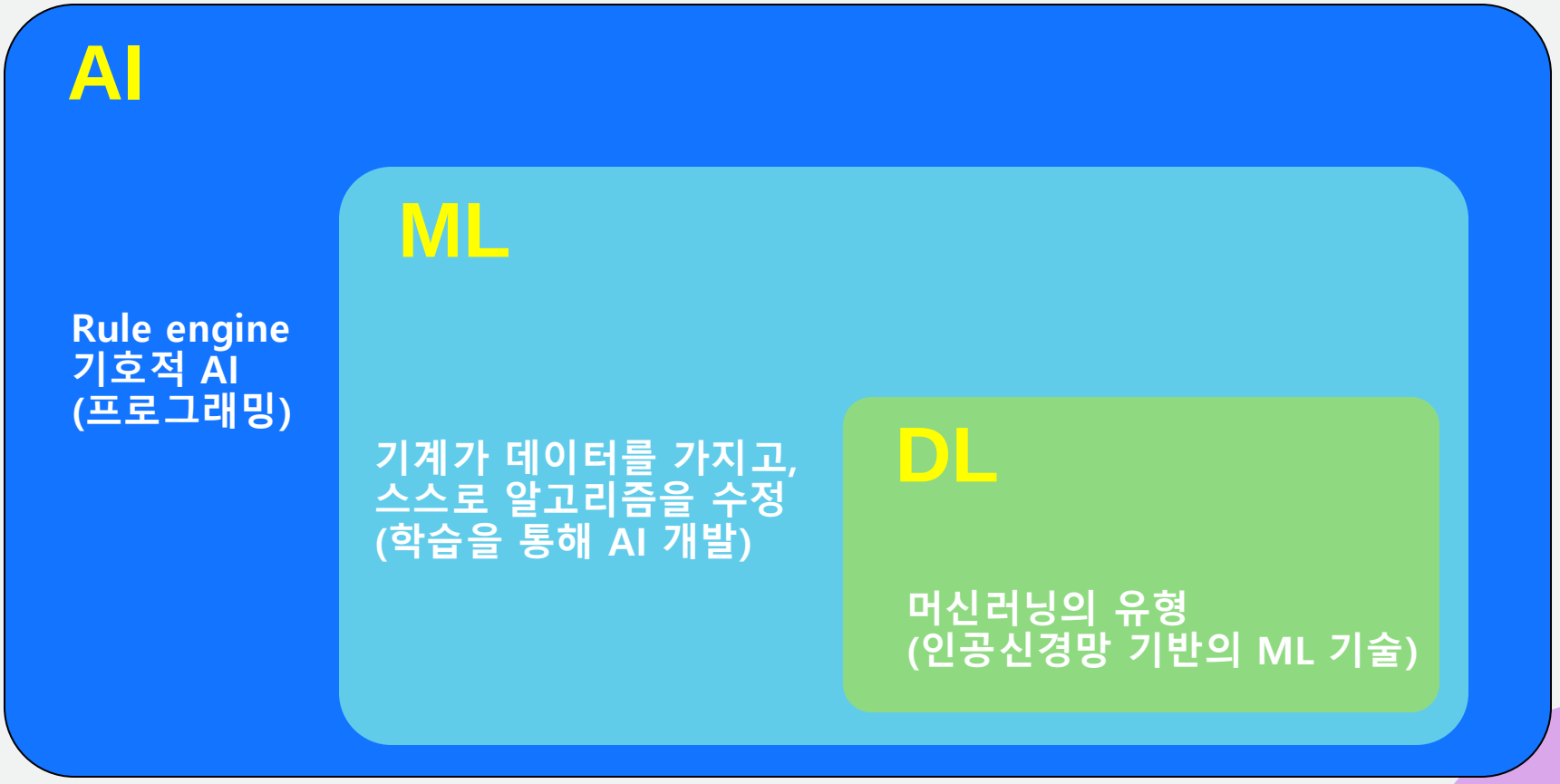
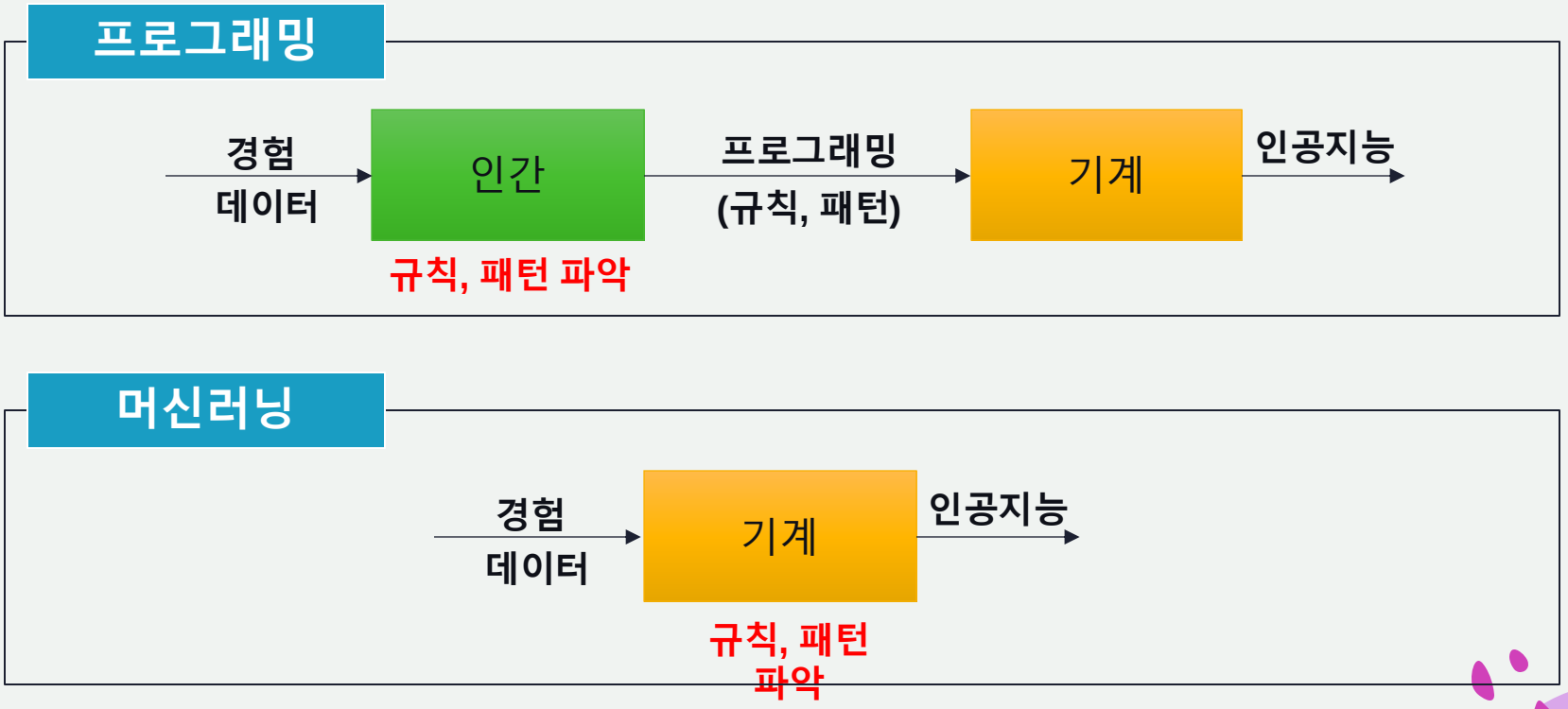
머신러닝 개요

강사 오승환

인공지능/머신러닝/딥러닝 구분



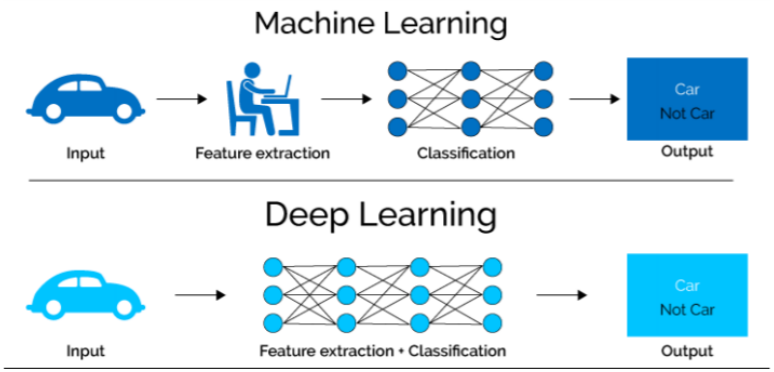
머신러닝 개념


프로그래밍
경험
데이터
인간
규칙, 패턴 파악
프로그래밍
(규칙, 패턴)
기계
인공지능
머신러닝
경험
데이터
기계
규칙, 패턴
파악
인공지능

머신러닝 특징

- 사람이 규칙을 정해주는 것이 아니라, 기계가 스스로 데이터 학습을 통해서 규칙을 찾는다.
- 기계가 데이터로부터 학습할 수 있게 하는 알고리즘과 기술을 개발하는 분야
- 컴퓨터 HW, SW 기술의 발달로 과거보다 훨씬 빠른 연산이 가능해짐에 따라, 모델의 예측 오차를 줄이는 방향으로 시행착오를 반복하여 빠르게 최적화

그림4 머신러닝 VS 딥러닝

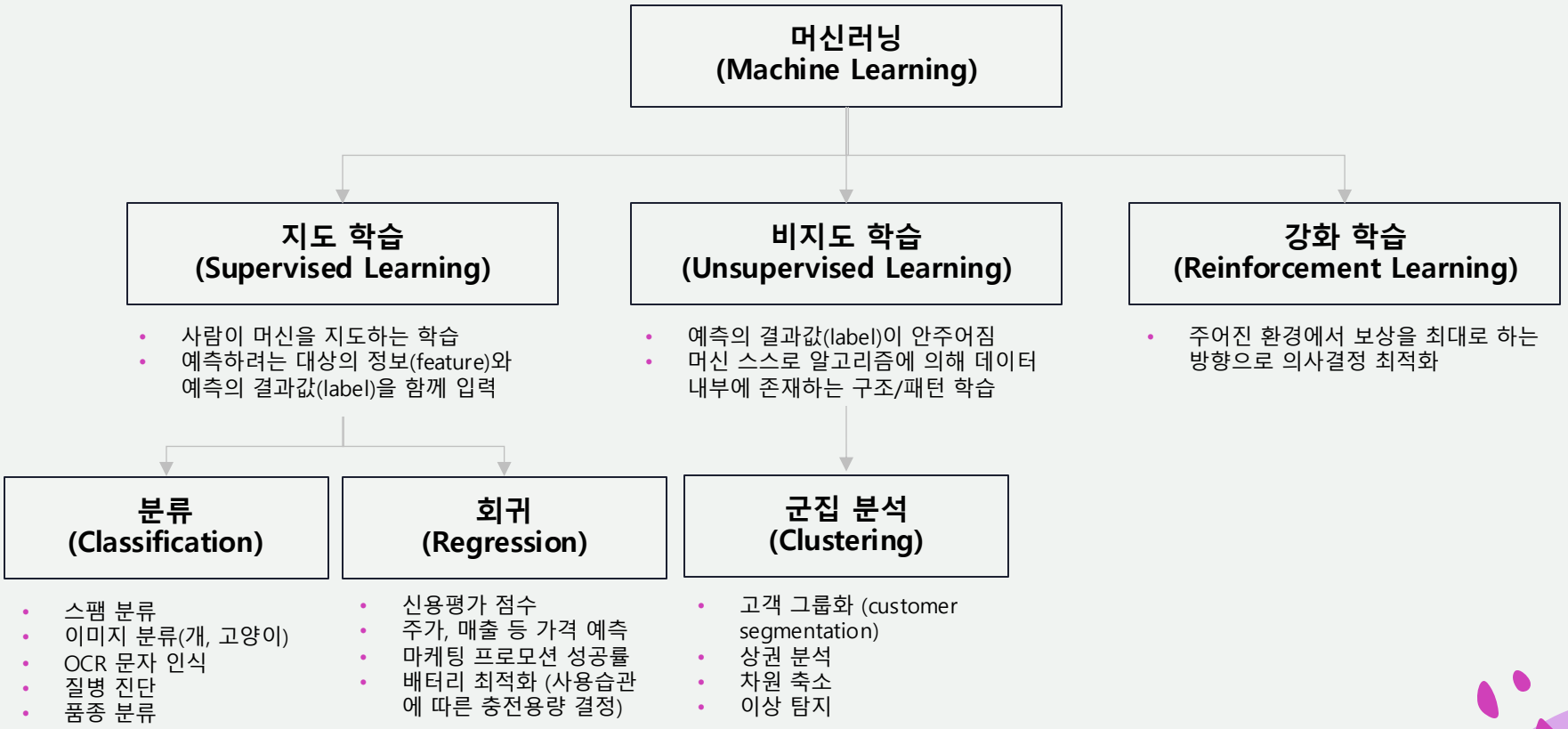


자료: Towards Data Science, 메리츠종금증권 리서치센터

머신러닝 분류

- 모델에 입력되는 정보의 양 (정답 label 포함 여부)
  - 지도 학습
  - 비지도 학습
  - 준지도 학습
- 실시간으로 학습하는지 여부
  - 온라인 학습
  - 배치 학습
- 기존 데이터와 비교하는 지, 또는 데이터에서 패턴을 찾는 것인지 여부
  - 사례 기반 학습
  - 모델 학습

