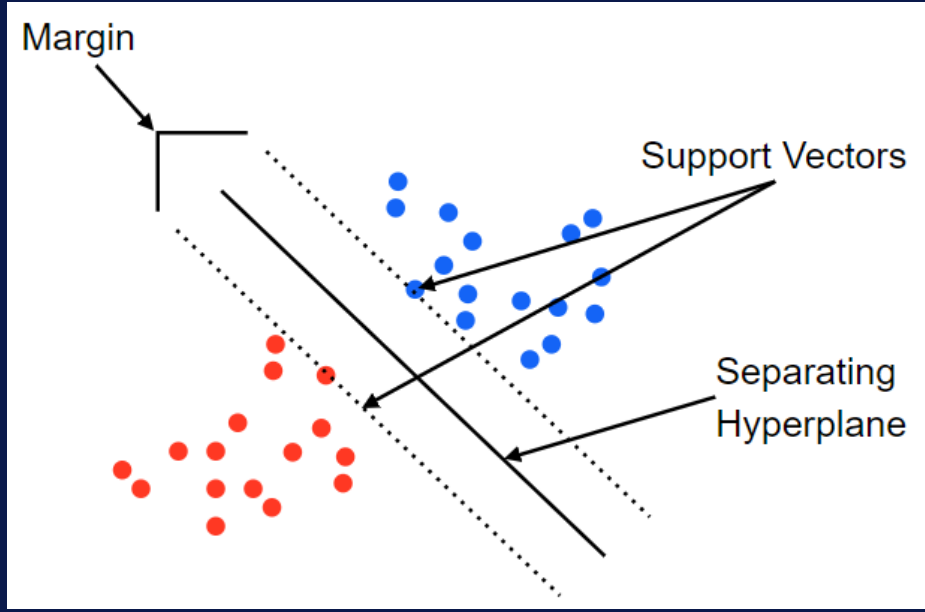
선형 분류, 비선형 분류, 회귀, 이상치 탐색에도 사용할 수 있는 다목적 머신 러닝 모델

