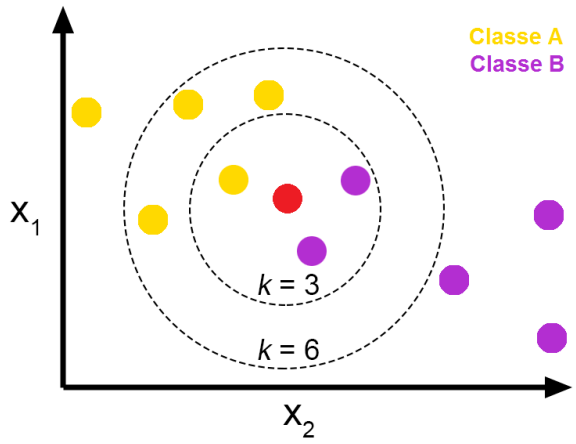
ML 알고리즘

강사: 김남범(nbumkim@gmail.com)

K-최근접 분류/회귀

K-nearest neighbors classification

- 지도학습으로서 분류(Classification) 나 회귀(Regression)에 사용되는 비모수적 방법
- 파라메터 학습을 위한 훈련과정이 없으나 훈련집합은 필요
- 각 데이터 간에 거리를 계산하기 위한 거리척도가 필요
- 초매개변수 k를 설정해야 함
- 거리에 대한 가중치



https://bkshin.tistory.com

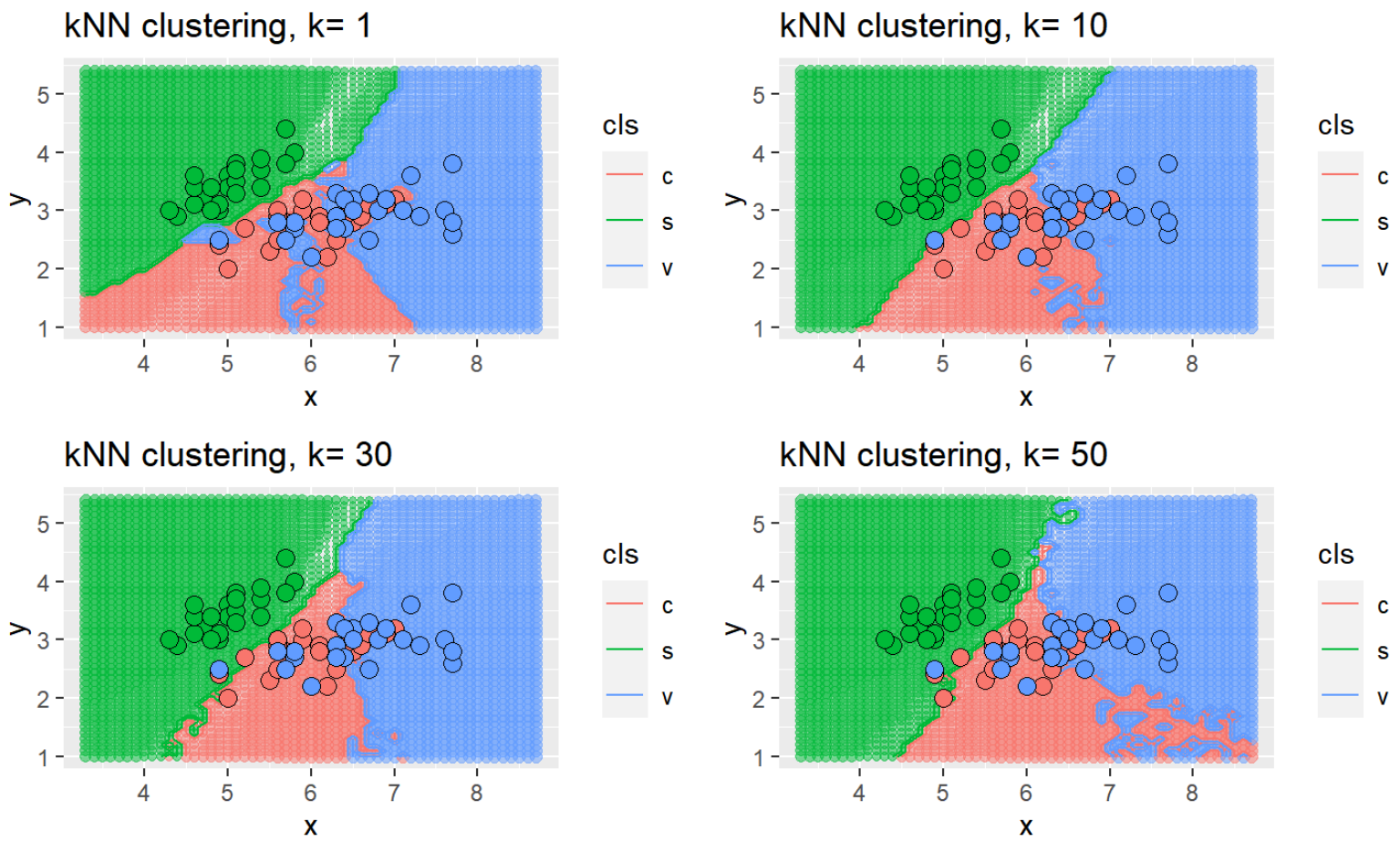
기하학적 거리 (Geometric distance measures)

- Euclidean: $d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{p}(x_i - y_i)^2}, \quad \vec{x}, \vec{y} \in p$
- Manhattan: $d(\vec{x}, \vec{y}) = \sum_{i=1}^{p} |x_i - y_i|$
- Minkowski: $d(\vec{x}, \vec{y}) = (\sum_{i=1}^{p} |x_i - y_i|^q)^{\frac{1}{q}}$
- Gower: Manhattan(Continuous) + Dice coefficient(Nominal)

type	length	width
품종 A	10	3
품종 A	12	6
품종 B	14	5
품종 B	15	4
품종 B	16	8
new	12	4

```
# A tibble: 6 x 5
  type    length width sq_sum abs_sum
  <chr>    <dbl> <dbl>  <dbl>   <dbl>
1 품종 A      10     3   2.24       3
2 품종 A      12     6   2          2
3 품종 B      14     5   2.24       3
4 품종 B      15     4   3          3
5 품종 B      16     8   5.66       8
6 new         12     4   0          0
```