머신러닝 유형


머신러닝
(Machine Learning)
지도 학습
(Supervised Learning)
비지도 학습
(Unsupervised Learning)
강화 학습
(Reinforcement Learning)
사람이 머신을 지도하는 학습
예측하려는 대상의 정보(feature)와 예측의 결과값(label)을 함께 입력
예측의 결과값(label)이 안주어짐
머신 스스로 알고리즘에 의해 데이터 내부에 존재하는 구조/패턴 학습
주어진 환경에서 보상을 최대로 하는 방향으로 의사결정 최적화
분류
(Classification)
회귀
(Regression)
군집 분석
(Clustering)
스팸 분류
이미지 분류(개, 고양이)
OCR 문자 인식
질병 진단
품종 분류
신용평가 점수
주가, 매출 등 가격 예측
마케팅 프로모션 성공률
배터리 최적화 (사용습관에 따른 충전용량 결정)
고객 그룹화 (customer segmentation)
상권 분석
차원 축소
이상 탐지

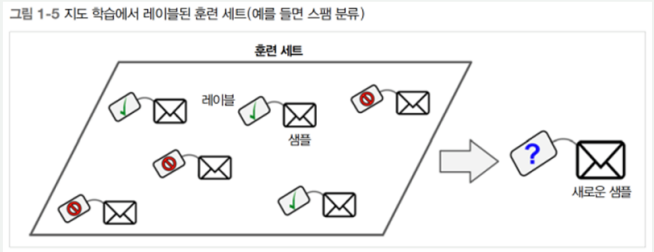
지도학습 - 분류

feature

label

순번	메일 제목	발신인	발신 일시	스팸 여부
1	(광고) 주식 VIP 정보서비스 에 당첨 되셨습니다.~	차**	2021-07-09 14:57	Y
2	[라이엇게임즈] 게임 이용 내역 알림	라이엇*	2021-07-09 15:52	N
3	(광고) 정부지원 생활안정자금! 지금 확인하세요~	이벤트메일	2021-07-10 7:43	Y
4	[클래스톡] 온라인 클래스 개설을 제안드립니다.🙏	김**	2021-07-11 13:52	N
5	어른들만 입장하세요	18894	2021-07-12 13:24	Y
...	...	...	...	...
1000	해 외 정 품 비 아 그 라	450804	2021-07-13 19:39	Y

스팸 분류



출처 https://tensorflow.blog

스팸 분류 모델
(머신러닝 알고리즘)

메일 제목	발신인	발신 일시
(광고) 개인사업자, 프리랜서분들~ 무료로 절세솔루션 받으실 분?!	프리미엄***	2021-07-22 14:27

예측할 데이터
(새로운 샘플)

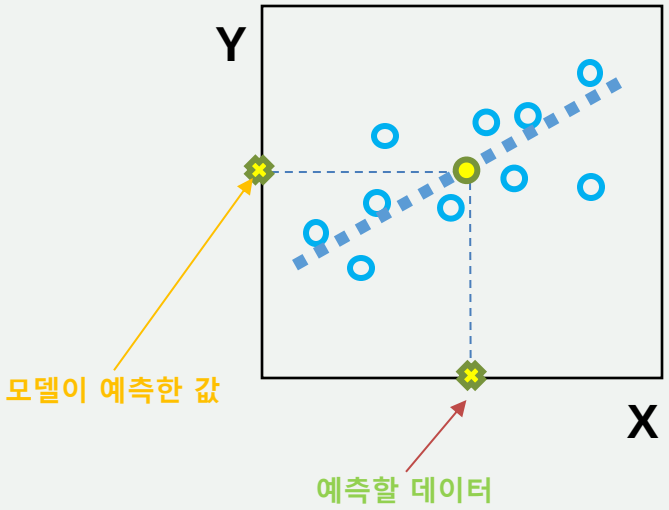
스팸 여부
Y

지도학습 - 회귀

feature

label

순번	공급면적(평)	방 개수	브랜드	500M 이내 지하철 수	...	아파트 가격
1	24	2	래미안	2	...	6.5억
2	32	3	힐스테이트	1	...	12억
3	32	3	래미안	0	...	4.8억
4	48	4	아이파크	1	...	7억
5	24	3	대상	3	...	15억
...	...	...	...	...	...	...
1000	32	3	아이파크	2		10억

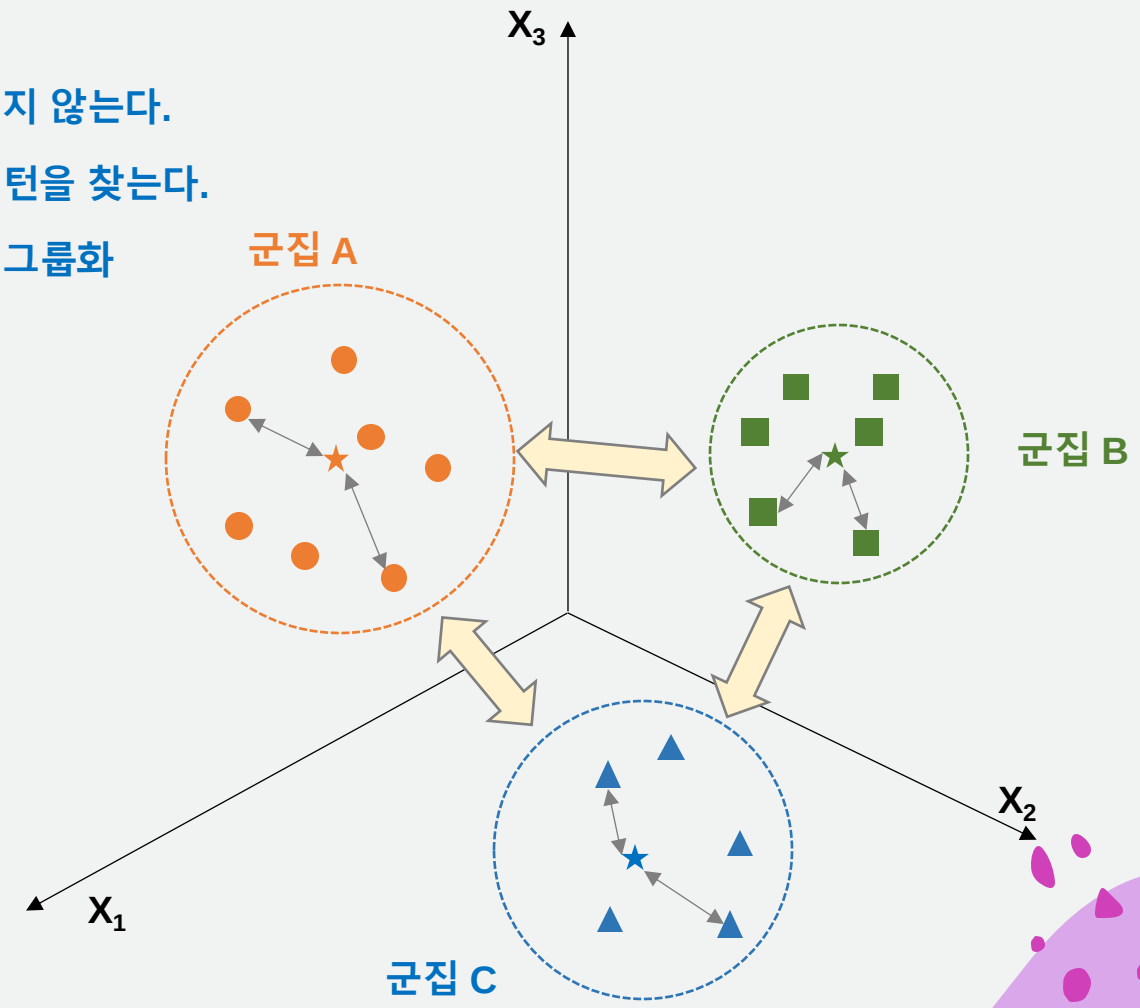


회귀 vs. 분류

- 회귀: 예측하려는 label 값이 연속형 실수
- 분류: 예측하려는 label 값이 범주형 자료

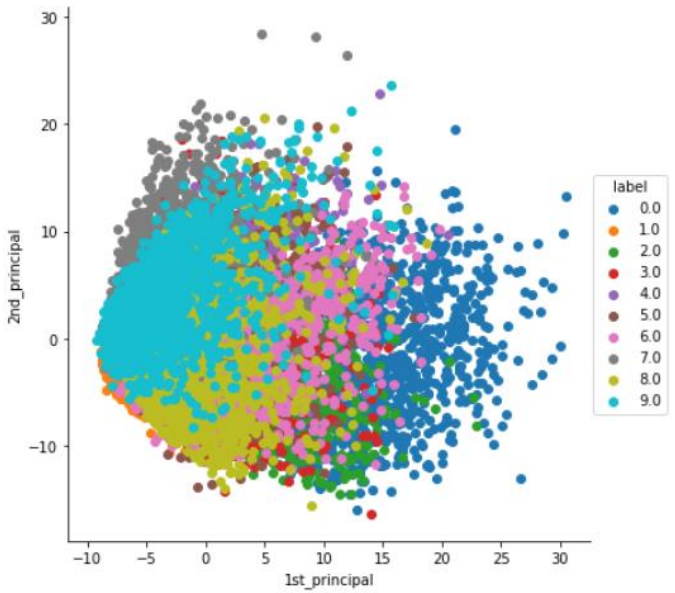
비지도학습 - 군집

1. 정답에 해당하는 label이 주어지지 않는다.
2. 데이터에 숨어 있는 규칙이나 패턴을 찾는다.
3. 비슷한 속성을 갖는 데이터끼리 그룹화

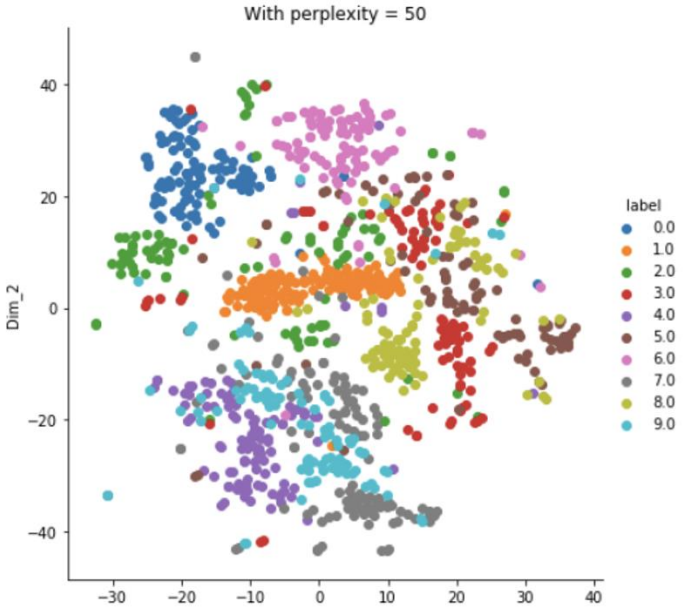


비지도학습 - 차원축소

1. 고차원을 저차원으로 축소하여 분석 (PCA, SVD)
2. 시각화 분석 (t-SNE)