클래스가 다른 데이터들을 가장 큰 마진(margin)으로 분리해내는 선 또는 면을 찾아내는 것

기본 아이디어 : 클래스 사이에 가장 폭 넓은 도로를 찾는 것

특히 복잡한 분류 문제에 잘 들어맞으며 작거나 중간 크기의 데이터셋에 적합
특성이 비슷하고 스케일이 비슷할 때 성능이 좋다



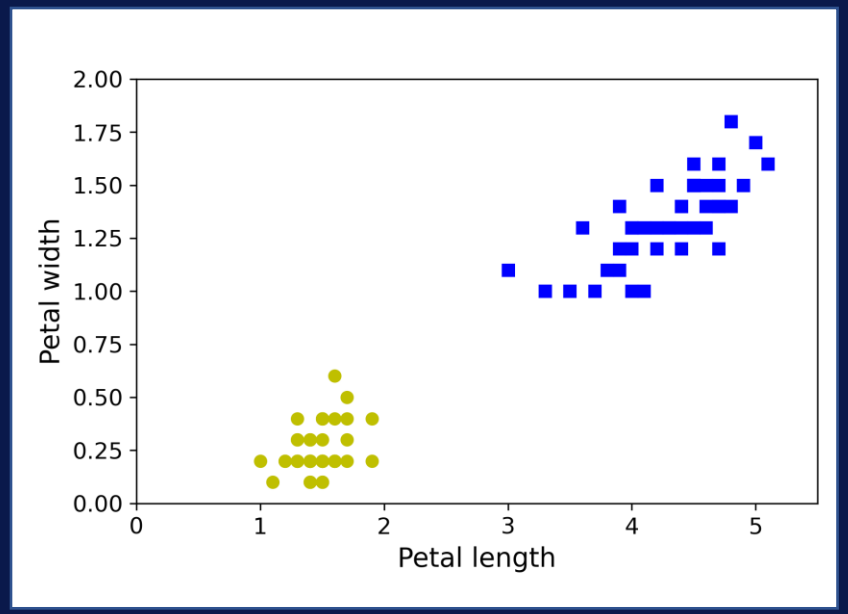


01. SVM

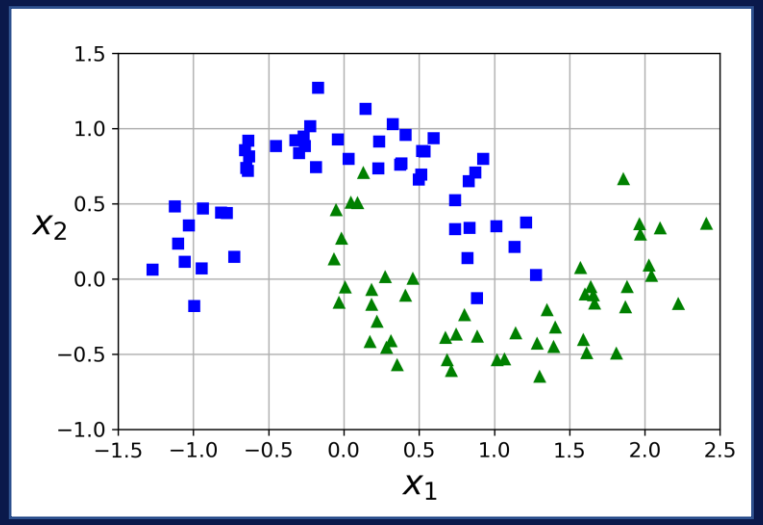
SVM 종류

선형 SVM
비선형 SVM -> 다항 특성을 사용한 선형 SVM / Kernel-SVM

선형 SVM 사용하는 경우



비선형 SVM 사용하는 경우



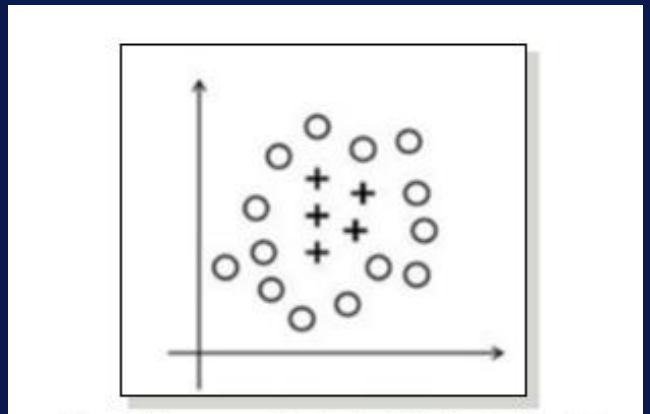
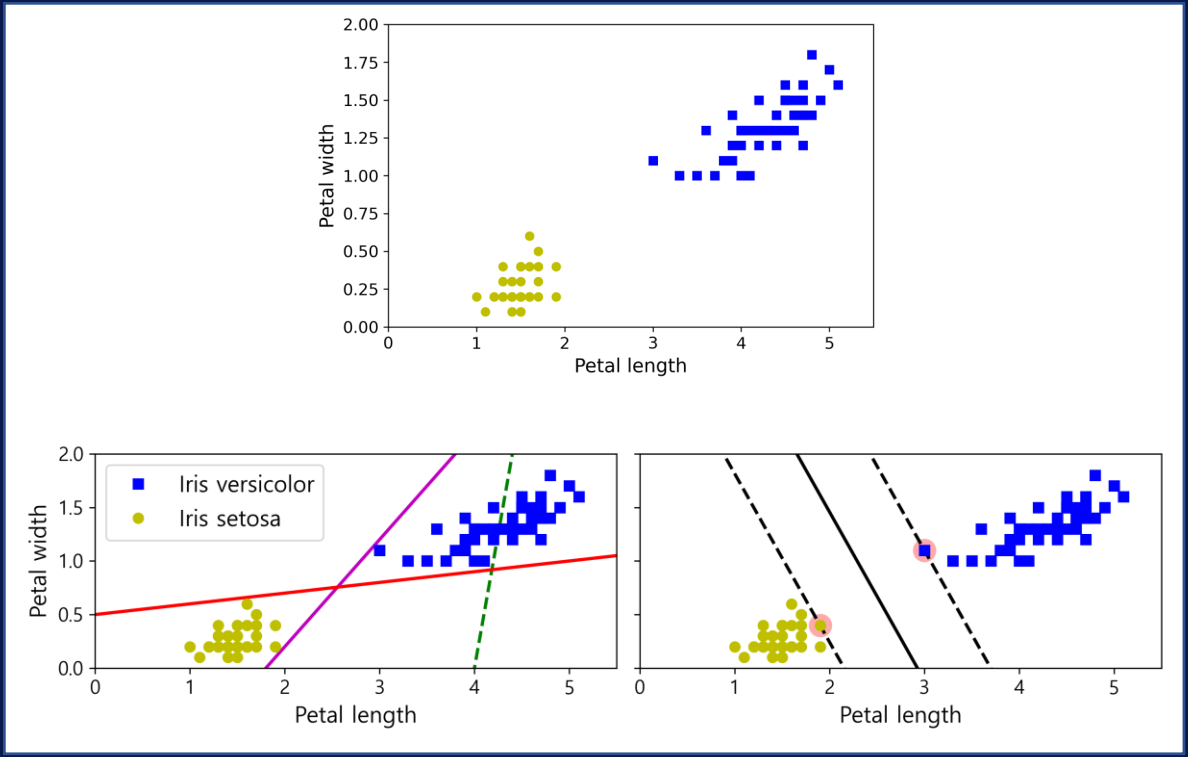


그림7. 선형 SVM으로는 제대로 분류할 수 없는 상황

01. SVM

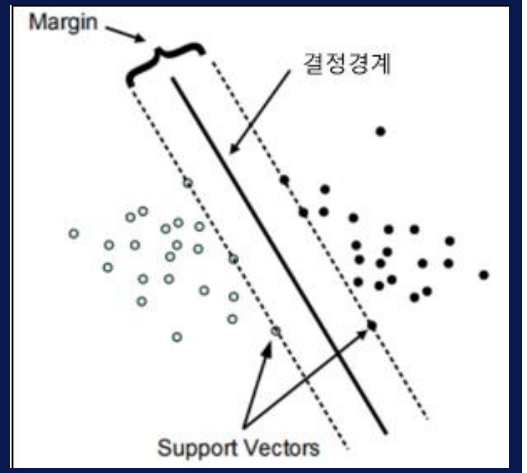
선형 SVM 분류

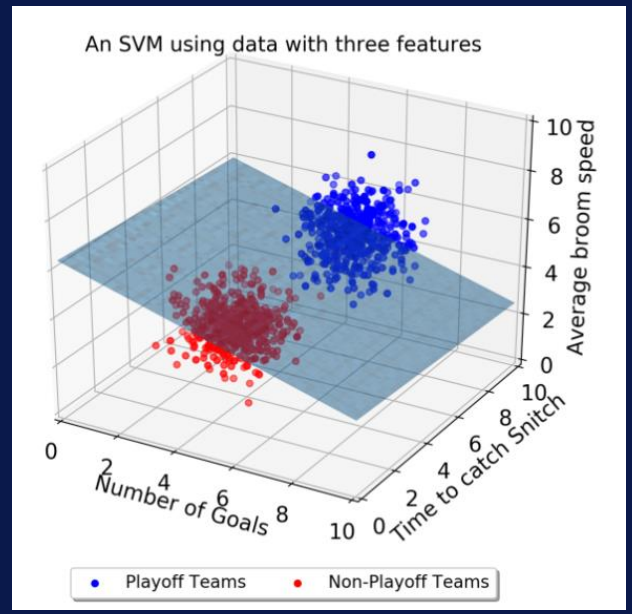
클래스를 선형적으로 구분



결정 경계 [Decision Boundary]

분류를 위한 기준 선



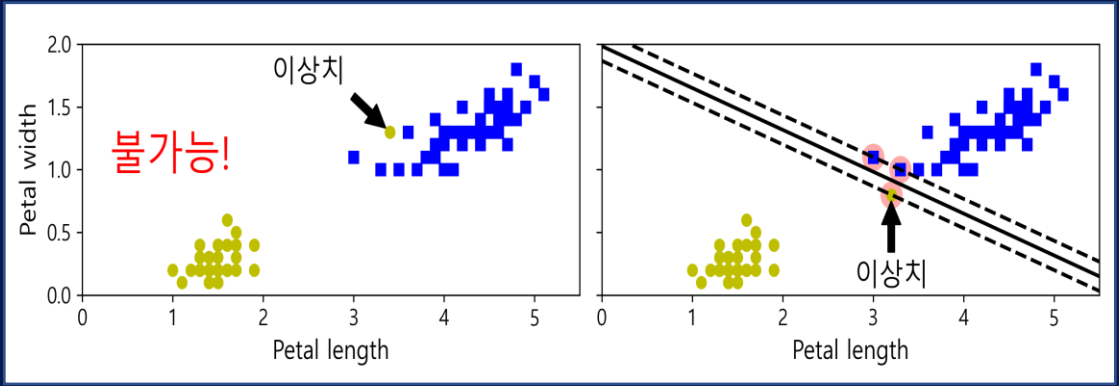


01. SVM

Hard Margin

모든 샘플이 도로 바깥쪽에 올바르게 분류가 된 경우

문제점
- 데이터가 선형적으로 구분 될 수 있어야 제대로 작동
- 이상치에 민감

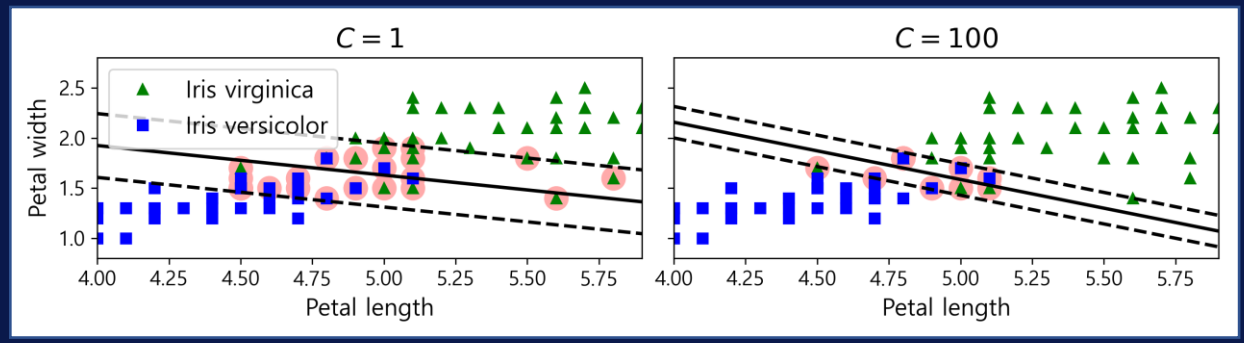


Soft Margin

좀 더 유연한 모델
도로의 폭을 가능한 넓게 유지하는 것과 마진 오류 사이에 적절한 균형을 잡는 것

하이퍼파라미터 C를 조절해서 마진오류를 조절할 수 있다.