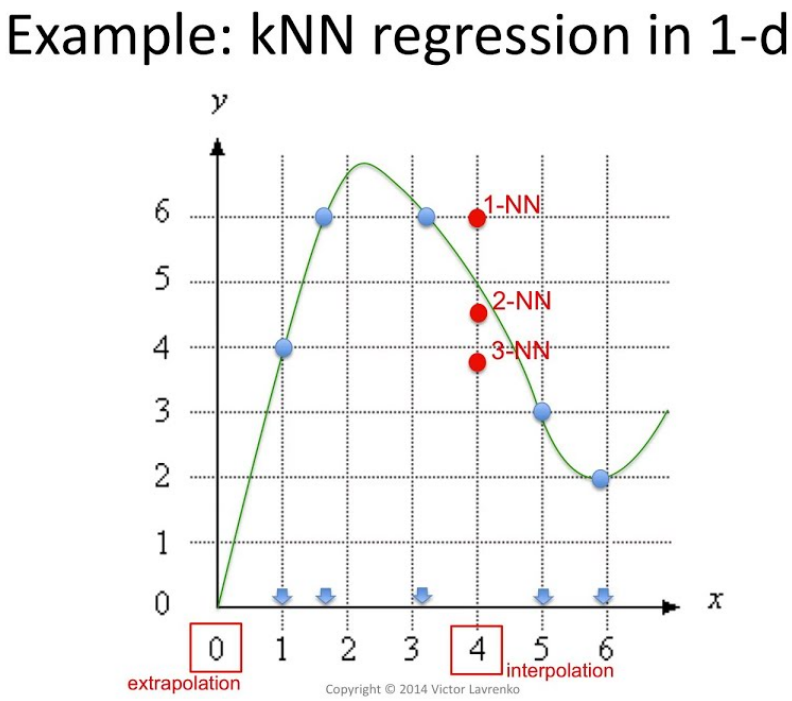
KNN clustering area vs k



K-최근접 회귀 (K-nearest neighbors regression, KNR)