PCA

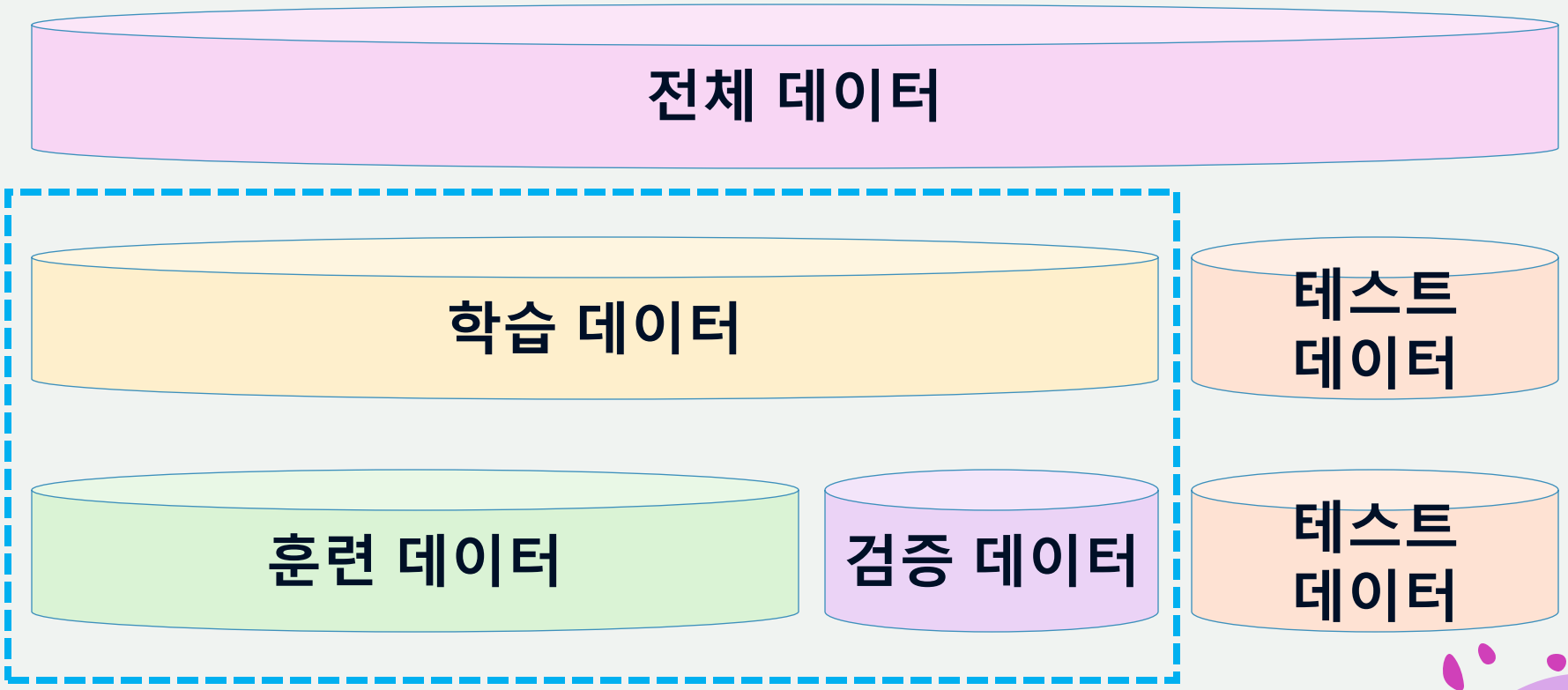


t-SNE

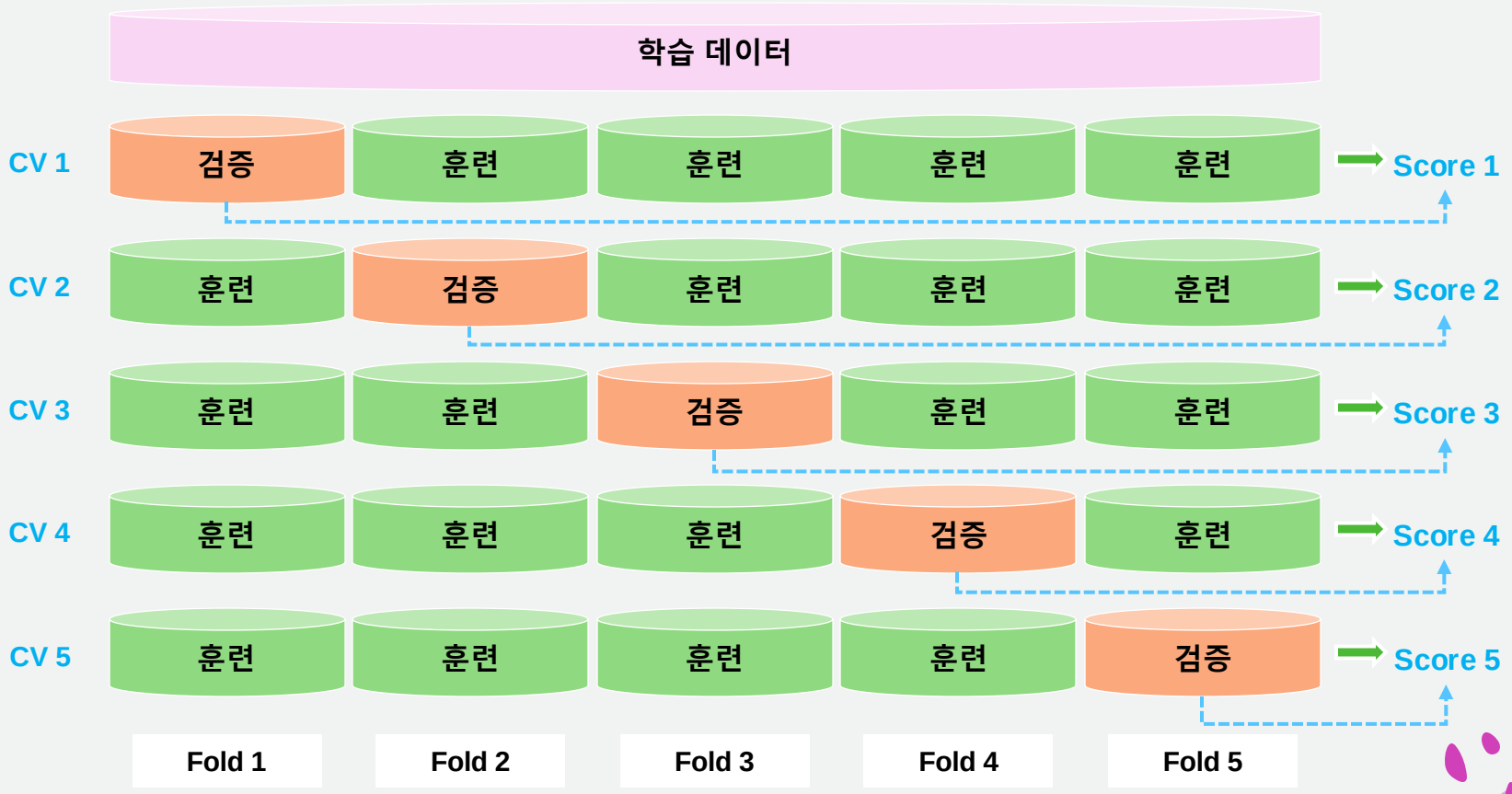
머신러닝 모델링 절차


데이터 수집 및 전처리
데이터 분리 (훈련/검증)
알고리즘 선택
모델 학습 (훈련 데이터)
예측 (검증 데이터)
모델 성능 평가
모델 적용

Hold-Out 교차검증



K-Fold 교차검증


학습 데이터
CV 1
검증
훈련
훈련
훈련
훈련
Score 1
CV 2
훈련
검증
훈련
훈련
훈련
Score 2
CV 3
훈련
훈련
검증
훈련
훈련
Score 3
CV 4
훈련
훈련
훈련
검증
훈련
Score 4
CV 5
훈련
훈련
훈련
훈련
검증
Score 5
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5

회귀 분석

주택가격 예측

Boston
Housing Price

이미지 출처: https://www.hellodd.com/news/articleView.html?idxno=69865

주택가격 예측

데이터 로딩

데이터프레임 변환

```
# 보스턴 주택 데이터
from sklearn import datasets
housing = datasets.load_boston()
data = pd.DataFrame(housing.data, columns=housing.feature_names)
data['MEDV'] = housing.target

data.head()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	36.2

주택가격 예측

Number of Instances:	506
Number of Attributes:	13 numeric/categorical predictive. Median Value (attribute 14) is usually the target.
Attribute Information (in order):	• CRIM per capita crime rate by town • ZN proportion of residential land zoned for lots over 25,000 sq.ft. • INDUS proportion of non-retail business acres per town • CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise) • NOX nitric oxides concentration (parts per 10 million) • RM average number of rooms per dwelling • AGE proportion of owner-occupied units built prior to 1940 • DIS weighted distances to five Boston employment centres • RAD index of accessibility to radial highways • TAX full-value property-tax rate per $10,000 • PTRATIO pupil-teacher ratio by town • B 1000(Bk - 0.63)^2 where Bk is the proportion of black people by town • LSTAT % lower status of the population • MEDV Median value of owner-occupied homes in $1000's
Missing Attribute Values:	None
Creator:	Harrison, D. and Rubinfeld, D.L.

출처: https://scikit-learn.org/stable/datasets/toy_dataset.html?highlight=boston%20housing

주택가격 예측

- ## 상관계수 분석 - 변수 선택

```
data.corr().loc[:'LSTAT', 'MEDV'].abs().sort_values(ascending=False)
```

```
LSTAT      0.737663
RM         0.695360
PTRATIO    0.507787
INDUS      0.483725
TAX        0.468536
NOX        0.427321
CRIM       0.388305
RAD        0.381626
AGE        0.376955
ZN         0.360445
B          0.333461
DIS        0.249929
CHAS       0.175260
Name: MEDV, dtype: float64
```

```
X = data.loc[:, ['LSTAT', 'RM', 'PTRATIO']]
X.head(2)
```

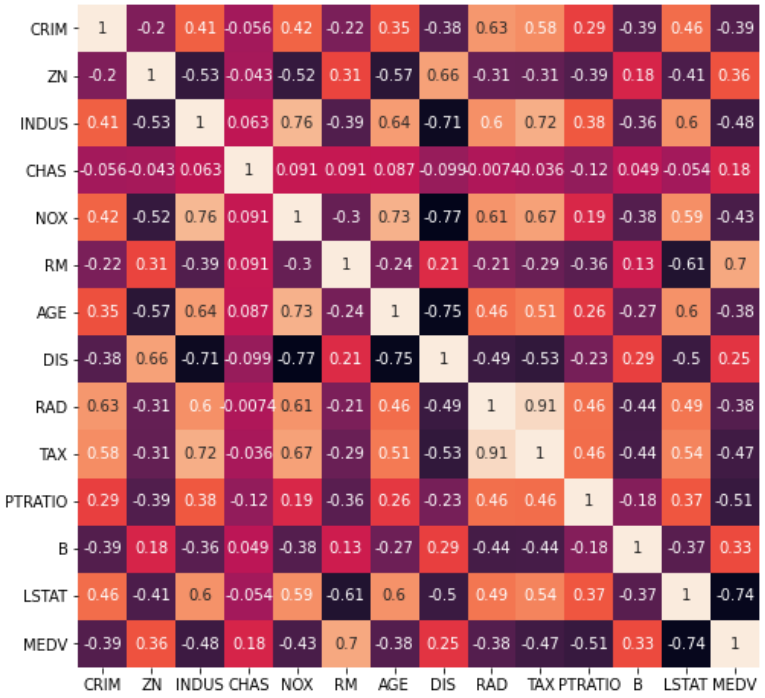
	LSTAT	RM	PTRATIO
0	4.98	6.575	15.3
1	9.14	6.421	17.8

입력 변수

```
y = data['MEDV']
y.shape
```

```
(506,)
```

목표 변수



주택가격 예측

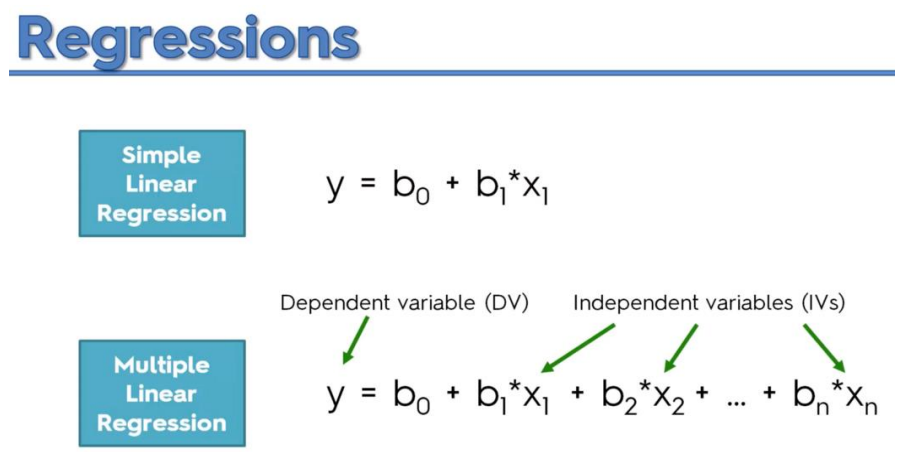
- **Train-Test Split (훈련-검증 데이터셋 분할)**
  - 모델은 과거에 수집된 데이터를 사용하여 학습 (훈련 데이터)
  - 미래 시점의 데이터를 미리 사용 불가 (검증 데이터)
  - 훈련 데이터로 학습하고, 검증 데이터를 이용하여 모델 성능을 평가

8:2 비율

```
from sklearn.model_selection import train_test_split
X_tr, X_val, y_tr, y_val = train_test_split(X, y, test_size=0.2, random_state=2021)

print(X_tr.shape, y_tr.shape)
print(X_val.shape, y_val.shape)
```

```
(404, 3) (404,)
(102, 3) (102,)
```

선형회귀(Linear Regression)



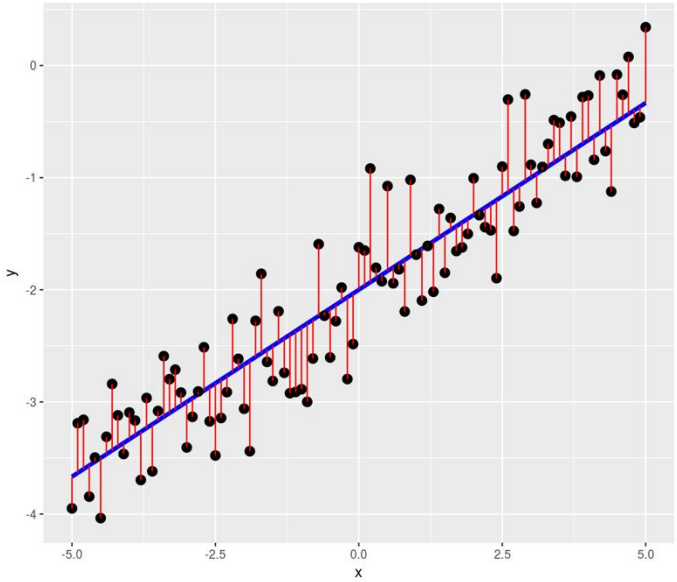
출처: https://medium.com/@manjabogicevic/multiple-linear-regression-using-python-b99754591ac0

모델 생성 & 훈련