마진 오류[margin violation] : 샘플이 도로 중간이나 반대쪽에 있는 경우



01. SVM

비선형 SVM 분류

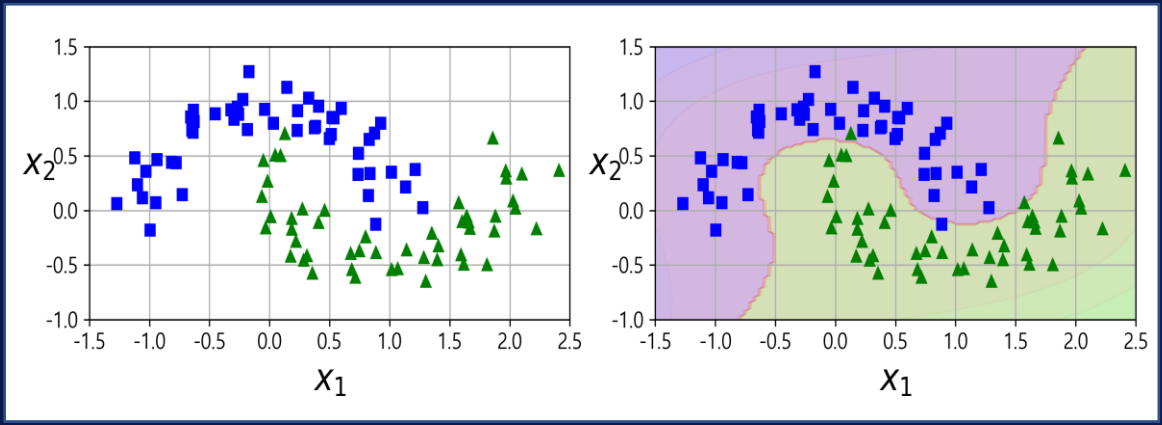
다항 특성을 활용한 선형 SVM

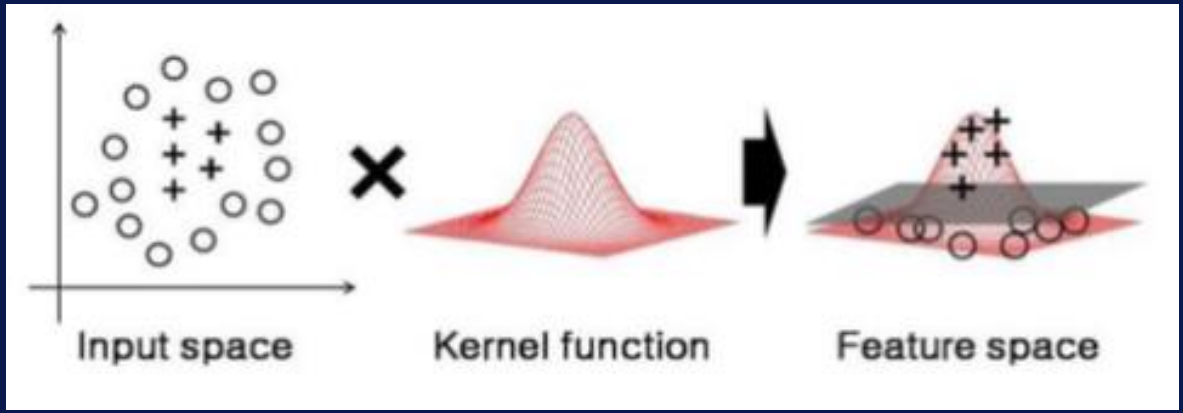
PolynomialFeatures을 이용

Kernel-SVM

Kernel 기법: 주어진 데이터를 고차원 특징 공간으로 사상해주는 것

다항식 커널, 가우시안 RBF 커널, 등이 존재





01. SVM

Hyperparameter

선형 SVM은 LinearSVC() or SVC(kernel="linear")로 사용 가능

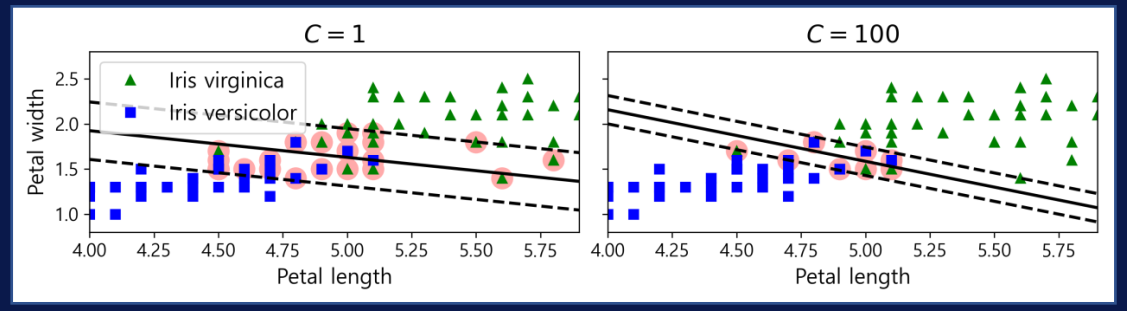
SVC : 분류
SVR : 회귀

SVC(*, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', break_ties=False, random_state=None)

Parameter	설명
C	float, default=1.0 마진 오류를 얼마나 허용할 것인지 지정. 클수록 hard margin, 작을 수록 soft margin. 반드시 양수
kernel	'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' 알고리즘에 사용할 커널 유형을 지정
degree	int, default=3 다항식 커널 함수('poly')의 차수. 다른 커널에서는 무시됨
gamma	'scale', 'auto' (or float) 'rbf', 'poly' 및 'sigmoid'에 대한 커널 계수 결정 경계를 얼마나 유연하게 그릴지 결정. 클수록 overfitting될 가능성이 높아짐
coef0	float, default=0.0 커널 함수의 독립 항. 'poly'와 'sigmoid'에서만 적용됨