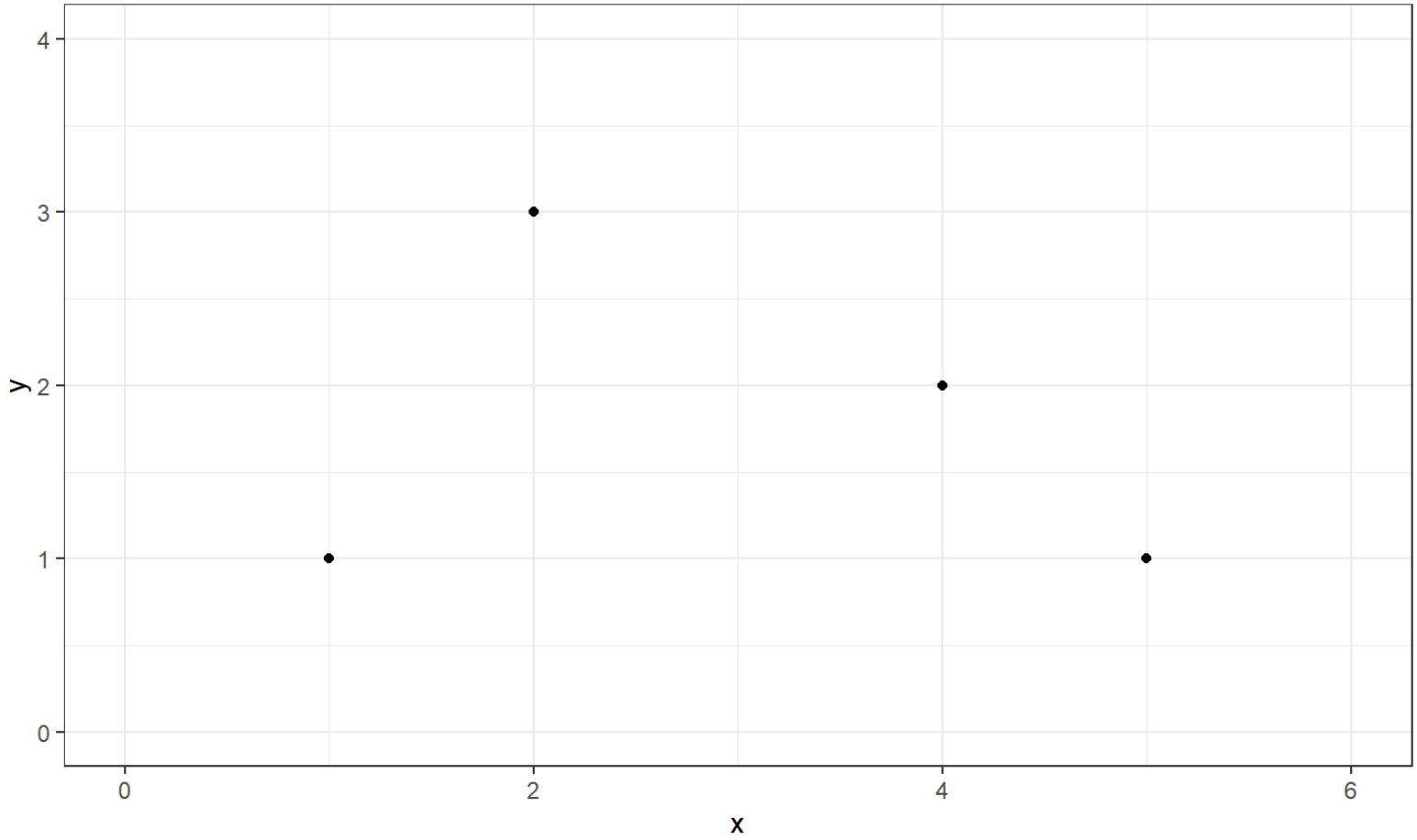
KNR 예제

- 초매개변수 k를 설정해야 함
- 거리에 대한 가중치로 weight하는 방법이 사용될 수 있음

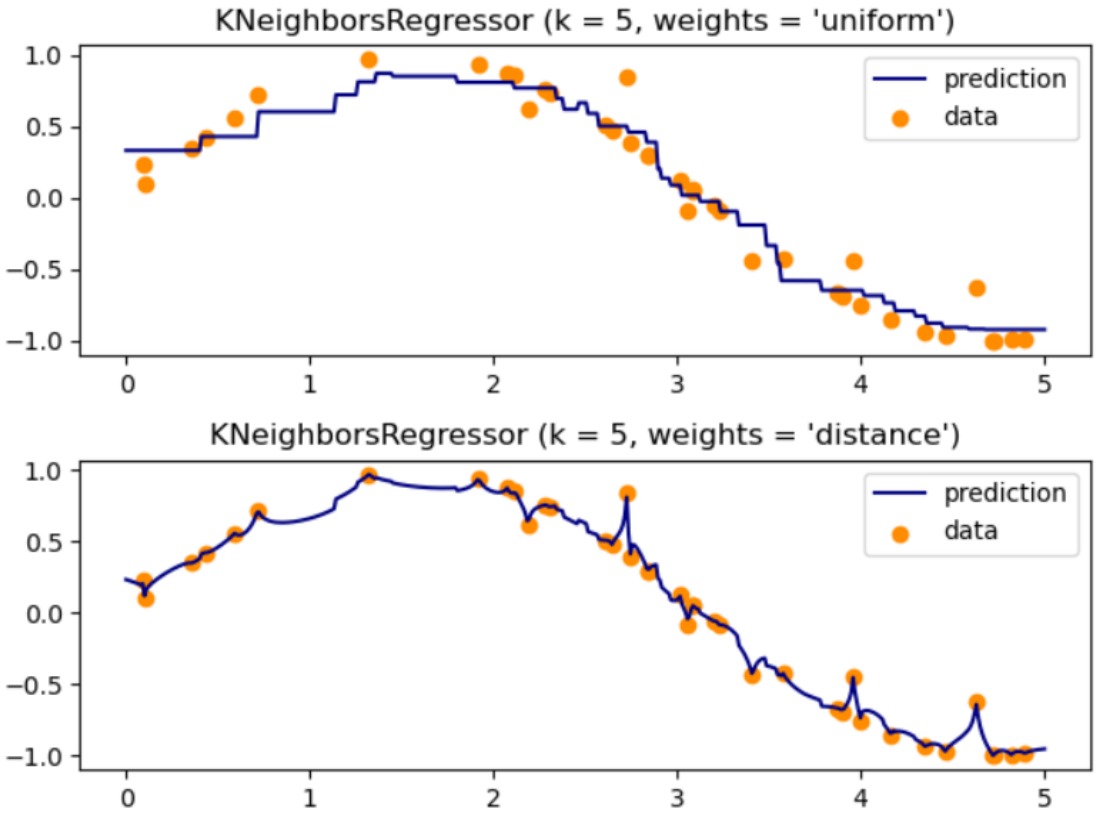


https://www.youtube.com/watch?v=3lp5CmSwrHI

Hand-on KNR



KNR weight 'Uniform' vs 'distance'



https://scikit-learn.org/stable/auto_examples/neighbors/plot_regression.html#sphx-glr-auto-examples-neighbors-plot-regression-py

선형회귀 (Linear regression)

선형회귀

- 지도학습
- 목적변수 (반응변수)가 연속형인 경우
- 정규성, 독립성, 등분산성을 만족해야 함
- Feature 가 하나인 경우 단순회귀 (Simple), Feature 가 여러 개 인 경우 중회귀 (Multiple)
- 2차항 이상이 포함된 경우 다항회귀 (Polynomial regression)

단순회귀 (Simple linear regression)

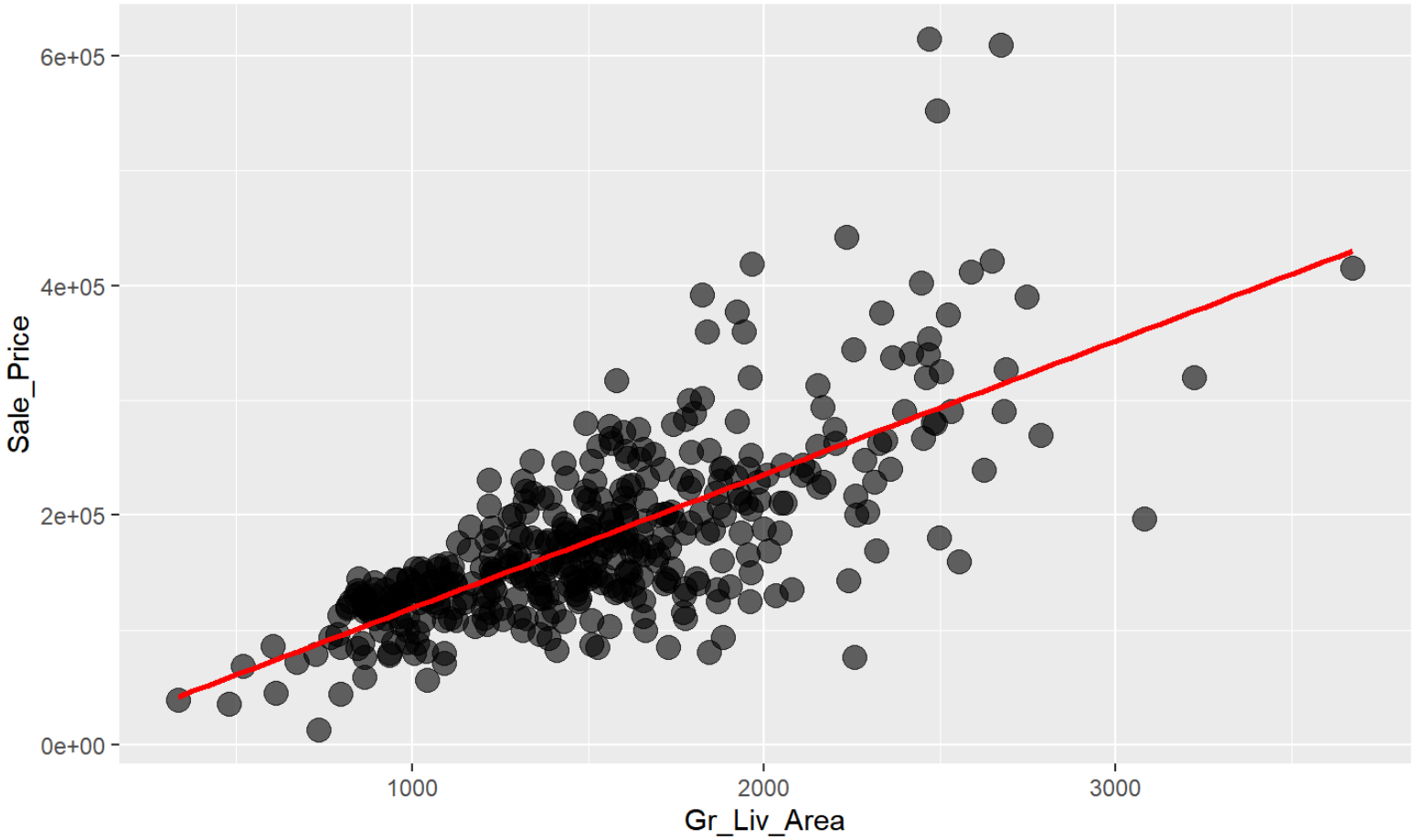
$$f(x_i) = Y_i = \beta_0 + \beta_1 X_i + \epsilon_i, \quad for\ i = 1, 2, \ldots, n$$

- $\epsilon_i \sim^{i.i.d} N(0, \sigma^2),\ \forall i$
- $Independency,\ Normality,\ Homoscedasticity$

```
head(ames[, c("Sale_Price", "Gr_Liv_Area")], n = 10)
```