```
: from sklearn.linear_model import LinearRegression
  lr_model = LinearRegression()
  lr_model.fit(X_tr, y_tr)

: LinearRegression()

: # 계수
  lr_model.coef_

: array([-0.59262681,  4.87677377, -0.81057334])

: # 상수(절편)
  lr_model.intercept_

: 14.394472589800367
```

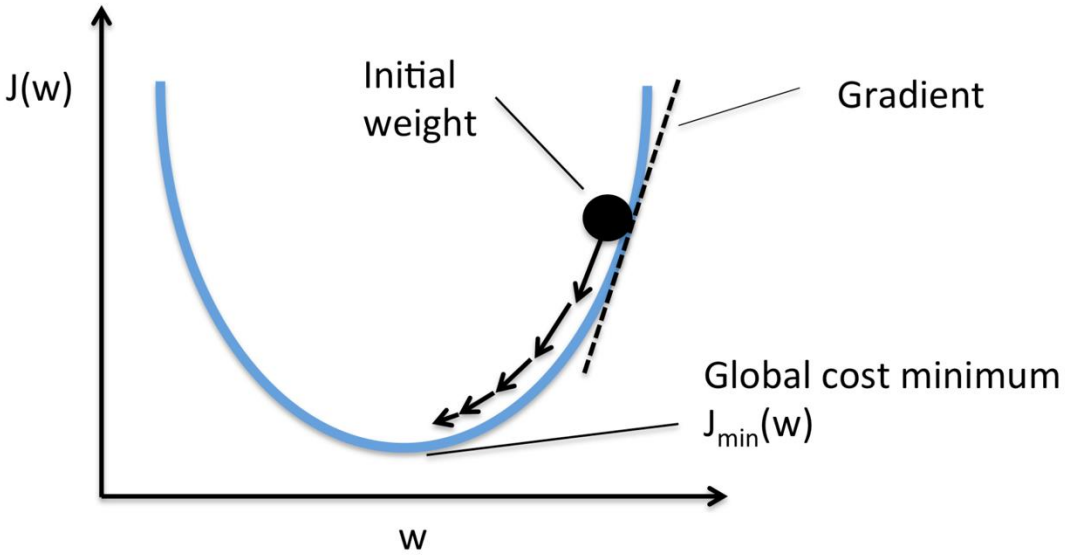
모델 학습

최소제곱법(OLS)



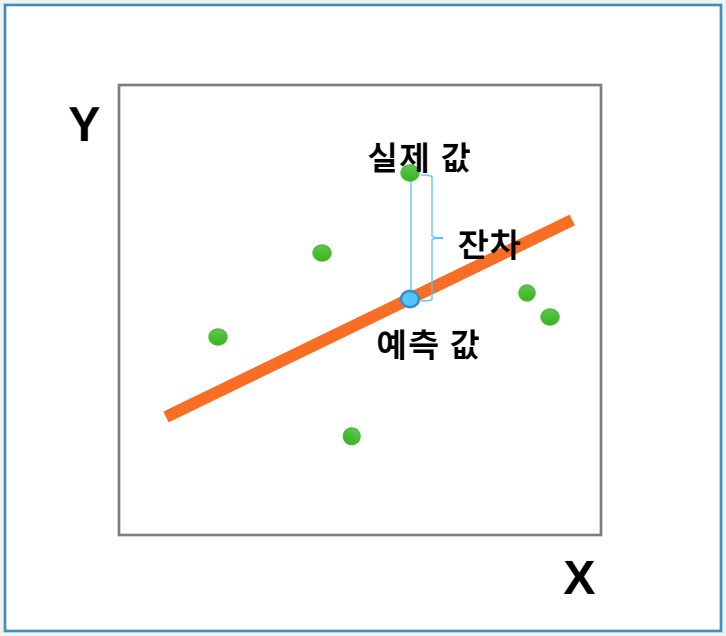
출처: https://vulstats.ucsd.edu/notes/bivariate-ols.html

경사하강법(Gradient Descent)



출처: http://rasbt.github.io/mlxtend/user_guide/general_concepts/gradient-optimization/

모델 성능 평가




```python
# 성능 평가 (R^2 결정계수)
print(f"훈련 셋: {lr_model.score(X_tr, y_tr)}")
print(f"검증 셋: {lr_model.score(X_val, y_val)}")
```

```
훈련 셋: 0.6849366532289445
검증 셋: 0.6336516343112433
```

```python
# 성능 평가 (MSE)
from sklearn.metrics import mean_squared_error
print(f"훈련 셋: {mean_squared_error(y_tr, lr_model.predict(X_tr))}")
print(f"검증 셋: {mean_squared_error(y_val, lr_model.predict(X_val))}")
```

```
훈련 셋: 28.231273347792094
검증 셋: 23.400980040876647
```

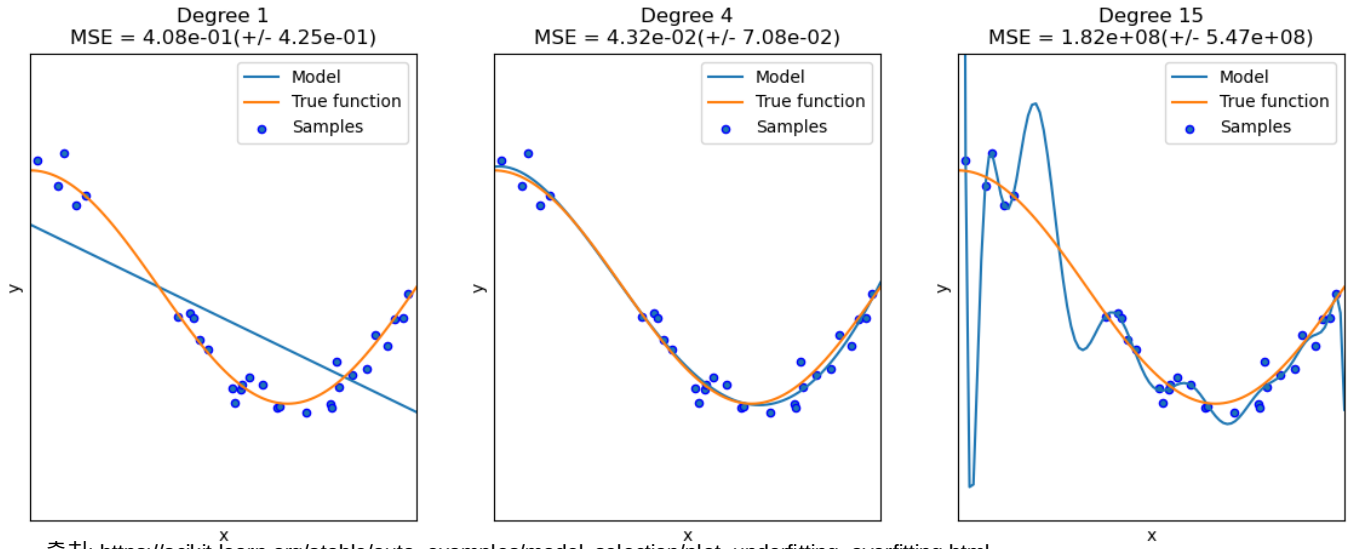
```python
# 성능 평가 (MAE)
from sklearn.metrics import mean_absolute_error
print(f"훈련 셋: {mean_absolute_error(y_tr, lr_model.predict(X_tr))}")
print(f"검증 셋: {mean_absolute_error(y_val, lr_model.predict(X_val))}")
```

```
훈련 셋: 3.6824616133245254
검증 셋: 3.5630807724232714
```

회귀 모델 평가지표

MSE	$\frac{1}{N}\sum_{i}^{N}(pred_i - target_i)^2$	• 오차에 제곱을 하므로, 이상치에 민감
MAE	$\frac{1}{N}\sum_{i}^{N}\lvert(pred_i - target_i)\rvert$	• 실제 단위와 같은 차원에서 직관적 비교 가능 • 모델 예측값의 과대/과소 판단이 불가능
RMSE	$\sqrt{\frac{1}{N}\sum_{i}^{N}(pred_i - target_i)^2}$	• MSE의 제곱근으로 실제 단위로 환산 • MAE보다 이상치에 민감
R^2	$\frac{SSR}{SST}$	• 평균값 대비 예측 모델의 성능 향상 수준을 평가 • 1에 가까우면 좋은 모델 (음수값 나올 수 있음)

과대 적합 vs. 과소 적합



출처: https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html

```
X_tr, X_val, y_tr, y_val = train_test_split(X, y, test_size=0.2, random_state=2021)

lr_model = LinearRegression()
lr_model.fit(X_tr, y_tr)

print(f"훈련 셋: {lr_model.score(X_tr, y_tr)}")
print(f"검증 셋: {lr_model.score(X_val, y_val)}")
```

```
훈련 셋: 0.7561240805539942
검증 셋: 0.6352336167833779
```

과대 적합

규제(Regularization)

- 모델의 특정 피처에 적용되는 가중치(w)가 커지는 것을 규제 (penalty 부과)
- 특정 피처의 영향력을 제한하여, 모델의 복잡도를 낮추는 효과

L1 Regularization

$$\text{Cost} = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2 + \lambda \sum_{j=0}^{M} |W_j|$$

L2 Regularization

$$\text{Cost} = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2 + \lambda \sum_{j=0}^{M} W_j^2$$

Loss function

Regularization Term

http://laid.delanover.com/difference-between-l1-and-l2-regularization-implementation-and-visualization-in-tensorflow/

Ridge

- 가중치의 제곱합에 penalty 부과
- 선형회귀 모델에 L2 규제

Lasso

- 가중치 절대값의 합에 penalty 부과
- 선형회귀 모델에 L1 규제

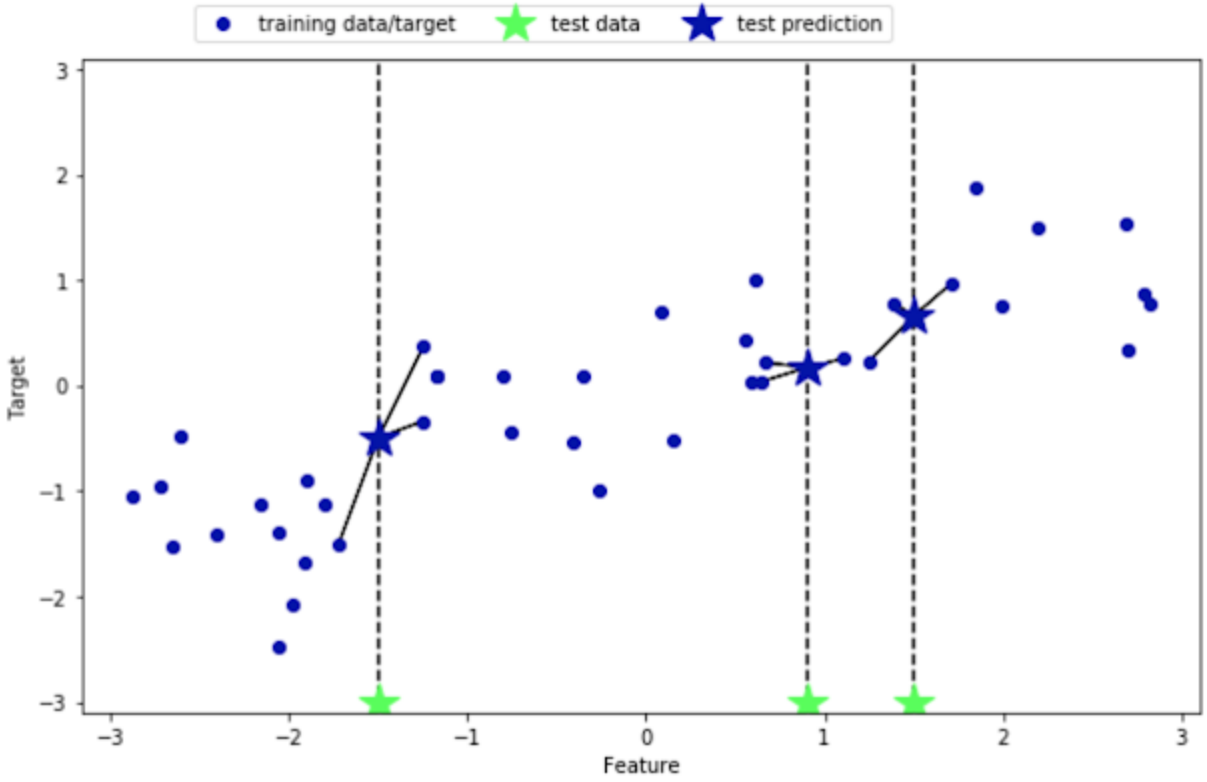
ElasticNet

- 선형회귀 모델에 L1/L2 규제를 동시에 적용

비선형 회귀 모델

- **KNN (K-Nearest Neighbors)**
  - 예측값 새로운 데이터 포인트에 가까운 이웃의 출력값 평균

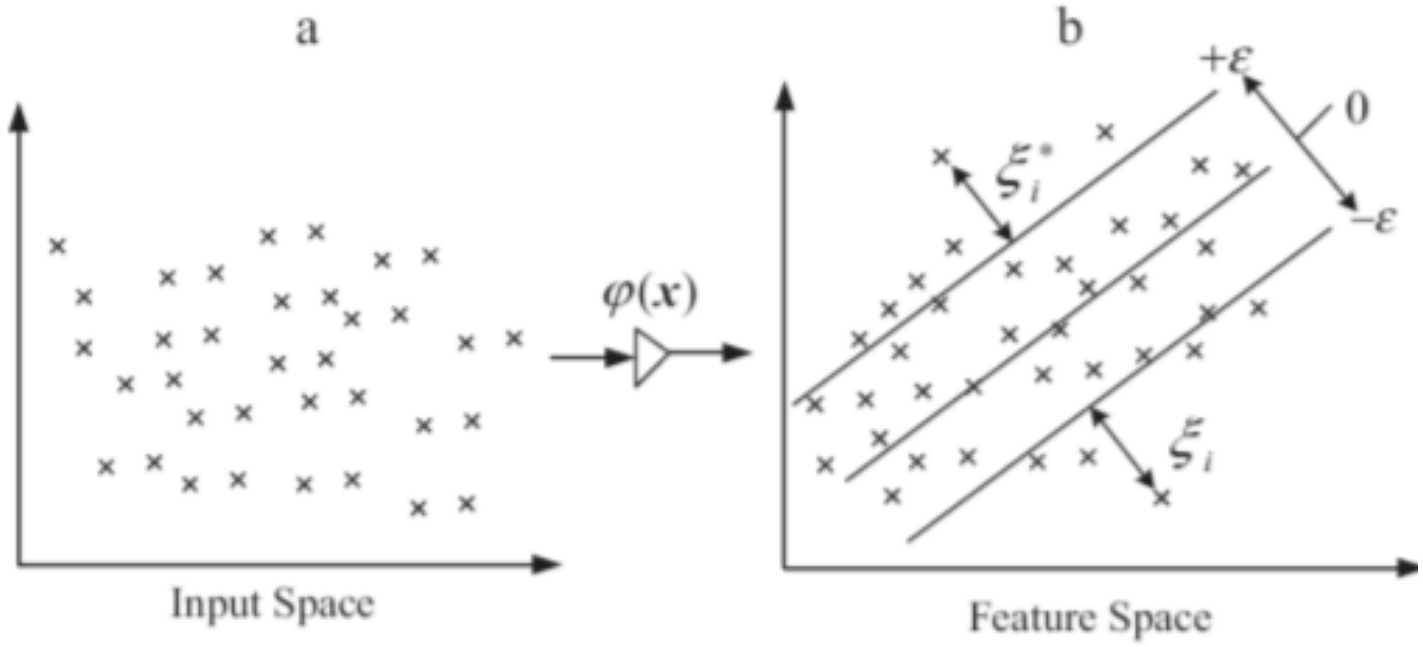
k=3



http://the-ml-blogs.blogspot.com/2018/12/5-knn-regression.html

비선형 회귀 모델

- **서포트 벡터 머신(SVM)**
  - 선형변환을 통해 선형 관계를 갖는 Feature Space를 찾음



출처: https://www.sciencedirect.com/science/article/abs/pii/S0307904X1730344X

Feature Scaling

- **각 피처의 데이터 값 범위에 따른 영향을 제거**
  - MinMaxScaler: 각 피처의 데이터를 0~1 범위로 정규화
  - StandardScaler: 각 피처의 데이터를 평균 0, 표준편차 1 갖도록 변환