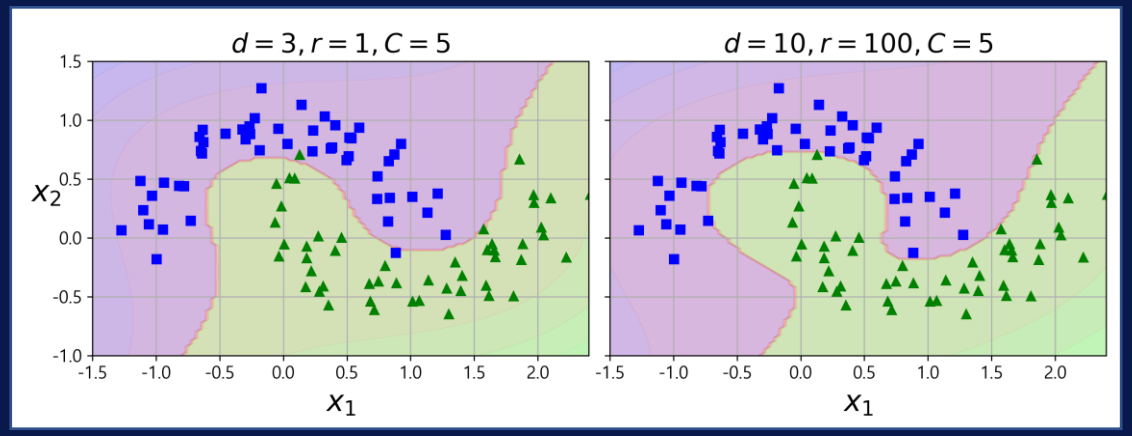
01. SVM

C 조정



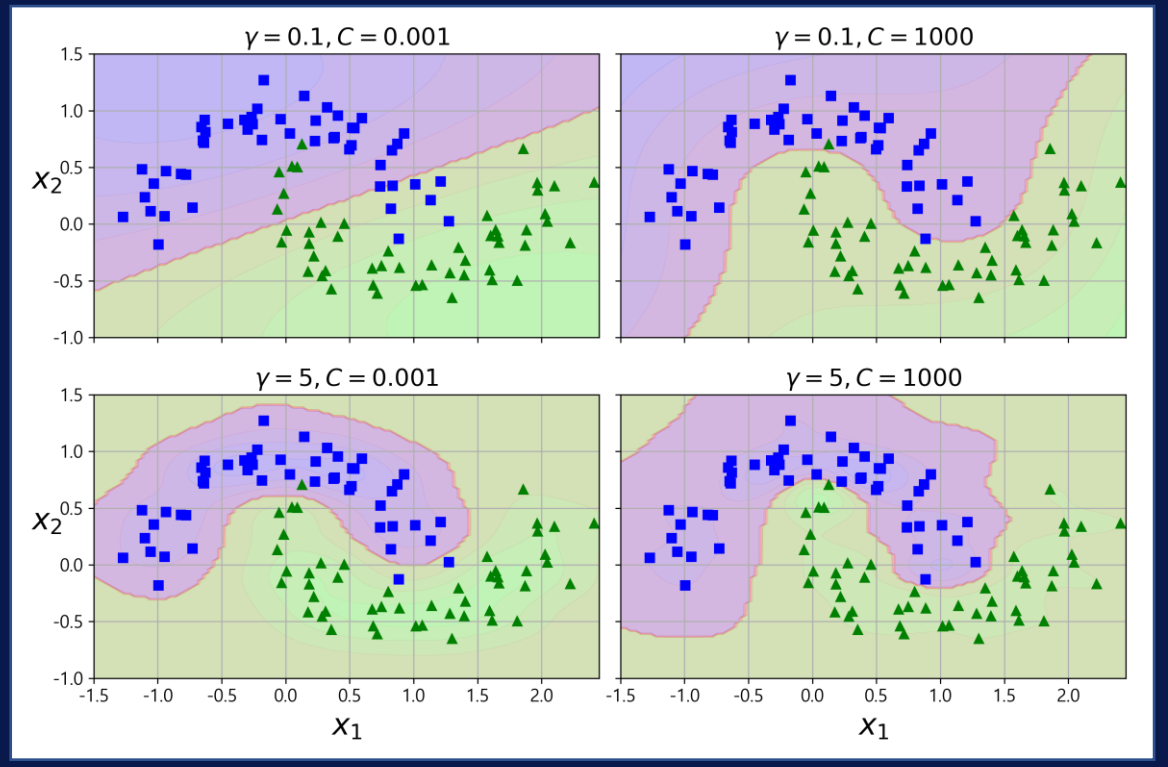
kernel="poly"로 했을 경우,

degree & coef0 조정



kernel="rbf" 로 했을 경우,

gamma & C 조정



01. SVM

장점

- 범주나 수치 예측 문제에 사용 가능
- 오류 데이터에 대한 영향이 적다
- 과적합 되는 경우가 적다
- 신경망보다 사용하기 쉽다

단점

- 최적의 모델을 찾기 위해 커널과 모델에서 다양한 테스트가 필요
- 여러 연산이 필요하고 입력 데이터셋이 많을 경우 학습 속도가 느리다
- 해석이 어렵고 복잡한 블랙박스 형태로 되어있다

02. Naïve Bayes

Naïve Bayes?

특성들 사이의 독립을 가정하는 “Bayes 정리”를 적용한 확률 분류기의 일종

Bayes 정리

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

Naïve의 사전적 의미 :

“(경험, 지식 부족 등으로) 순진해 빠진, (모자랄 정도로) 순진한”

Naïve가 붙은 이유 : 특징(feature)들이 서로 “확률적으로 독립”이라는 가정이 들어가 있기 때문

대중적으로 사용되는 분야

단순화 시켜 쉽고 빠르게 판단을 내릴 때 주로 사용됨.
Ex) 문서 분류, 질병 진단, 스팸 메일 분류 등에 많이 사용

02. Naïve Bayes

장점

- 간단하고, 계산 비용이 작아 빠른 모델
- 큰 데이터셋에 적합
- 연속형보다는 이산형 데이터에서 성능이 좋음
- Multiple class 예측에도 사용 가능

단점

- feature 간의 독립성이 있어야 함

실제 데이터에서 feature들이 독립인 경우는 드물다.

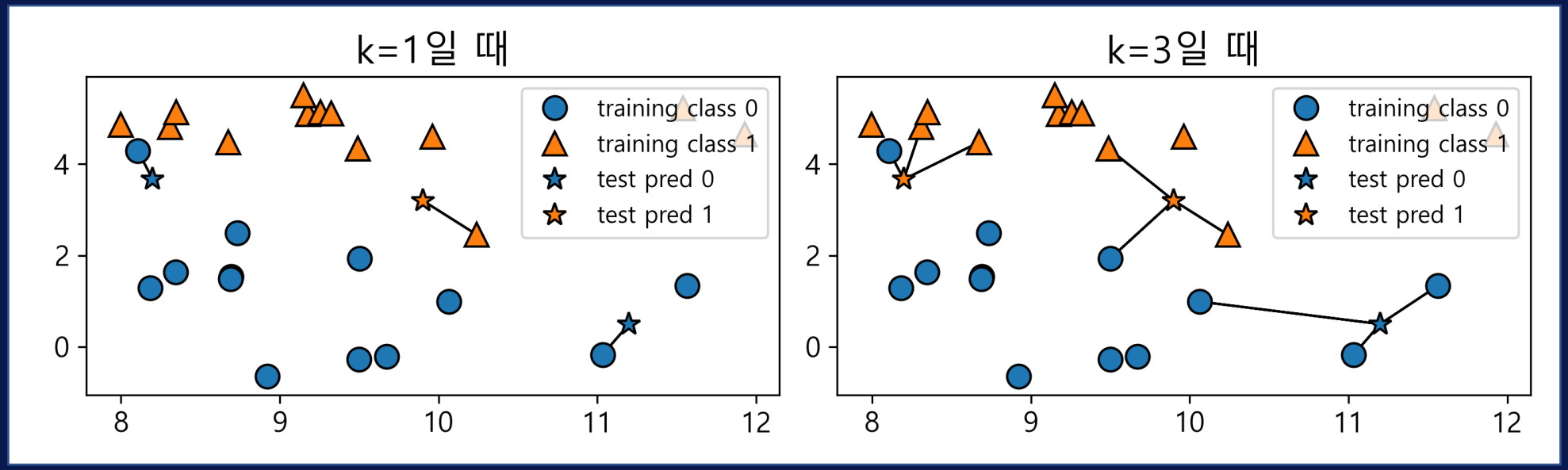
03. KNN

KNN이란?