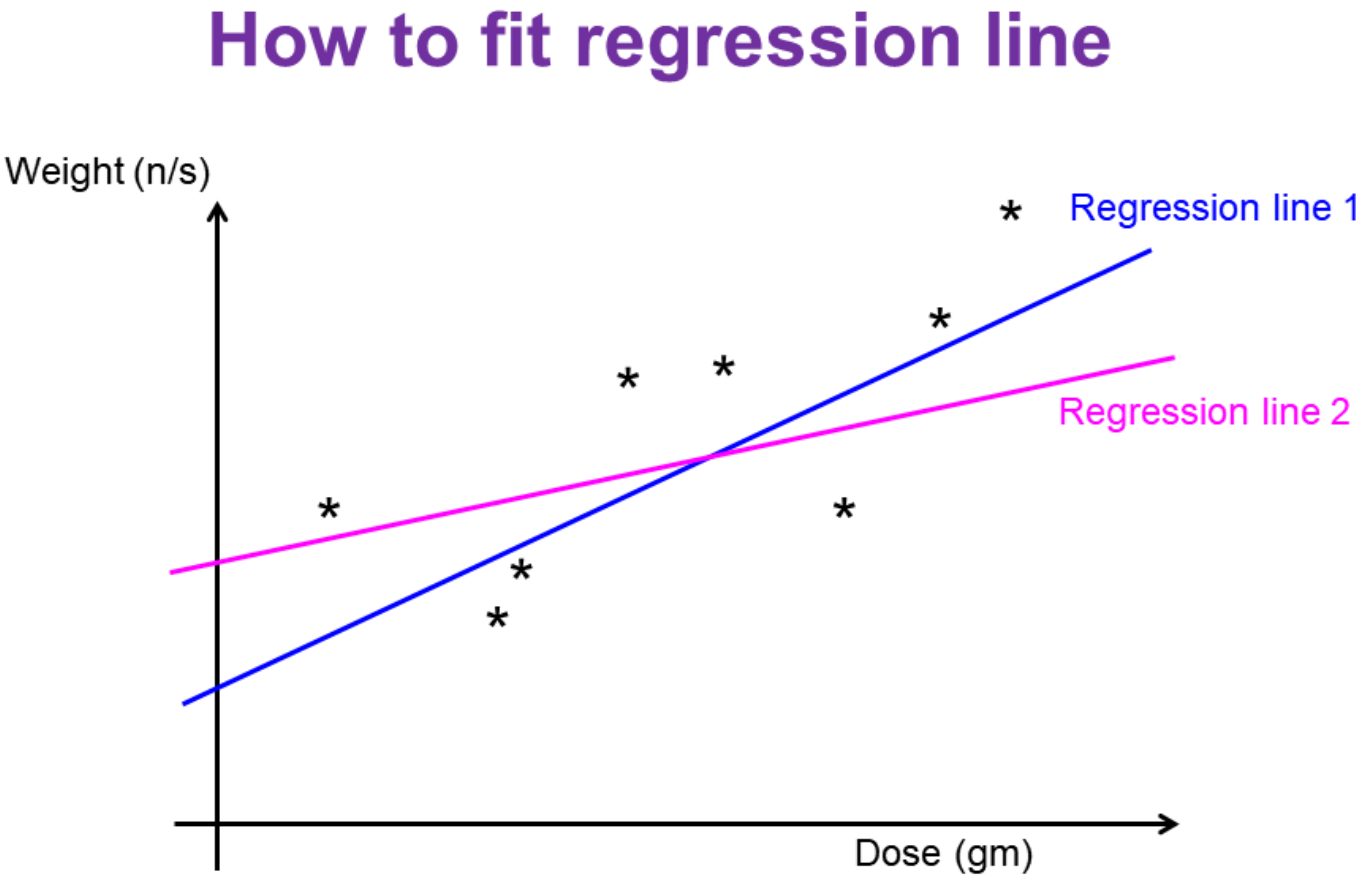
```
# A tibble: 10 x 2
   Sale_Price Gr_Liv_Area
        <int>       <int>
 1     215000        1656
 2     105000         896
 3     172000        1329
 4     244000        2110
 5     189900        1629
 6     195500        1604
 7     213500        1338
 8     191500        1280
 9     236500        1616
10     189000        1804
```

Gr_Liv_Area 와 Sale_Price 간의 관계



최소제곱추정량 (Ordinary least square estimation)

- Linear model, $\hat{y}(x_i) = \hat{\theta}_0 + \hat{\theta}_1 x_i$



손실함수 (Loss function, J)

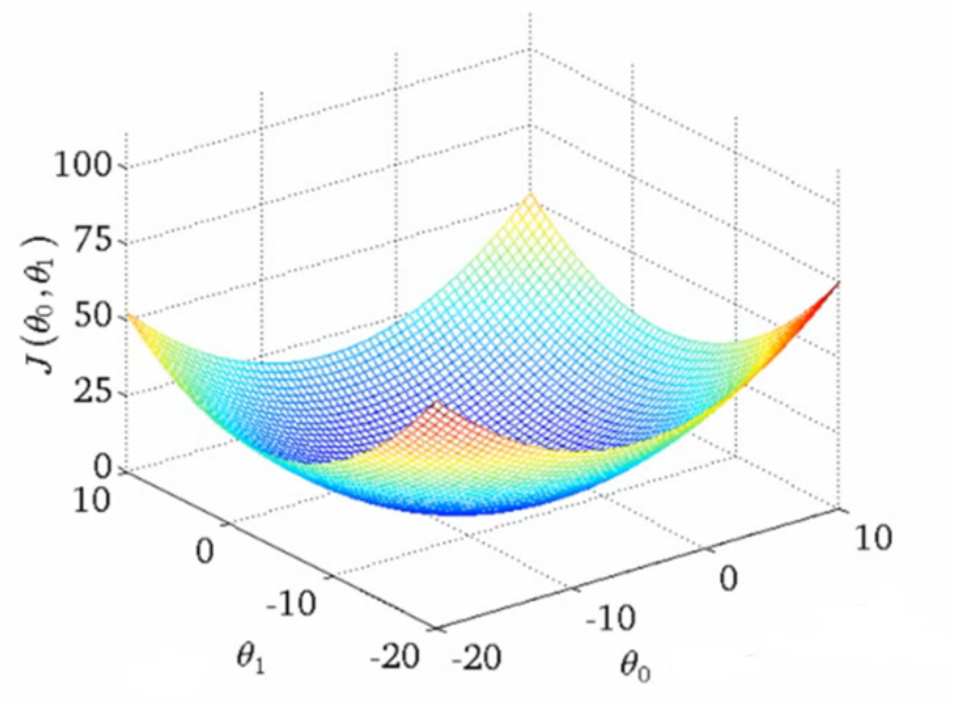
$J(\theta) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y_i - \hat{\theta}_0 - \hat{\theta}_1 x)^2 = \sum_{i=1}^{n} e_i^2$

- $\hat{y}_i = \hat{\theta}_0 + \hat{\theta}_1 x_i$
- Sum of Squares of the Errors (SSE) = $\sum_i^n e_i^2$

- Goal is to solve for $\hat{\theta}_0$ and $\hat{\theta}_1$ to minimize the objective function.