```
X.head()
```

	LSTAT	RM	PTRATIO
0	4.98	6.575	15.3
1	9.14	6.421	17.8
2	4.03	7.185	17.8
3	2.94	6.998	18.7
4	5.33	7.147	18.7

0~1 범위로 정규화

```
X_scaled.head()
```

	LSTAT	RM	PTRATIO
0	0.089680	0.577505	0.287234
1	0.204470	0.547998	0.553191
2	0.063466	0.694386	0.553191
3	0.033389	0.658555	0.648936
4	0.099338	0.687105	0.648936

분류 분석

붓꽃 품종 분류

setosa

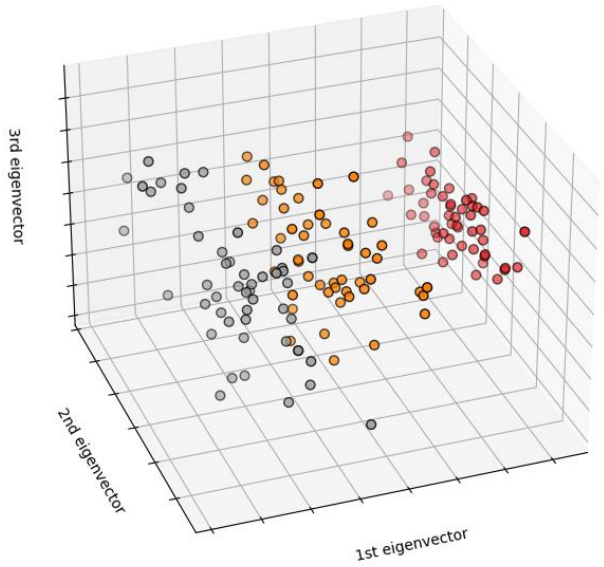
versicolor

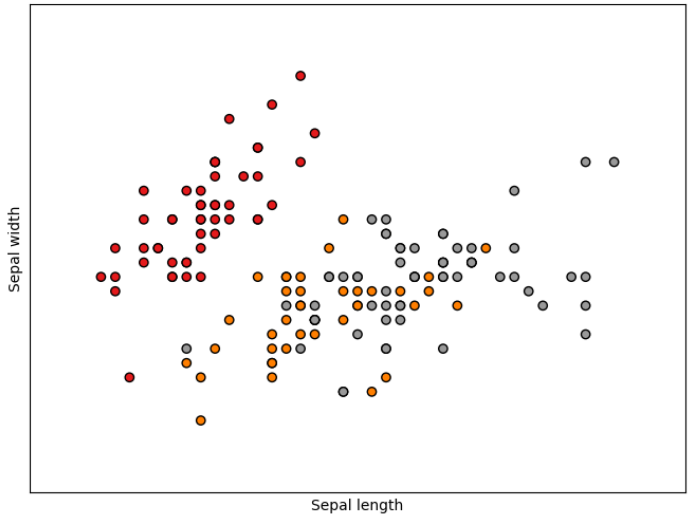
virginica

```
from sklearn import datasets
iris = datasets.load_iris()
df = pd.DataFrame(iris['data'], columns=iris['feature_names'])
df["Target"] = iris['target']
df.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	Target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

붓꽃 품종 분류

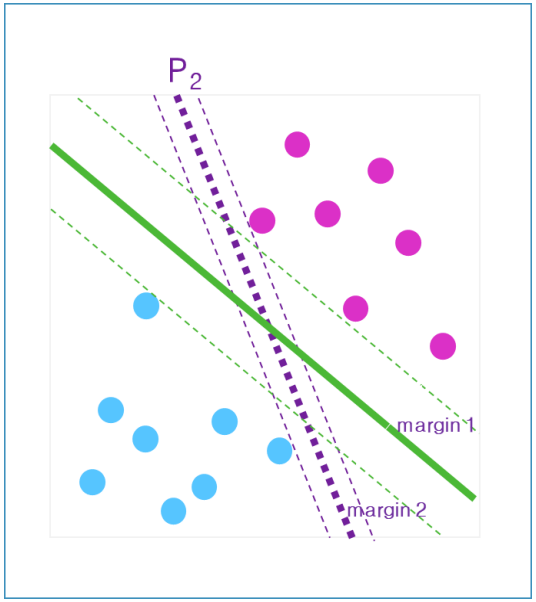




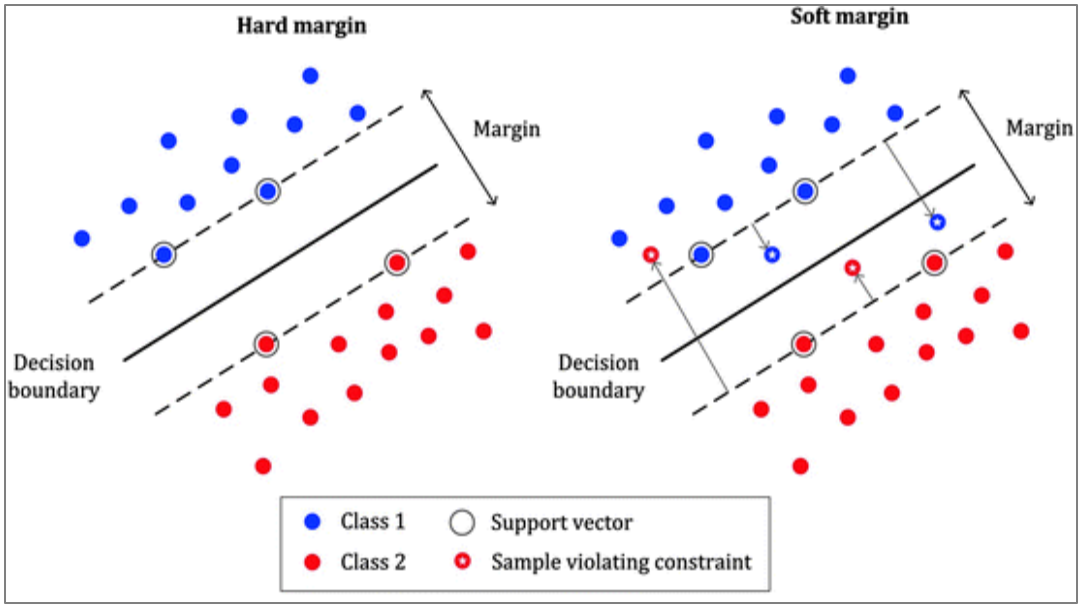
출처: https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html

SVM (서포트 벡터 머신)

결정 경계

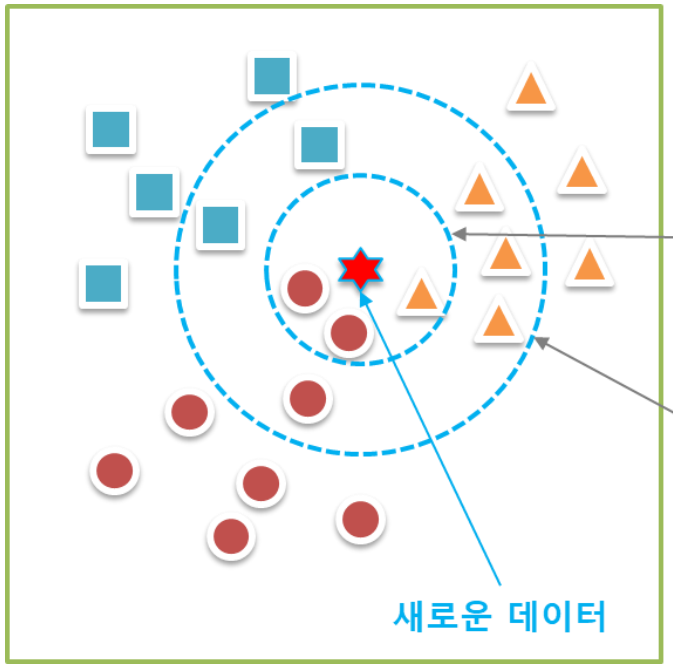


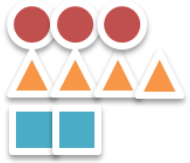
Hard margin vs. Soft margin



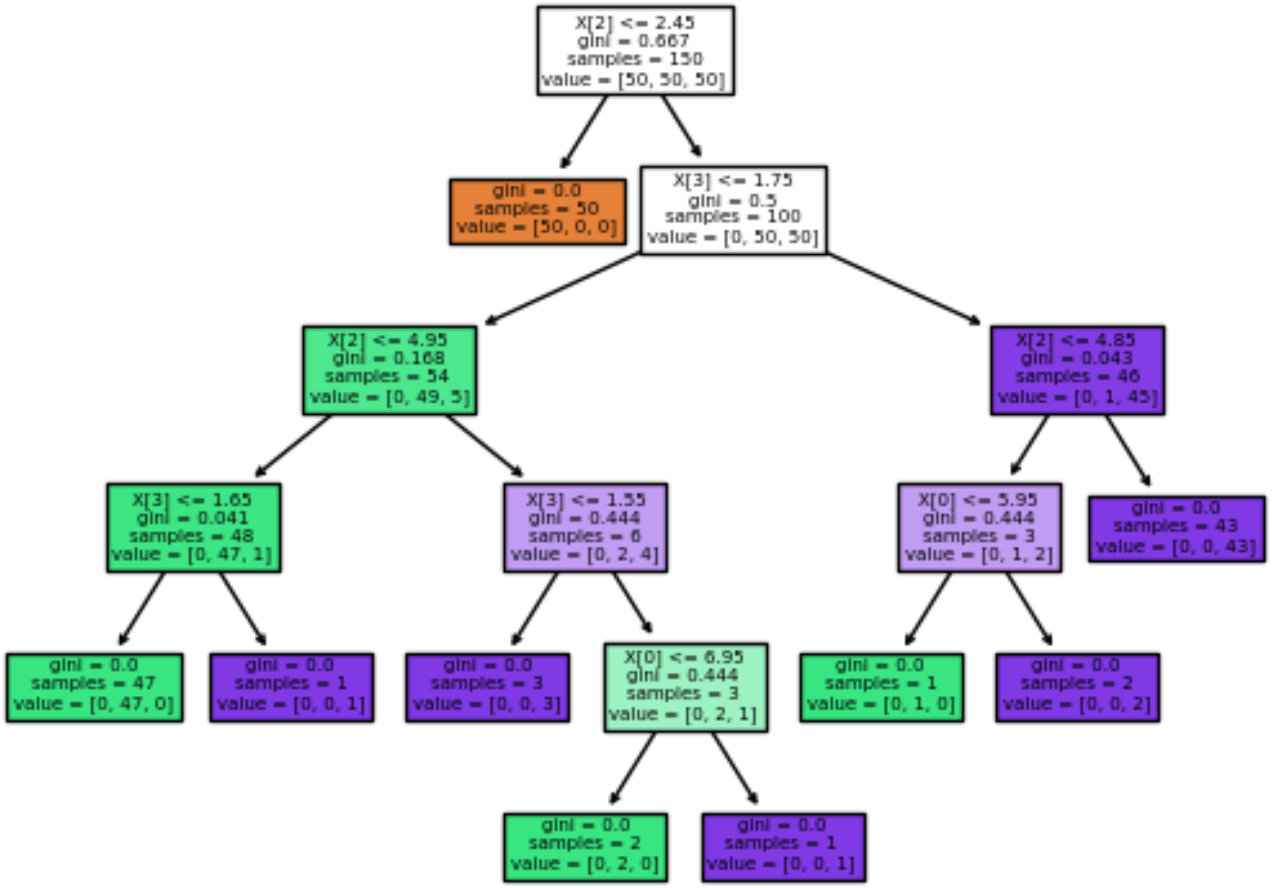
출처: https://ankitnitjsr13.medium.com/math-behind-svm-support-vector-machine-864e58977fdb