K-Nearest Neighbor, K-최근접 이웃

굉장히 직관적이고 간단

새로운 데이터가 주어지면 가장 가까운 k개 이웃의 데이터를 살펴본 뒤, 더 많은 데이터가 포함되어 있는 범주로 데이터를 예측하는 방식



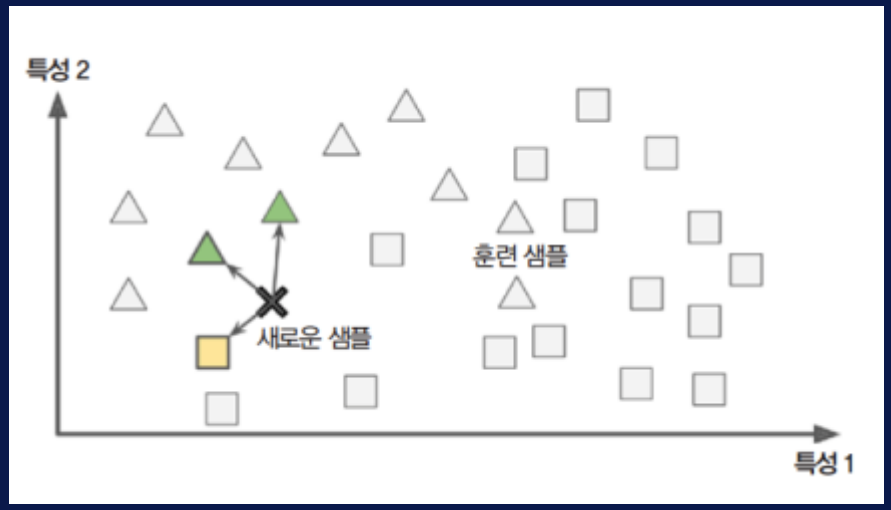
03. KNN

Instance-based Learning 사례 기반 학습

단순히 학습된 데이터들을 기억하는 것

1. 시스템이 훈련 데이터를 기억
2. 새로운 데이터가 들어오는 경우, 학습된 데이터와 새로운 데이터의 유사도 측정
3. 입력된 새로운 데이터를 가장 유사도가 높은 학습 데이터의 클래스로 분류

Ex. KNN, kernel machines, RBF networks

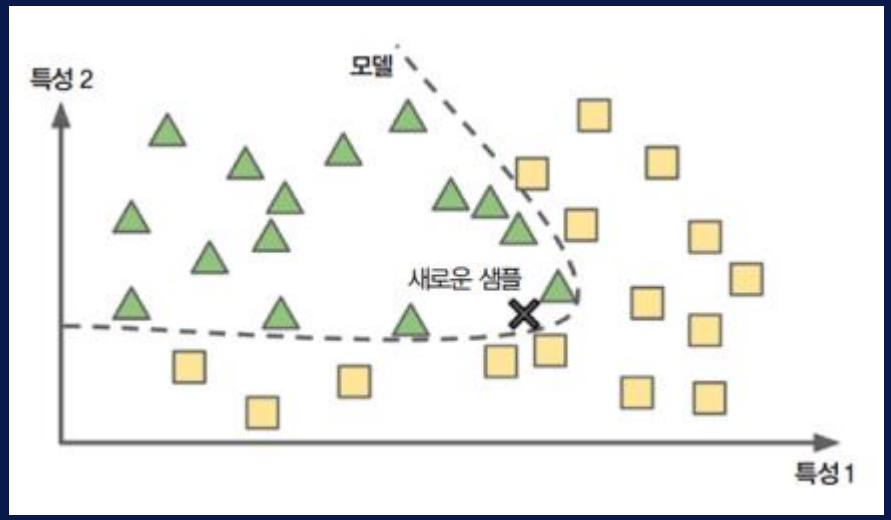


Model-based Learning 모델 기반 학습

데이터로부터 모델을 생성&이용하여 새로운 데이터를 예측

1. 데이터셋을 분석, 적합한 모델 선택
2. 훈련 데이터로 모델 훈련
3. 학습된 모델에 새로운 데이터를 적용해 예측

Ex. SVM, DecisionTree, LinearRegression 등



03. KNN

Hyperparameter

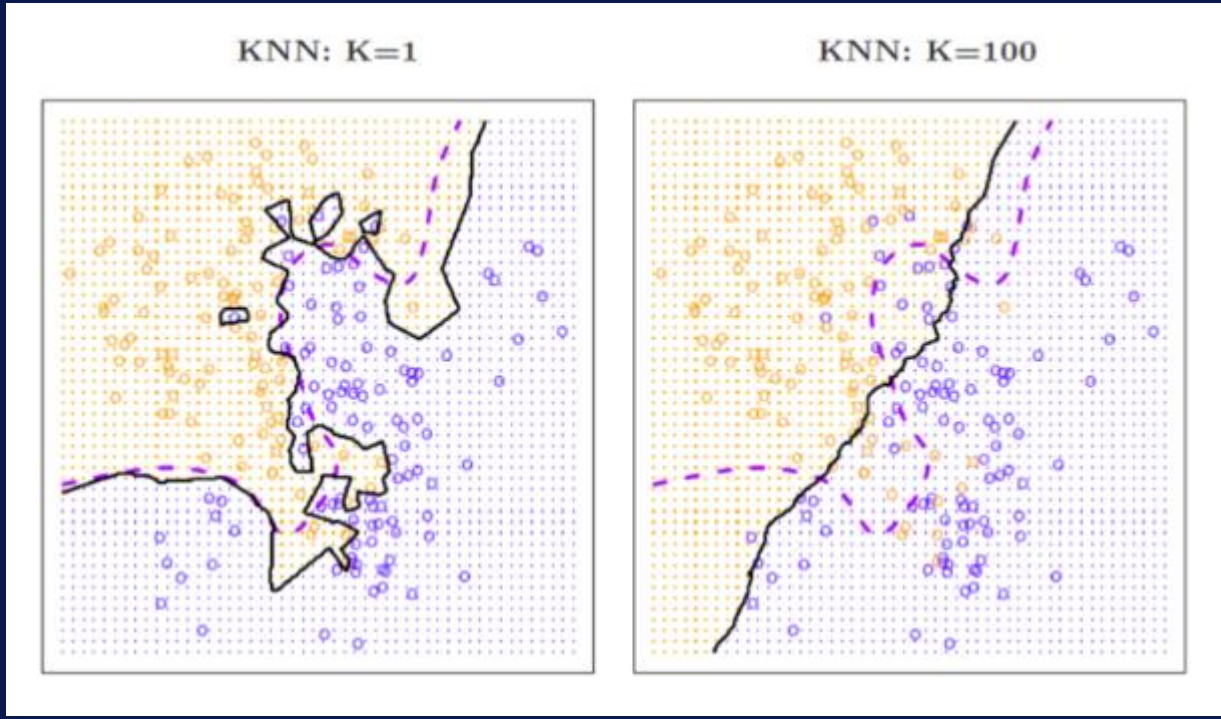
KNeighborsClassifier(n_neighbors=5, *, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None)

Parameter	설명
n_neighbors	int, default=5 이웃수, k
weights	예측에 사용되는 가중치 함수 (callable:사용자가 설정한 함수 이용) 'uniform': 균일한 가중치. 각 이웃의 모든 포인트는 동일한 가중치 부여 'distance' : 거리의 역수로 가중치 부여. 이 경우 포인트의 가까운 이웃이 멀리 있는 이웃보다 더 큰 영향을 미침
algorithm	가장 가까운 이웃을 계산하는 데 사용되는 알고리즘 'auto', 'ball_tree', 'kd_tree', 'brute'
p	int, default=2 Minkowski 메트릭에 대한 검정력 매개변수 p=1은 맨하튼 거리. p=2는 유클리드 거리를 사용하는 것과 같음
metric	str or callable, default='minkowski' 거리 계산에 사용할 미터법. 디폴트 값에서 p = 2일 때 표준 유클리드 거리 유효한 메트릭 값 참조 | scipy.spatial.distance & distance_metrics