$\hat{\theta}_o, \hat{\theta}_1 = argmin_{\theta_0,\theta_1} \sum_i^n e_i^2 = argmin_{\theta_0,\theta_1} J(\theta)$

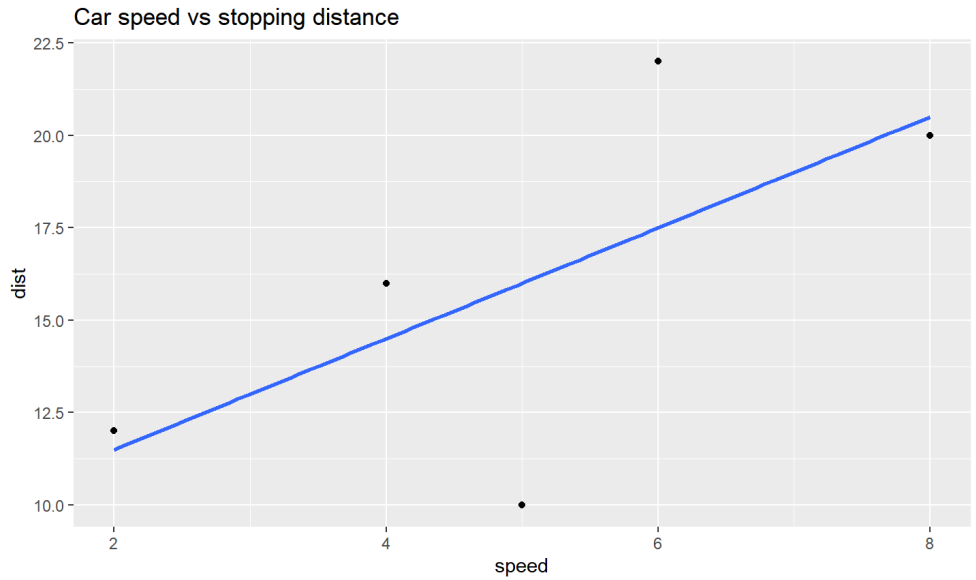
Convex function



- $\frac{\partial J(\theta)}{\partial \hat{\theta}_0} = 0, \quad \hat{\theta}_0 = \overline{y} - \hat{\theta}_1 \overline{x}$
- $\frac{\partial J(\theta)}{\partial \hat{\theta}_1} = 0, \quad \hat{\theta}_1 = \frac{\sum_i^n (x_i - \overline{x})(y_i - \overline{y})}{\sum_i^n (x_i - \overline{x})^2}$
- $\sum_i^n (x_i - \overline{x})(y_i - \overline{y}) = S_{xy}$
- $\sum_i^n (x_i - \overline{x})^2 = S_{xx}$

자동차 속력과 제동거리 간의 관계

```
# A tibble: 5 x 4
  speed speed_mean  dist dist_mean
  <dbl>      <dbl> <dbl>     <dbl>
1     2          5    12        16
2     4          5    16        16
3     5          5    10        16
4     8          5    20        16
5     6          5    22        16
```



중회귀 (Multiple linear regression)

$$f(x_i) = Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \epsilon_i, \quad for\ i = 1, 2, \ldots, n$$

- $\epsilon_i \sim^{i.i.d} N(\beta^T X, \sigma^2),\ \forall i$ - $\beta^T X = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i}$

```
head(ames[, c("Sale_Price", "Gr_Liv_Area", "Year_Built")], n = 6)
```

```
# A tibble: 6 x 3
  Sale_Price Gr_Liv_Area Year_Built
       <int>       <int>      <int>
1     215000        1656       1960
2     105000         896       1961
3     172000        1329       1958
4     244000        2110       1968
5     189900        1629       1997
6     195500        1604       1998
```

Multiple Regression Model in Matrix Form

- In matrix notation the multiple regression model is: $\mathbf{Y}=\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon}$ where

$$\mathbf{Y}=\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, \quad \varepsilon=\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}, \quad \beta=\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_\kappa \end{bmatrix}, \quad \mathbf{X}=\begin{bmatrix} 1 & X_{11} & \cdots & X_{1k} \\ 1 & X_{21} & & X_{2k} \\ \vdots & \vdots & & \vdots \\ 1 & X_{n1} & & X_{nk} \end{bmatrix}$$

- Note, $\mathbf{Y}$ and $\boldsymbol{\varepsilon}$ are $n\times 1$ vectors, $\boldsymbol{\beta}$ is a $(k+1)\times 1$ vector and $\mathbf{X}$ is a $n\times(k+1)$ matrix.
- The Gauss-Markov assumptions are: $E(\boldsymbol{\varepsilon})=\mathbf{0}$, $\mathrm{Var}(\boldsymbol{\varepsilon})=\sigma^2\mathbf{I}$.
- These result in $E(\mathbf{Y})=\mathbf{0}$, $\mathrm{Var}(\mathbf{Y})=\sigma^2\mathbf{I}$.
- The Least-Square estimate of $\boldsymbol{\beta}$ is $\mathbf{b}=(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$.