KNN (K-Nearest Neighbors)



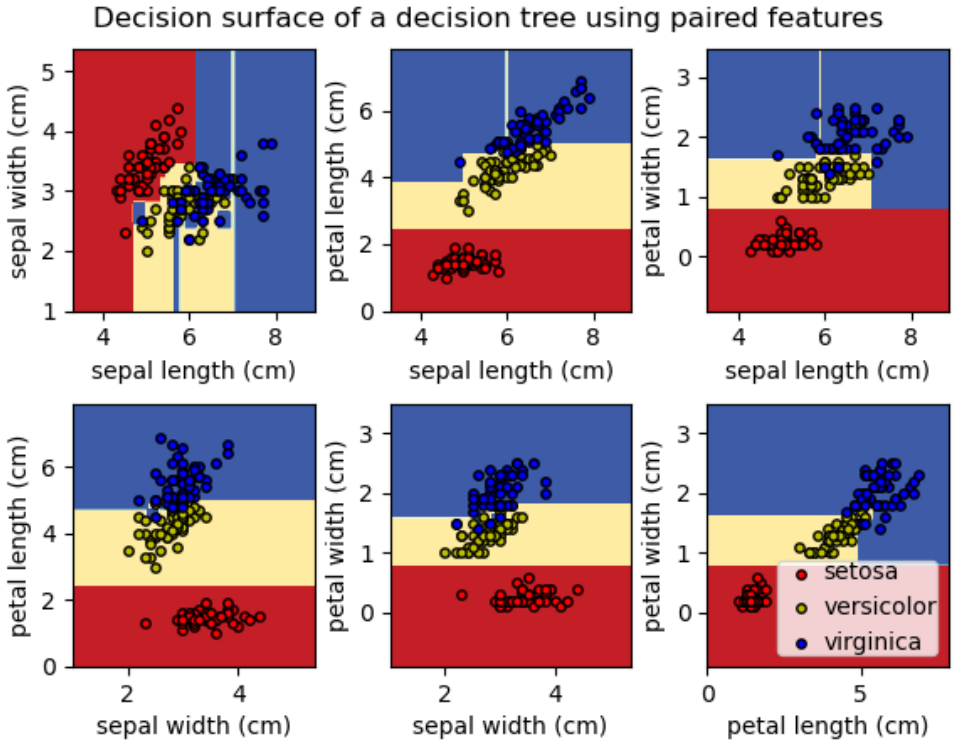
K	K-근접 이웃	최종 예측
K=3		
K=9		

의사결정나무 (Decision Tree)

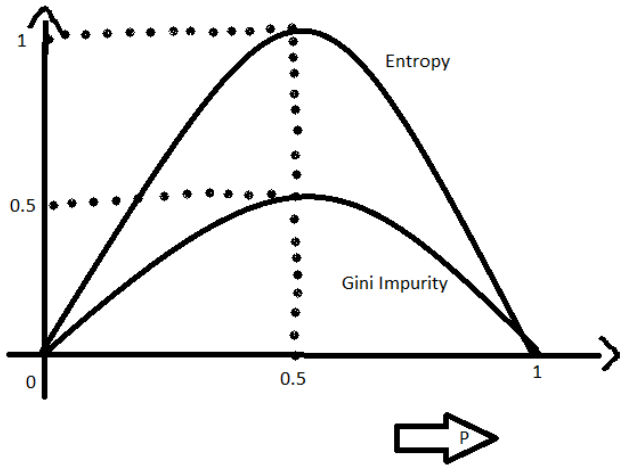


출처: https://scikit-learn.org/stable/auto_examples/tree/plot_iris_dtc.html

의사결정나무 (Decision Tree)



출처: https://scikit-learn.org/stable/auto_examples/tree/plot_iris_dtc.html



$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

$$Gini(E) = 1 - \sum_{j=1}^{c} p_j^2$$

출처: https://www.geeksforgeeks.org/gini-impurity-and-entropy-in-decision-tree-ml/

분류 모델 평가지표

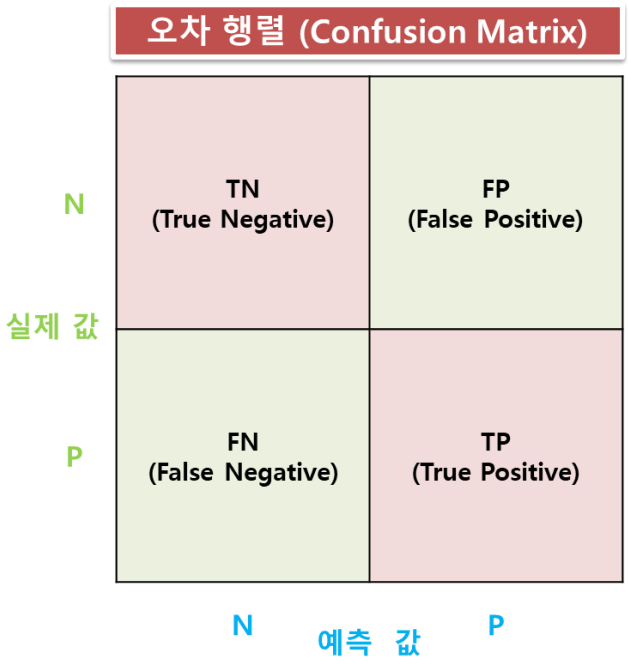
정확도 (Accuracy) = $\frac{\text{정확하게 분류한 데이터 개수}}{\text{전체 데이터 개수}}$

정밀도 (Precision) = $\frac{TP}{FP+TP}$

재현율 (Recall) = $\frac{TP}{FN+TP}$

F1 = $\frac{2}{\frac{1}{precision}+\frac{1}{recall}} = 2 * \frac{precision*recall}{precison+recall}$

ROC-AUC



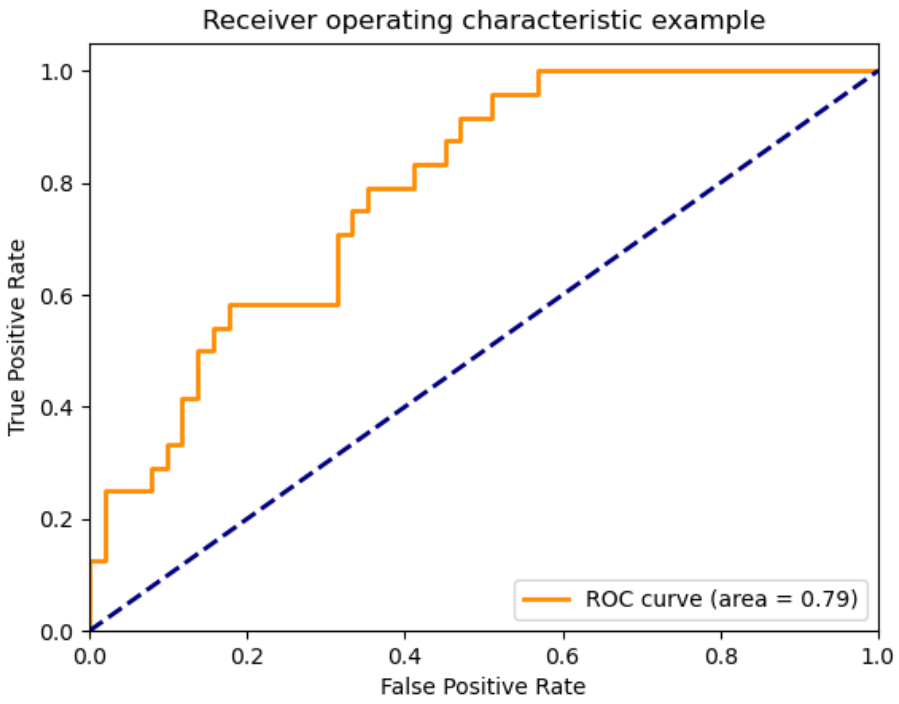
분류 모델 평가지표

ROC-AUC : FPR을 0부터 1까지 변경하면서 TPR(민감도)의 변화를 추적

FPR(False Positive Rate)

$= \frac{FP}{FP+TN}$

= 1 – TNR(특이성)

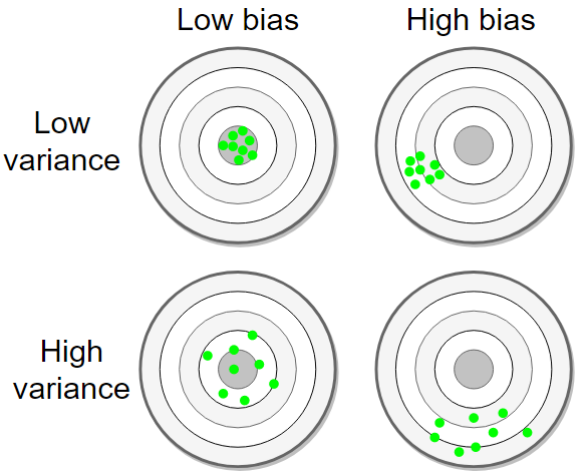


앙상블(Ensemble)

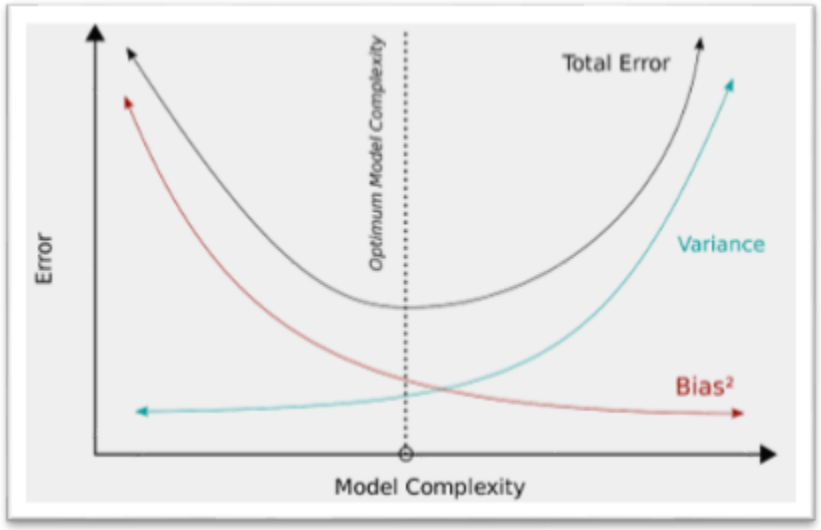
과대적합

Bias–variance tradeoff

- 편향(Bias) : 학습을 충분히 못했을 때 발생하는 오차
- 분산(Variance) : 학습 데이터에 과적합이 되었을 때 발생하는 오차



출처: http://www.machinelearningtutorial.net/2017/01/26/the-bias-variance-tradeoff/

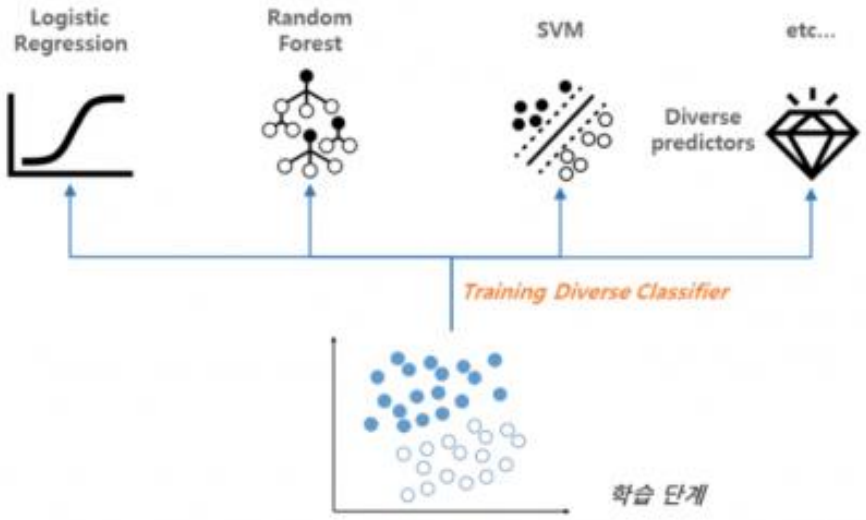


출처: https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff

앙상블 기법

Ensemble Learning

- 여러 개의 다른 모델을 결합하여 하나의 새로운 모델을 만드는 개념
- 분산(variance)을 감소시키는 효과, 편향(Bias)도 줄어드는 효과