03. KNN

k 조정

n_neighbors



03. KNN

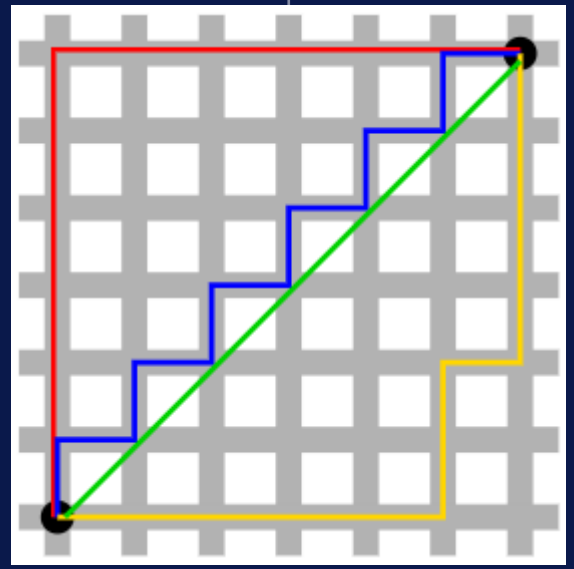
Euclidean Distance 유클리드 거리

두 관측치 사이의 직선 최단거리를 의미

가장 흔히 사용하는 거리 척도

공식

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_p - y_p)^2}$$



초록색 선 = 유클리드 거리

Manhattan Distance 맨해튼 거리

(= Taxi cab Distance)

A에서 B로 이동할 때 각 좌표측 방향으로만 이동할 경우에 계산되는 거리

아래의 세 경로(빨,파,노)는 맨해튼 거리 기준으로는 같은 거리

공식

$$d(x,y) = \sum_{i=1}^{p} |x_i - y_i|$$

빨간색, 파란색, 노란색 선 = 맨해튼 거리

03. KNN

Minkowski Distance 민코프스키 거리

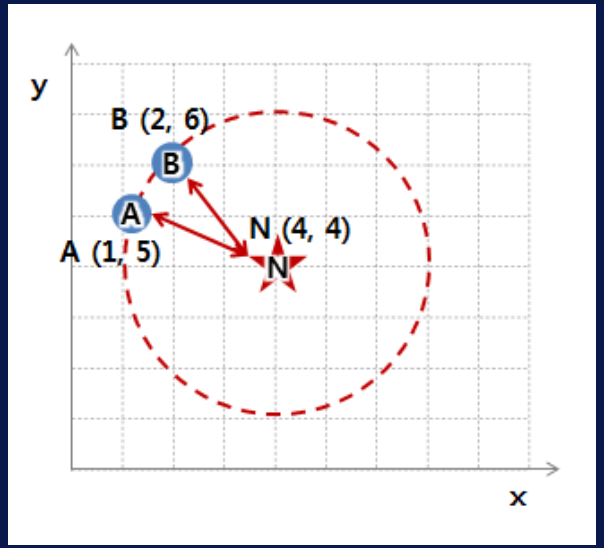
m차원 민코프스키 공간에서의 거리

m=1 일 때, 맨해튼 거리와 같고

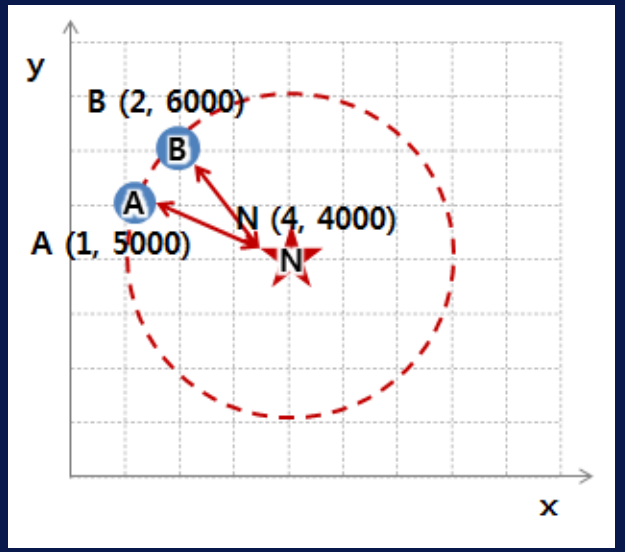
m=2 일 때, 유클리드 거리와 같음

공식

$$d(x,y) = \left(\sum_{i=1}^{n} | x_i - y_i |^m\right)^{\frac{1}{m}}$$



A-N간의 유클리드 거리는 3.162
B-N간의 유클리드 거리는 2.828로
B가 더 가깝다.



만약 단위가 위 그림처럼 바뀐다면
A-N간의 유클리드 거리는 1000.004
B-N간의 유클리드 거리는 2000.001
A가 더 가깝다.

이렇게 변수 별 단위가 무엇인지에 따라 가까운 순서가 달라지게 된다.

가까움의 정도가 달라지면 KNN의 분류 결과도 달라지기 때문에

KNN을 사용할 때는 데이터에 표준화를 해야 한다

03. KNN

장점

- 알고리즘과 원리가 간단하여 구현하기 쉽다
- 알고리즘의 이해하기 쉬운 모델
- 수치 기반 데이터 분류 작업에서 성능이 좋다
- 모델 구축이 빠르다

단점

- k와 어떤 거리척도가 분석에 적합한지 불분명해 데이터 각각의 특성에 맞게 사용자가 임의로 선정해야한다
- feature의 수가 크면 계산량이 많아지고, 변수간 거리가 멀어지기 때문에 정확도가 떨어진다.
- 학습 데이터셋이 커질 수록 느려진다
- sparse dataset과 같은 특정 데이터 셋에서는 잘 작동하지 않는다
- 범주형 feature은 차원 간의 거리를 찾기 어려워 잘 작동하지 않는다

sparse data
: 차원, 전체 공간에 비해 데이터가 있는 공간이 매우 협소한 데이터

04. Decision Tree

Decision Tree (결정 트리) 란?