https://www.slideserve.com/verne/analysis-of-variance-in-matrix-form

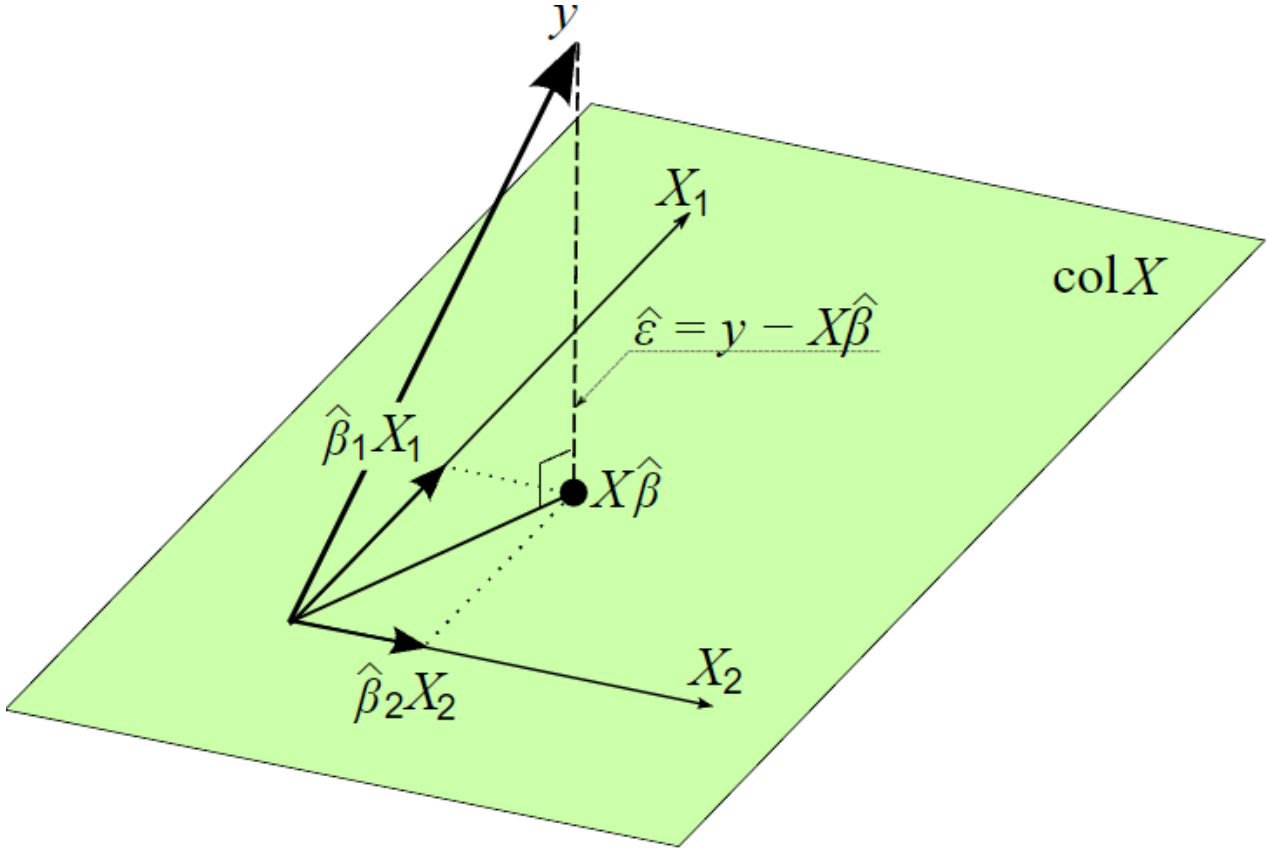
OLS of Multiple linear regression

$\hat{\beta} = (X'X)X^{-1}y$

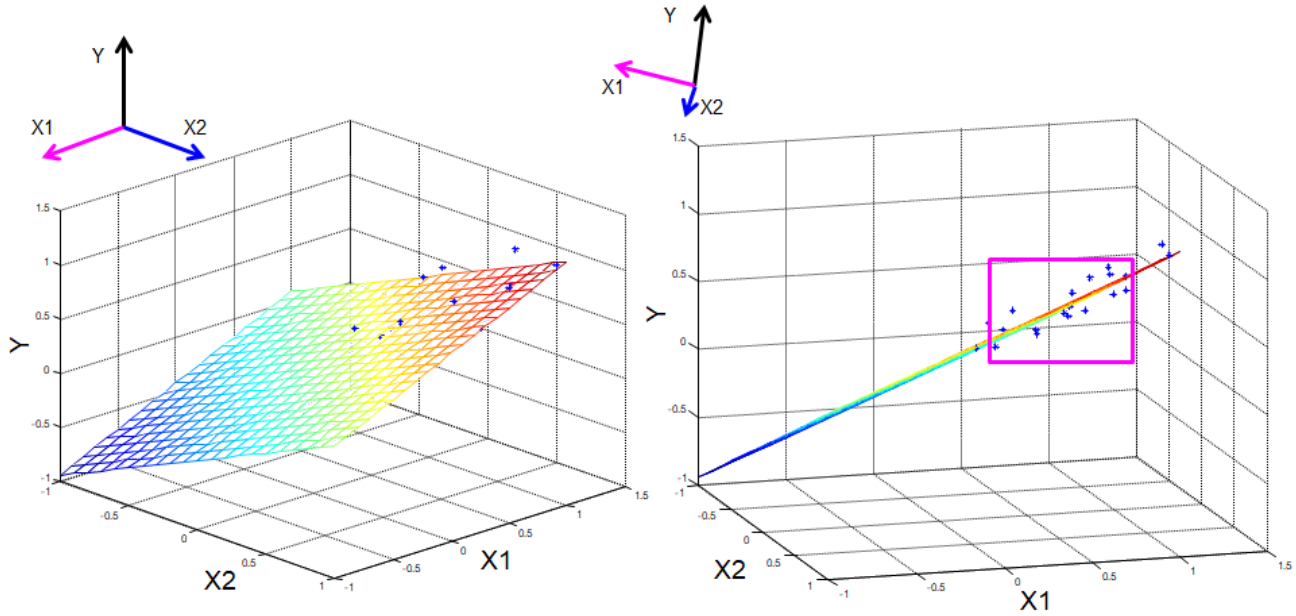
where, X = Feature matrix

y = Target vector

OLS geometric interpretation

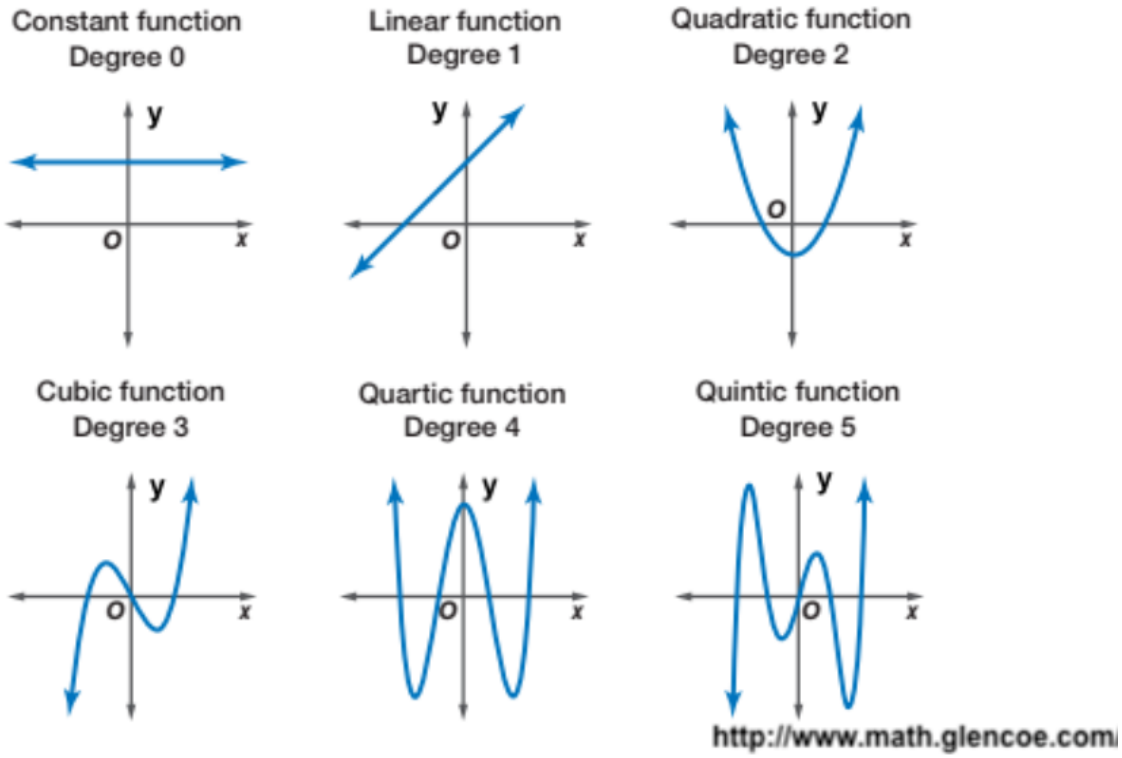


2D regression plane

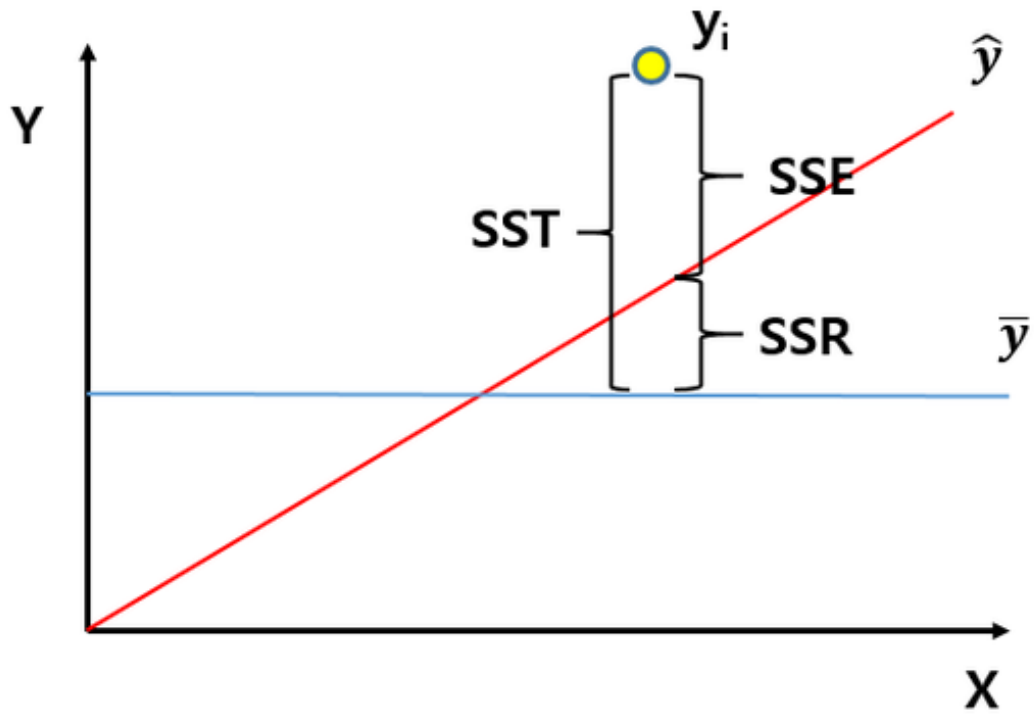