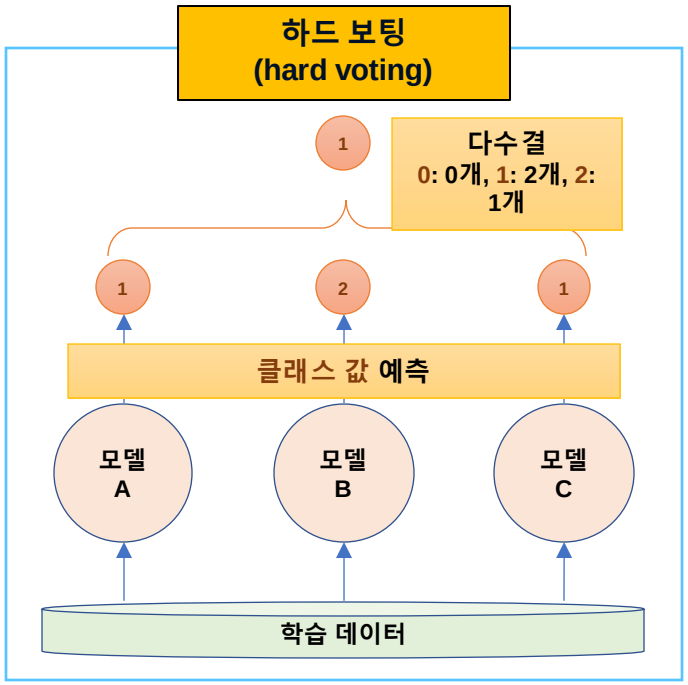


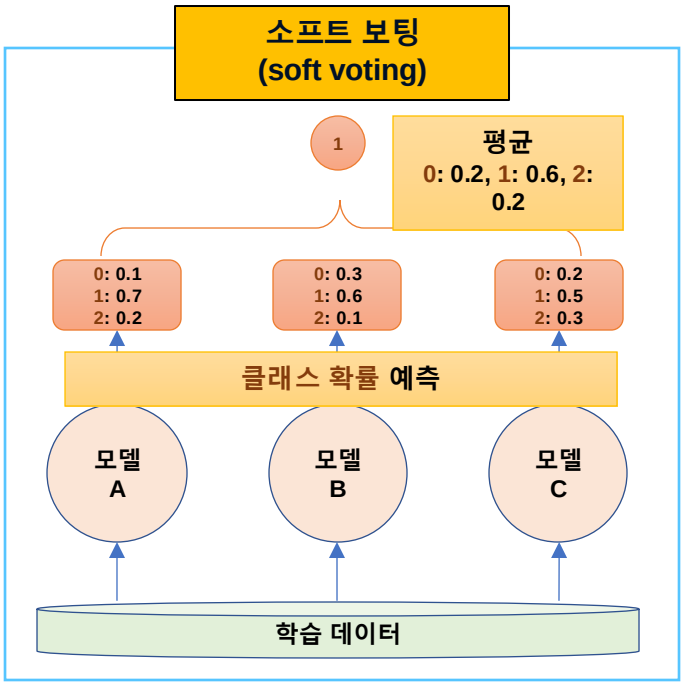
출처: https://itwiki.kr/w/%EC%95%99%EC%83%81%EB%B8%94_%EA%B8%B0%EB%B2%95

앙상블 기법

Voting (보팅)

- 여러 개의 모델이 예측한 값들 중에서 다수결로 최종 예측값을 결정

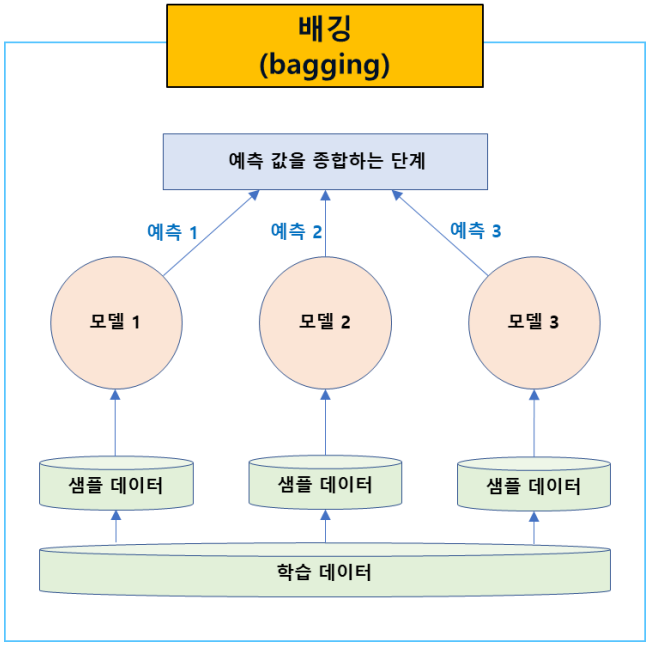




앙상블 기법

Bagging (배깅)

- Bootstrap 방식으로 데이터를 랜덤 Resampling(복원 추출) 후 결합

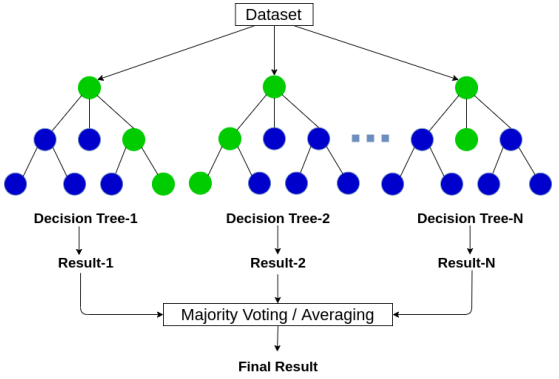


[Boostrap 방식의 장단점]

- **장점**: 분산(variance) 감소
- **단점**: 랜덤 중복 추출을 하면서, 일부 샘플은 계속 사용되고 일부 샘플은 전혀 사용되지 않을 가능성

[Random Forest]

- **여러 개의 의사결정나무를 활용하여 배깅 (Bagging)**

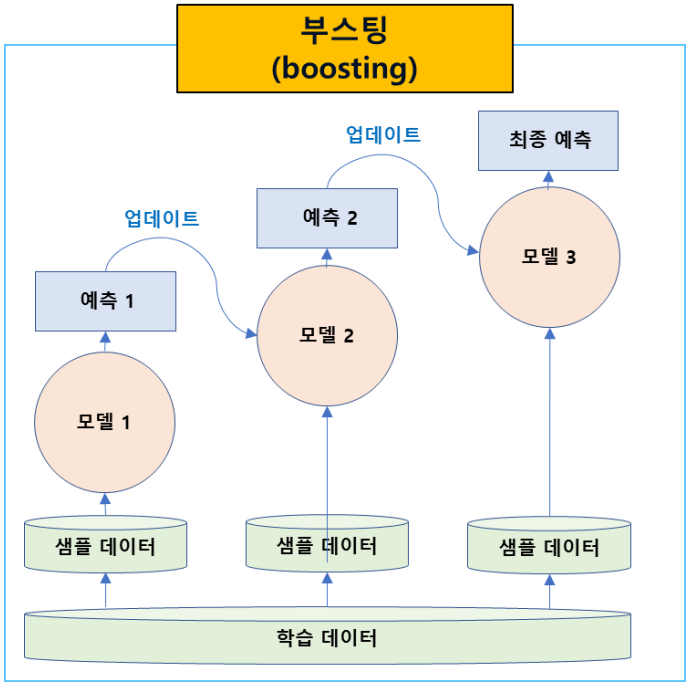


출처: https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/

앙상블 기법

Boosting (부스팅)

- 성능이 약한 모델을 순차적으로 학습시켜 성능을 높이는 방법



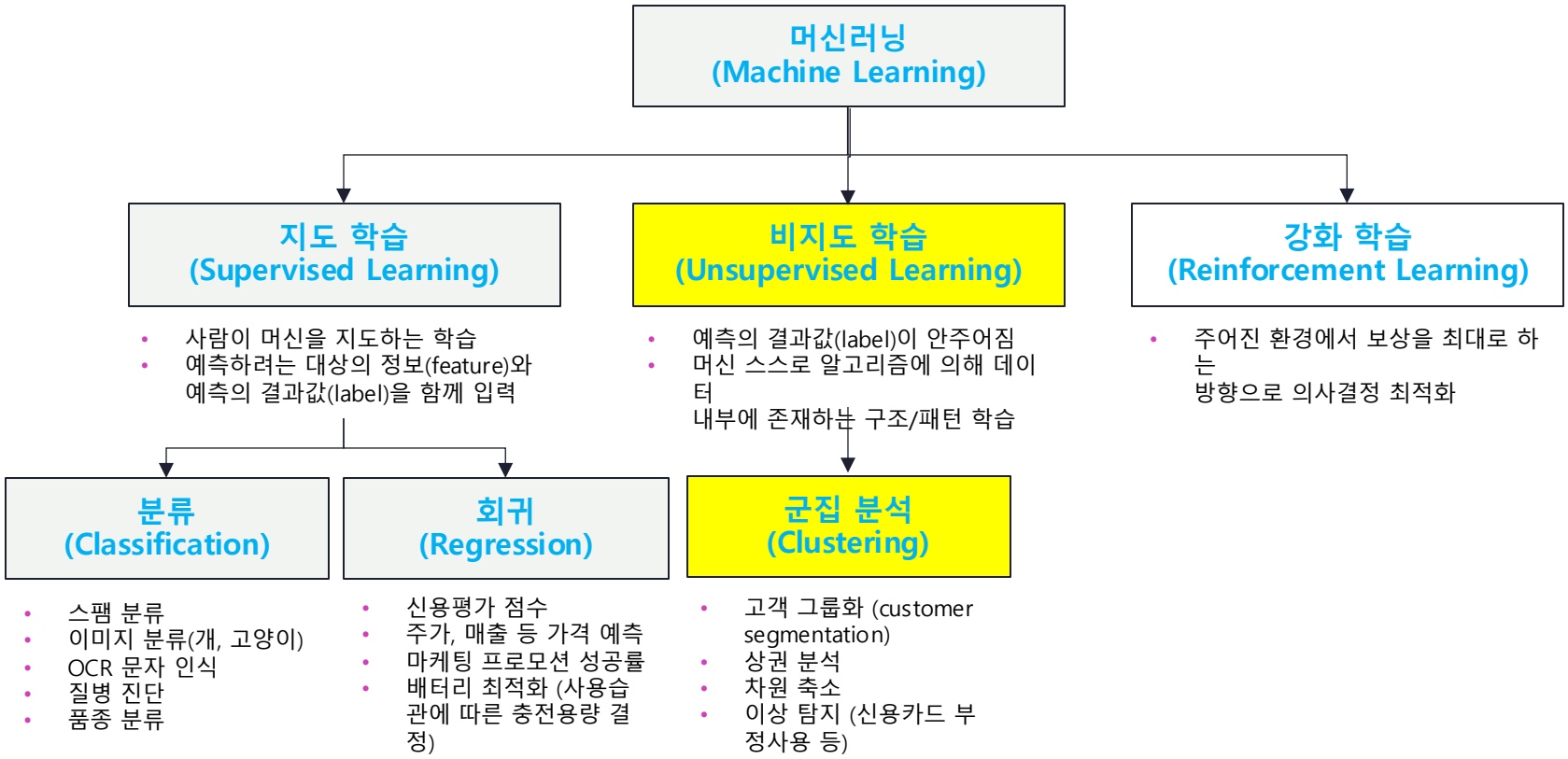
[Boosting 모델 장단점]

- **장점**: 오답에 가중치를 높여서, 오답에 더 집중
- **단점**: 이상치(outlier)가 있으면 높은 가중치를 부여하여 더 크게 영향을 받을 가능성

출처: https://towardsdatascience.com/what-is-boosting-in-machine-learning-2244aa196682

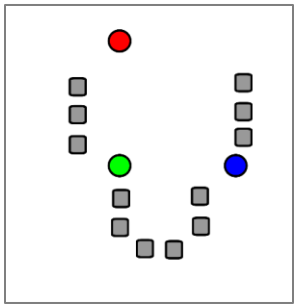
군집 분석

비지도 학습

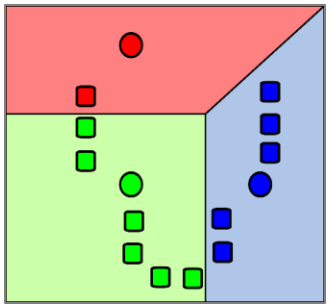

머신러닝
(Machine Learning)
지도 학습
(Supervised Learning)
비지도 학습
(Unsupervised Learning)
강화 학습
(Reinforcement Learning)
사람이 머신을 지도하는 학습
예측하려는 대상의 정보(feature)와 예측의 결과값(label)을 함께 입력
예측의 결과값(label)이 안주어짐
머신 스스로 알고리즘에 의해 데이터
내부에 존재하는 구조/패턴 학습
주어진 환경에서 보상을 최대로 하는
방향으로 의사결정 최적화
분류
(Classification)
회귀
(Regression)
군집 분석
(Clustering)
스팸 분류
이미지 분류(개, 고양이)
OCR 문자 인식
질병 진단
품종 분류
신용평가 점수
주가, 매출 등 가격 예측
마케팅 프로모션 성공률
배터리 최적화 (사용습관에 따른 충전용량 결정)
고객 그룹화 (customer segmentation)
상권 분석
차원 축소
이상 탐지 (신용카드 부정사용 등)

k-Means 군집분석

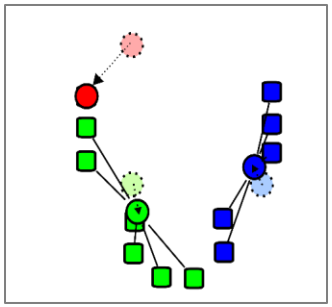
1. 전체 데이터를 k 개의 평균점을 갖는 군집으로 구분
2. 같은 군집의 데이터는 중심 가까이에 위치하고, 서로 다른 군집 간에 일정한 거리 이상 유지



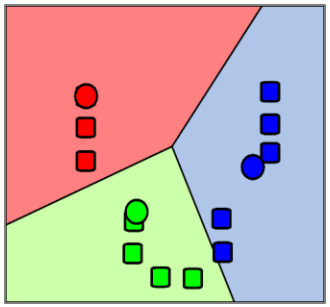
1) 초기 *k* "평균값" (위의 경우 *k*=3) 은 데이터 오브젝트 중에서 무작위로 뽑힌다. (색칠된 동그라미로 표시됨).