특정 규칙에 따라서 레이블을 분류하는 모델

데이터의 어떤 기준을 바탕으로 규칙을 만드느냐가 성능을 결정하는 중요한 요소

분류와 회귀, 다중 출력이 가능한 머신 러닝 알고리즘

Random Forest의 기본 구성 요소

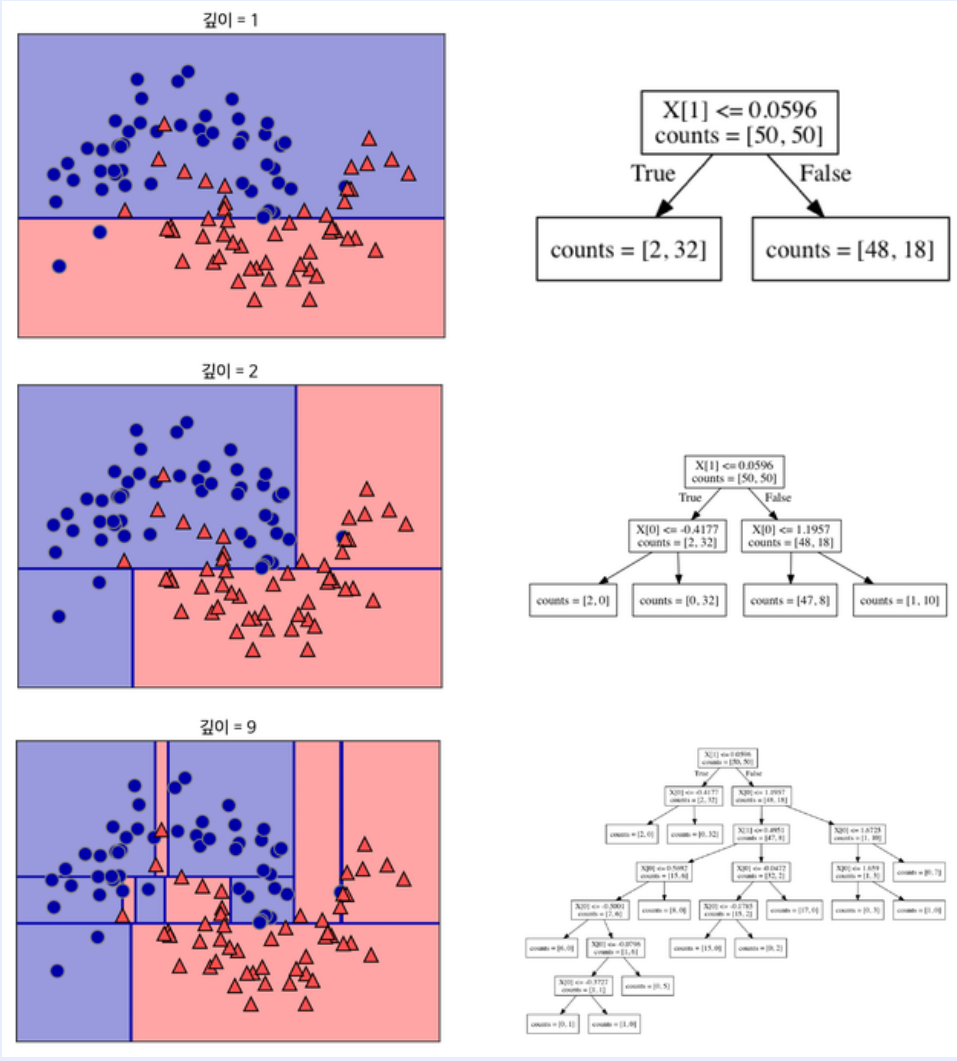
지나치게 많은 규칙으로 분류할 경우, overfitting 발생 가능성이 높다.

가지치기 (pruning)

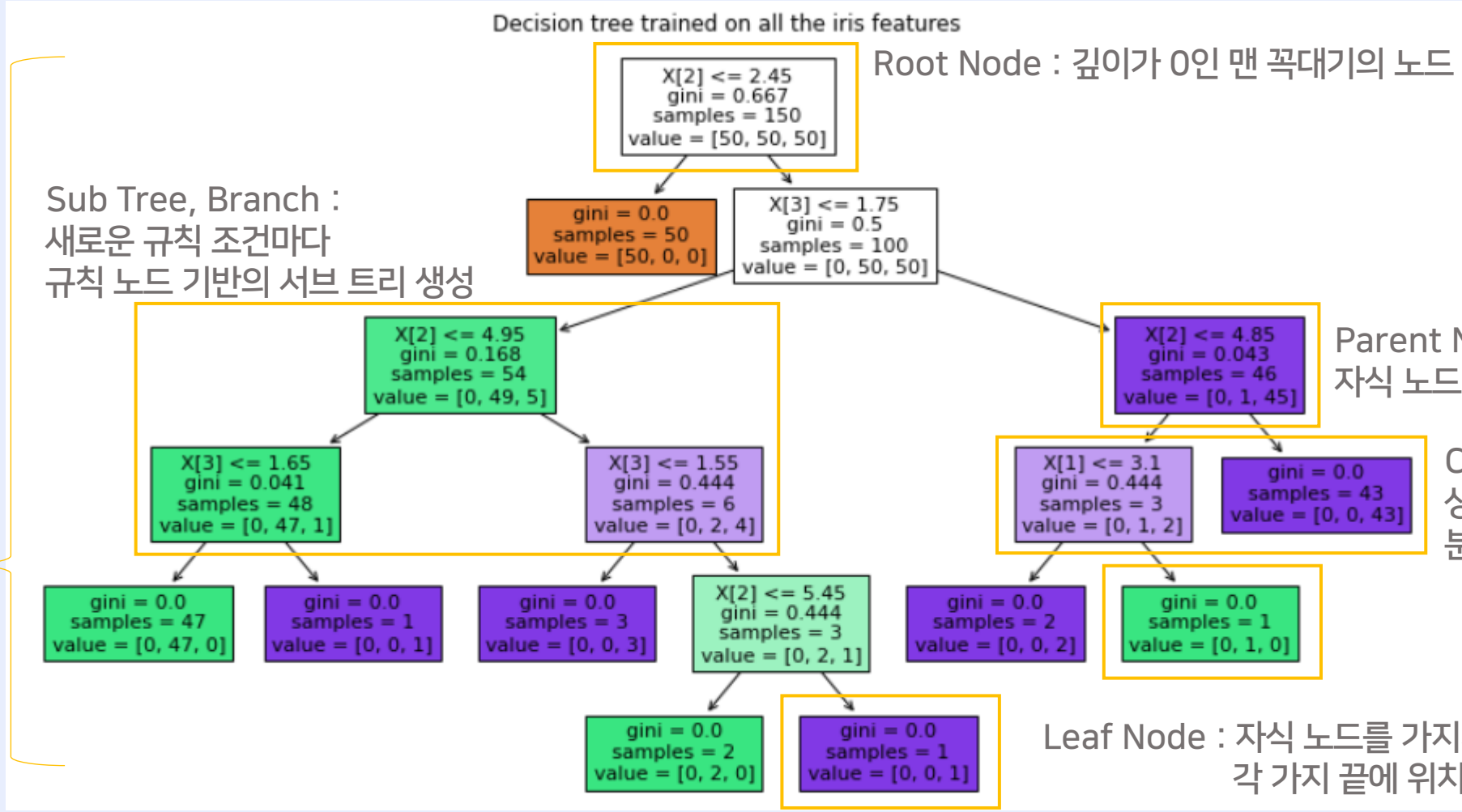
과적합을 막기 위한 전략

최대 깊이나 리프 노드의 최대 개수, 한 노드가 분할하기 위한 최

소 데이터 수를 제한하는 것



04. Decision Tree



Root Node : 깊이가 0인 맨 꼭대기의 노드

Sub Tree, Branch :
새로운 규칙 조건마다
규칙 노드 기반의 서브 트리 생성

Parent Node :
자식 노드의 상위 노드

Child Node :
상위 노드에서
분리된 하위 노드

깊이 :
가지를 이루고 있는
노드의 분리 층수

Leaf Node : 자식 노드를 가지지 않는 노드
각 가지 끝에 위치한 노드

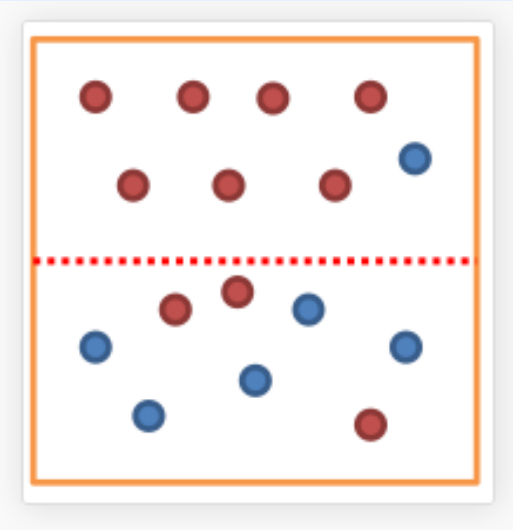
04. Decision Tree

불순도(Impurity)란?