다항회귀 (Polynomial regression)

- $y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_p x^p + \epsilon_i$



결정계수 (R-squared)



SST(Y의 전체 변동) : $\sum(y_i - \bar{y})^2$

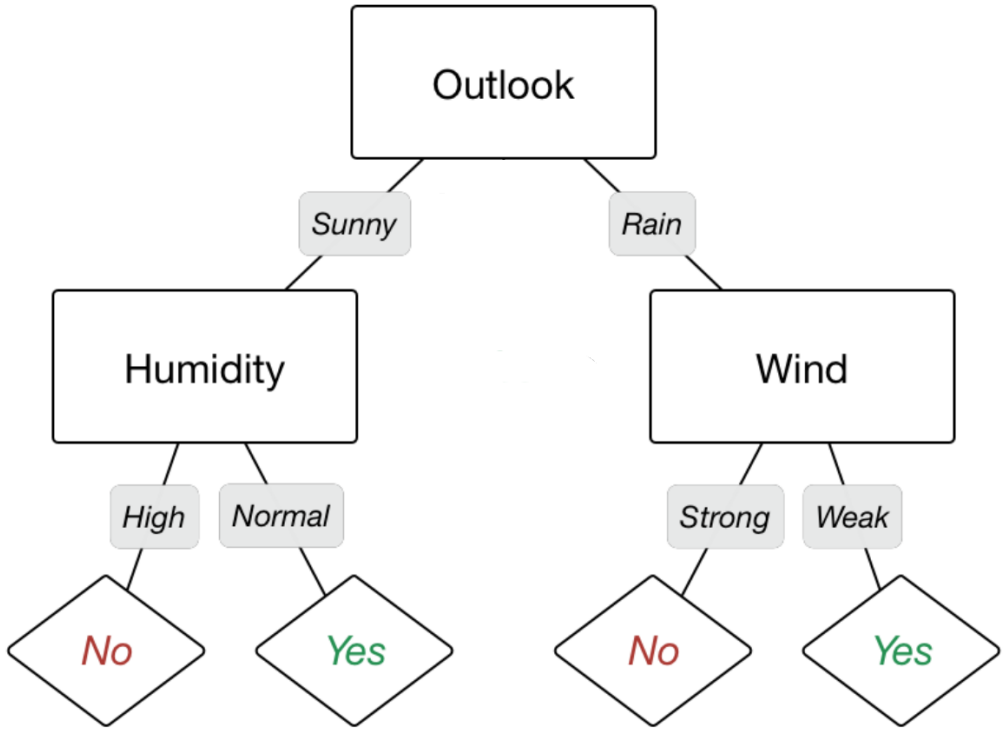
SSR(모형에 의해 설명되는 변동) : $\sum(\hat{y}_i - \bar{y})^2$

SSE(모형에 의해 설명이 되지 않는 변동) : $\sum(y_i - \hat{y}_i)^2$

결정트리 (Decision tree) 과 랜덤포레스트 (Random forest)