2) k 각 데이터 오브젝트들은 가장 가까이 있는 평균값을 기준으로 묶인다. 평균값을 기준으로 분할된 영역은 보로노이 다이어그램 으로 표시된다.



3) k개의 클러스터의 중심점을 기준으로 평균값이 재조정된다.

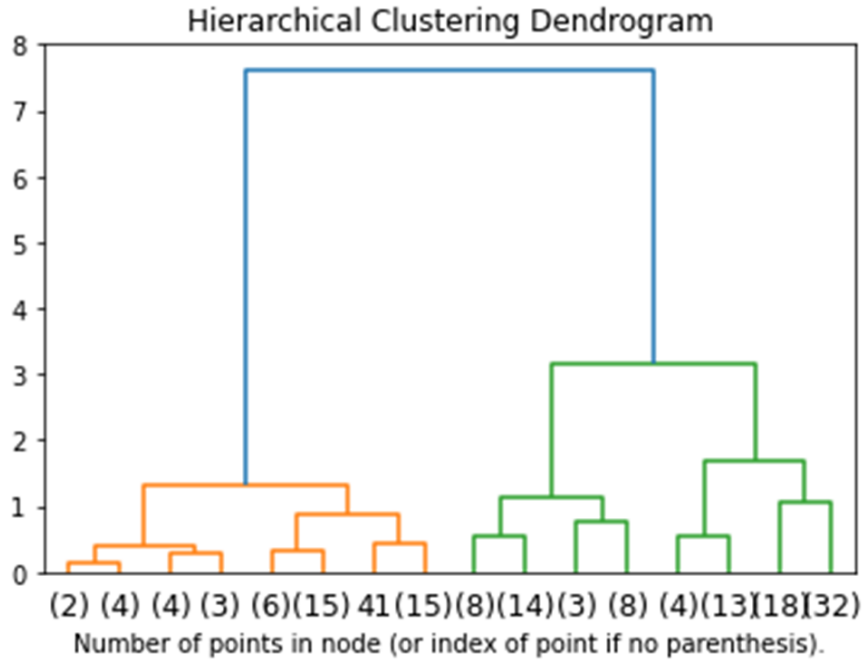


4) 수렴할 때까지 2), 3) 과정을 반복한다.

출처: https://ko.wikipedia.org/wiki/K-평균_알고리즘

계층적 군집화

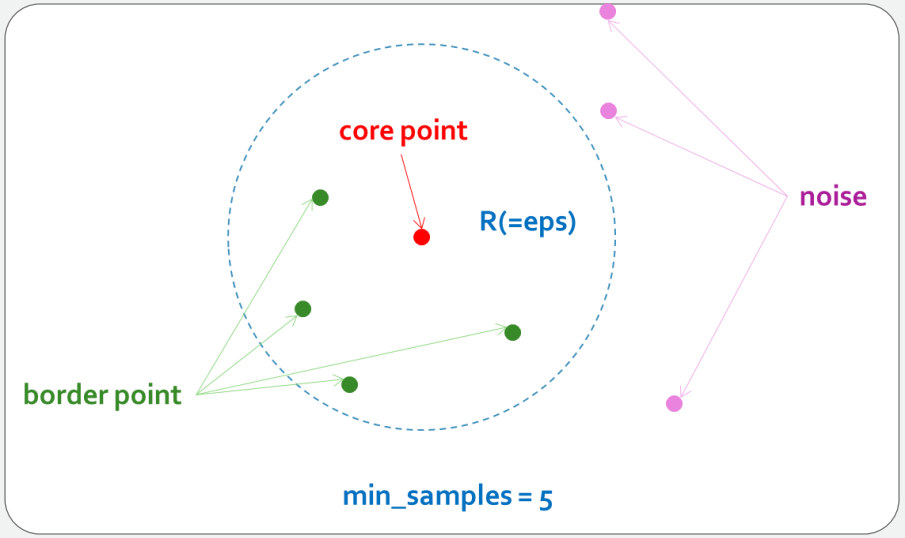
1. 서로 비슷한 데이터끼리 군집으로 결합
2. 군집끼리 서로 떨어진 거리를 계산
3. 가장 가까운 군집끼리 하나로 합침
4. 위 과정을 반복하면서 군집을 형성



DBSCAN

Density-Based Spatial Clustering of Applications with Noise

1. 각 데이터가 위치한 공간의 밀집도를 계산
2. 반지름(R) 안에 최소 샘플 개수를 갖는 데이터를 중심점(core point)라고 정의
3. 반지름(R) 안에 위치한 다른 데이터를 경계점(border point)라고 부름
4. 경계 밖의 데이터를 잡음(Noise)라고 정의
5. 반지름(R) 안에 함께 있는 중심점들을 결합하는 방식으로 군집을 형성



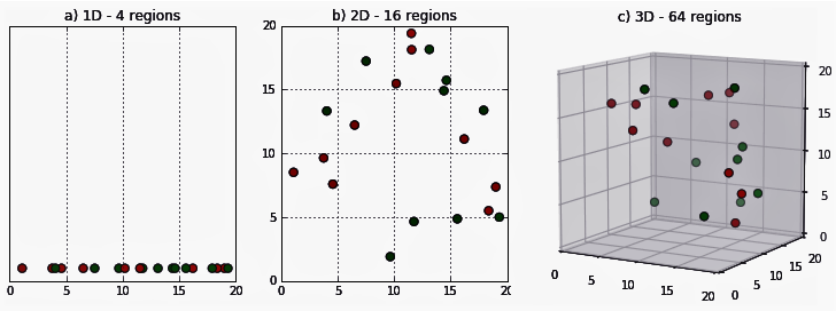
차원 축소

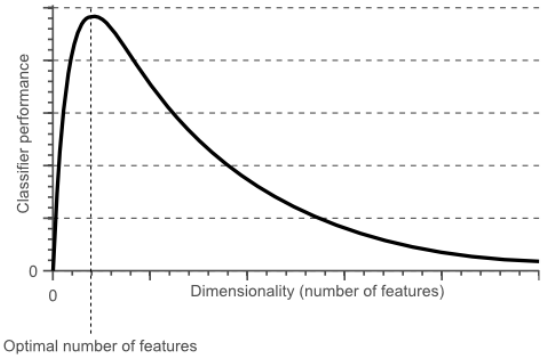
차원의 저주

차원의 저주

- 차원(특성 또는 변수의 개수)이 늘어나면, 데이터의 밀도가 급격하게 감소

- ML 모델의 성능은 차원이 늘면 일반적으로 향상되지만, 차원이 일정 수준 이상 커지면 급격하게 모델의 예측 성능이 나빠지는 현상

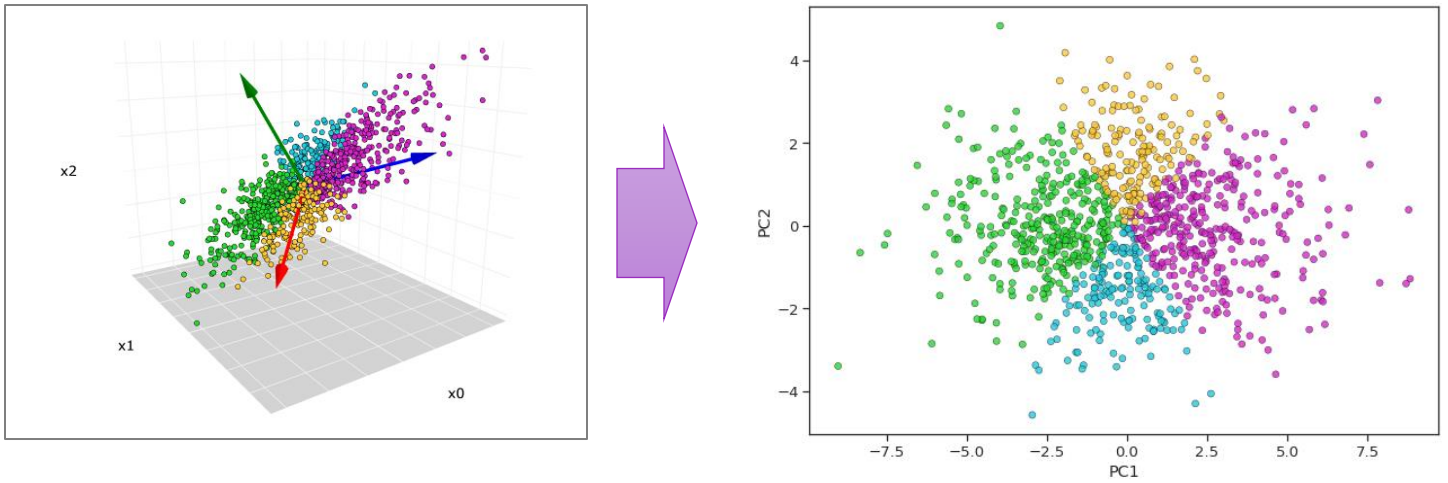




출처: https://medium.com/analytics-vidhya/the-curse-of-dimensionality-and-its-cure-f9891ab72e5c

주성분 분석 (PCA)

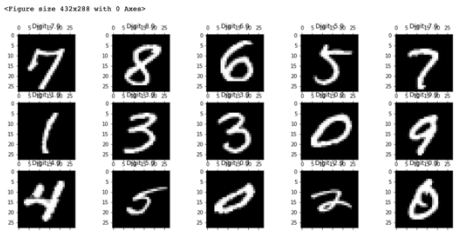
1. 고차원의 데이터를 저차원으로 변환
2. 데이터 분포를 서로 독립인 고유벡터를 축으로 갖는 공간으로 직교 변환
3. 데이터를 각 축으로 투영했을 때 분산이 가장 큰 축을 제1 주성분으로 하고,
   그 다음 큰 축을 제2 주성분으로 표현함
4. 주성분 탐색, 데이터 압축, 노이즈 제거에 활용

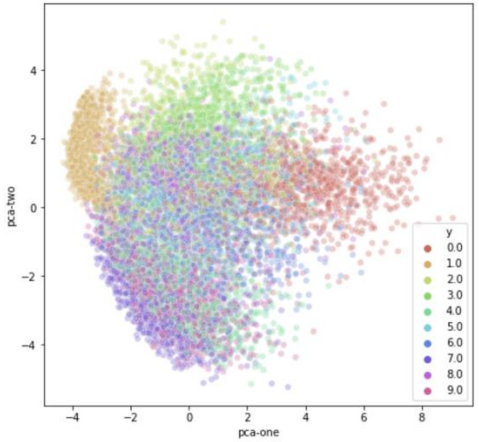


출처: https://towardsdatascience.com/principal-component-analysis-pca-explained-visually-with-zero-math-1cbf392b9e7d

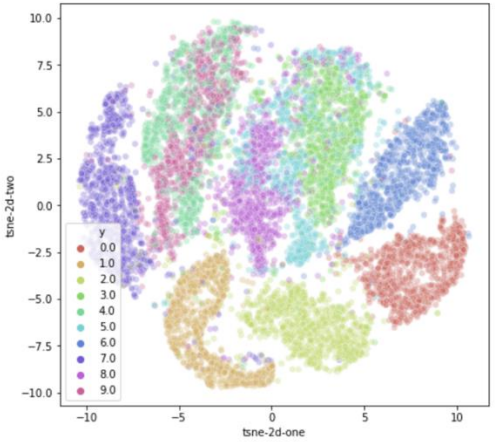
t-SNE

1. 고차원의 데이터를 저차원으로 변환
2. 고차원에서의 데이터 간의 상대적 거리를 저차원 공간에서도 그대로 유지
3. 시각화 목적으로 한정하여 사용





PCA



t-SNE

출처: https://towardsdatascience.com/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b