해당 범주 안에 서로 다른 데이터가 얼마나 섞여 있는지를 의미

위쪽 범주는 불순도가 낮고(순도가 높고), 아래쪽 범주는 불순도가 높다(순도가 낮다).

이 불순도를 최소화 하는 방향으로 학습을 진행



정보의 불순도 측정 방법

- 엔트로피(Entropy)를 이용한 정보 이득(Information gain)
- 지니계수(Gini Index)

04. Decision Tree

엔트로피(Entropy)

불순도를 수치적으로 나타낸 척도

엔트로피가 높다 = 불순도가 높다 / 엔트로피가 낮다 = 불순도가 낮다

Entropy

$$Entropy = -\sum_{i}(p_i)\, log_2(p_i)$$

(p_i : 한 영역 안에 존재하는 데이터 가운데 범주 i에 속하는 데이터의 비율)

지니계수(Gini Index)

지니계수가 낮을 수록 데이터 균일도가 높다고 해석

0일 때 가장 균일, 1에 가까울 수록 균일하지 않음
지니 계수가 낮은 속성을 기준으로 분할

지니 계수

$$G.I(A) = \sum_{i=1}^{d}(R_i\,(1 - \sum_{k=1}^{m} p_{ik}^2))$$

정보 이득

데이터를 분리할 때,
특정 노드 이전과 이후에 엔트로피 차이를 측정하는 척도

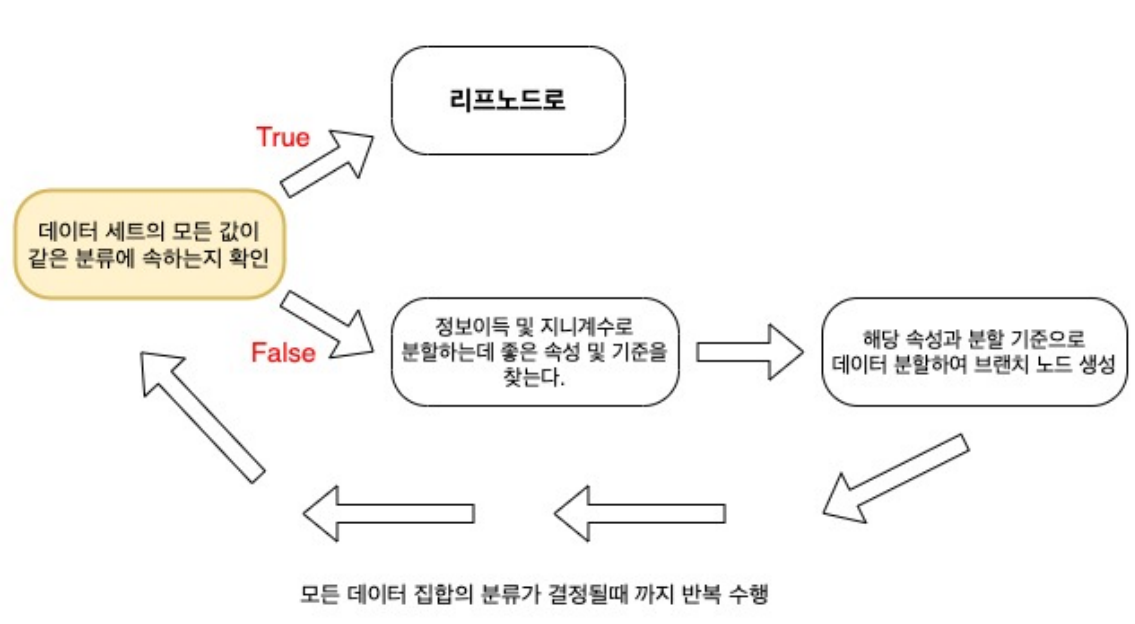
모든 feature의 정보 이득 계산 후, 정보 이득이 높은 feature를 기준으로 분리

정보 이득

$$Information_{Gain} = Entropy_{before} - Entropy_{after}$$

04. Decision Tree

결정 트리 과정



CART 훈련 알고리즘

Classification And Regression Tree

이진 트리만 만드는 알고리즘

scikit-learn에서 Decision Tree를 훈련시키기 위해 사용

CART 알고리즘이 subset을 나누는 과정

정의된 최대 깊이가 되면 중지 or
불순도를 줄이는 분할을 찾을 수 없을 때 중지

04. Decision Tree

Hyperparameter

DecisionTreeClassifier(*, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0)

Parameter	설명
max_depth	최대 깊이 지정 지정된 값까지 tree 깊이가 늘어나거나 노드가 가지는 데이터 수가 min_samples_split 보다 작아질 때까지 계속 분할
min_samples_split	분할되기 위해 노드가 가져야 하는 최고 샘플 수 작게 설정할 수록 분할되는 노드가 많아져 과적합 가능성 증가
min_samples_leaf	리프 노드가 가지고 있어야 할 최소 샘플 수 비대칭적 데이터의 경우, 특정 클래스의 데이터가 극도로 작을 수 있어 이 경우 작게 설정

04. Decision Tree

Hyperparameter

DecisionTreeClassifier(*, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0)

Parameter	설명
max_features	int, float or {"auto", "sqrt", "log2"} 각 노드에서 분할에 사용할 feature의 최대 수 int로 지정 시 대상 feature 개수, float로 지정 시 전체 feature 중 대상 feature의 퍼센트 “sqrt” : 전체 feature 중 제곱근 값 “auto” : sqrt와 동일 “log2” : 전체 feature를 log2()로 선정
max_leaf_nodes	리프 노드의 최대수
min_weight_fraction_leaf	min_samples_leaf 와 같지만 가중치가 부여된 전체 샘플 수에서의 비율

04. Decision Tree

장점

- 이해하고 해석하기 쉬우며, tree를 시각화 할 수 있다.
- 다중 출력 문제를 처리할 수 있다.
- 범주형, 연속형 수치 모두 예측 가능
- feature의 스케일링이나 정규화 등의
  데이터 전처리가 필요하지 않다.

단점

- 데이터 수가 적을 경우 불안정
- 복잡한 구조를 가지면, overfitting 가능성이 높아진다.

첨부자료 출처

01. SVM

SVM 이미지 출처
: https://img1.png
: https://img1.mtistory2
: https://bskyvision.com/gamma
: https://hleecaster.com/svm02-1536x1278.png

03. KNN

KNN 이미지 출처
: https://t1.daumcdn.net/cfile/tistory/994BB8505B72775C0B
: https://t1.daumcdn.net/cfile/tistory/99B2E23C5B7278870B
: https://bskyvision.com/gamma
: https://hleecaster.com/svm02-1536x1278.png

04. DT

DT 이미지 출처
: https://bkshin.tistory.com/Decision-Tree
: https://velog.velcdn.com/images.jpg

폰트

네이버 글꼴 모음 _ 나눔 스퀘어 사용
출처 : https://hangeul.naver.com/font

D&A

ML Session 4차시 분류 모델

Thank You.

2022 / 09 / 27

D&A 운영진 이예진