Structure

- Decision tree for conditions to play tennis



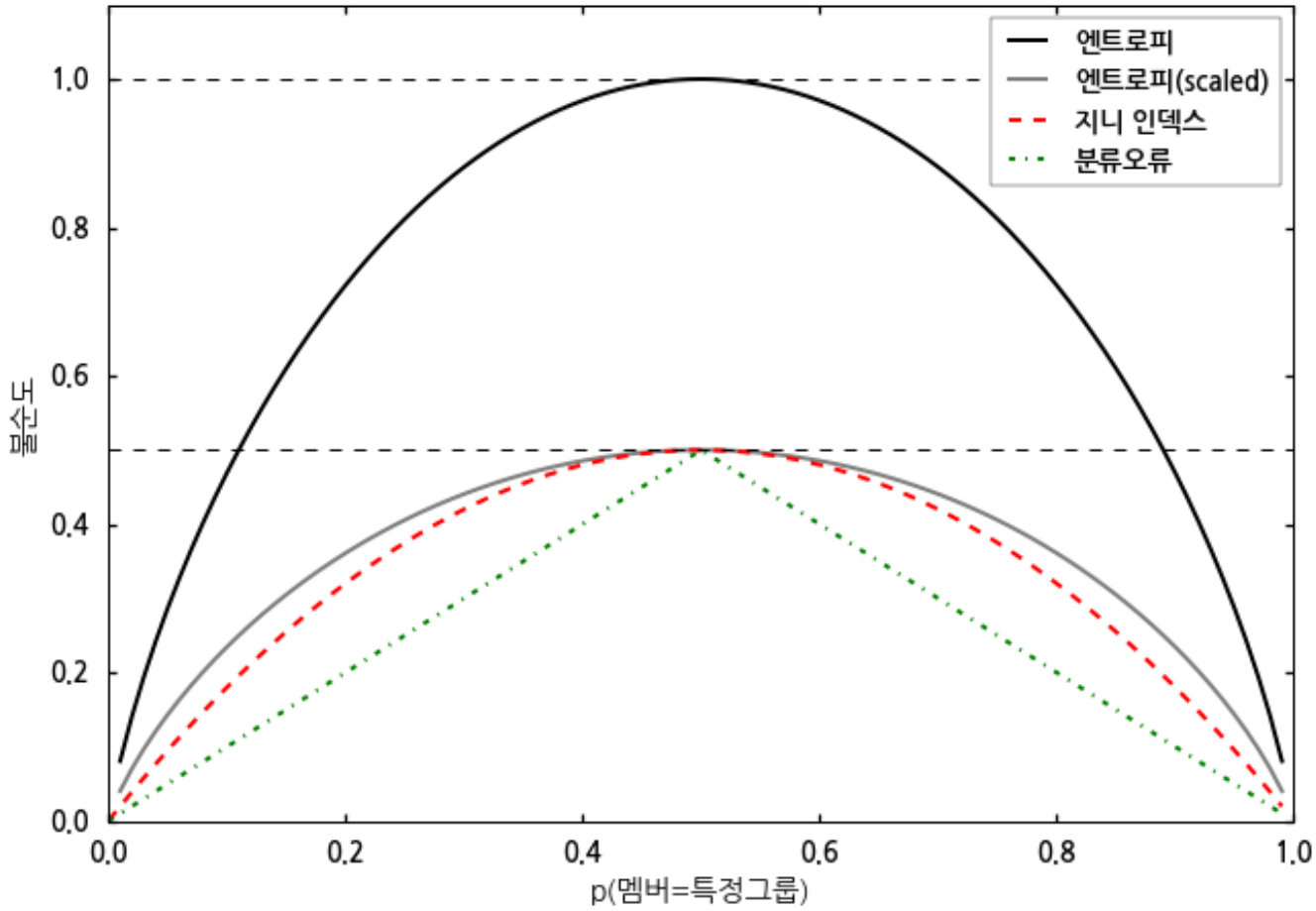
https://medium.com/@anshulraghav2222/decision-tree-classification-9390d6038ac8

Partitioning rule of classification case

- Maximize the reduction in entropy or the Gini index
  - Gini index: $I_G = 1 - \sum_{j=1}^{c} p_j^2$, c is number of class
  - Entropy: $Entropy(p_j) = -\sum_{j=1}^{c} p_j log(p_j)$

Disease	Hypertension	chestPain	Cholesterol
Yes	No	Yes	208
Yes	No	No	282
Yes	No	Yes	235
No	Yes	Yes	277
Yes	No	Yes	280
No	Yes	Yes	242
No	No	No	275
No	Yes	No	214
Yes	Yes	Yes	231
Yes	Yes	No	206

Ginni and Entropy



https://m.blog.naver.com/samsjang/220978650404

Decision tree when many features are available

- Iris data

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa

The decision tree boundary of Iris data

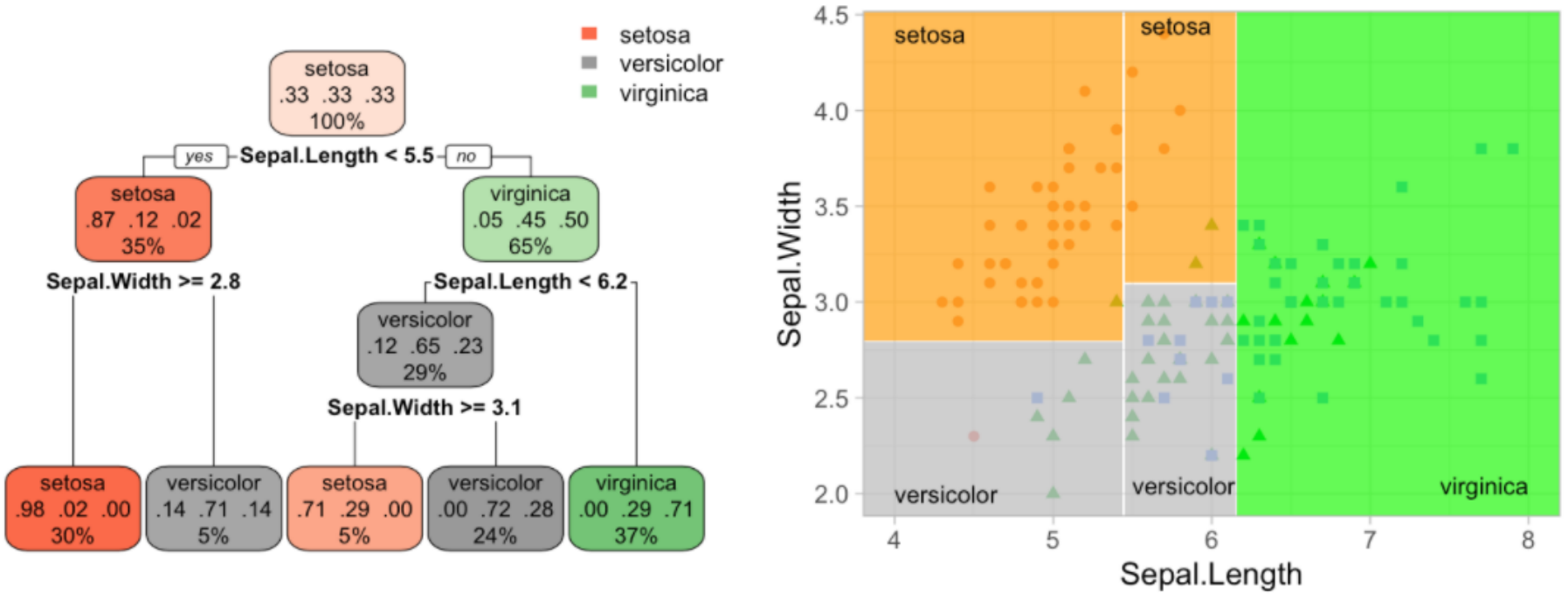


Figure 9.5: Decision tree for the iris classification problem (left). The decision boundary results in rectangular regions that enclose the observations. The class with the highest proportion in each region is the predicted value (right).

https://bradleyboehmke.github.io/HOML

Partitioning rule of regression case

- Minimize the total SSE
  $SSE = \sum_{i \in R_1} (y_i - c_1)^2 + \sum_{i \in R_2} (y_i - c_1)^2$

	mpg	wt	vs	cyl
Mazda RX4	21.0	2.620	0	6
Mazda RX4 Wag	21.0	2.875	0	6
Datsun 710	22.8	2.320	1	4
Hornet 4 Drive	21.4	3.215	1	6
Hornet Sportabout	18.7	3.440	0	8
Valiant	18.1	3.460	1	6
Duster 360	14.3	3.570	0	8
Merc 240D	24.4	3.190	1	4
Merc 230	22.8	3.150	1	4
Merc 280	19.2	3.440	1	6

Decision tree illustrating the single split on feature x

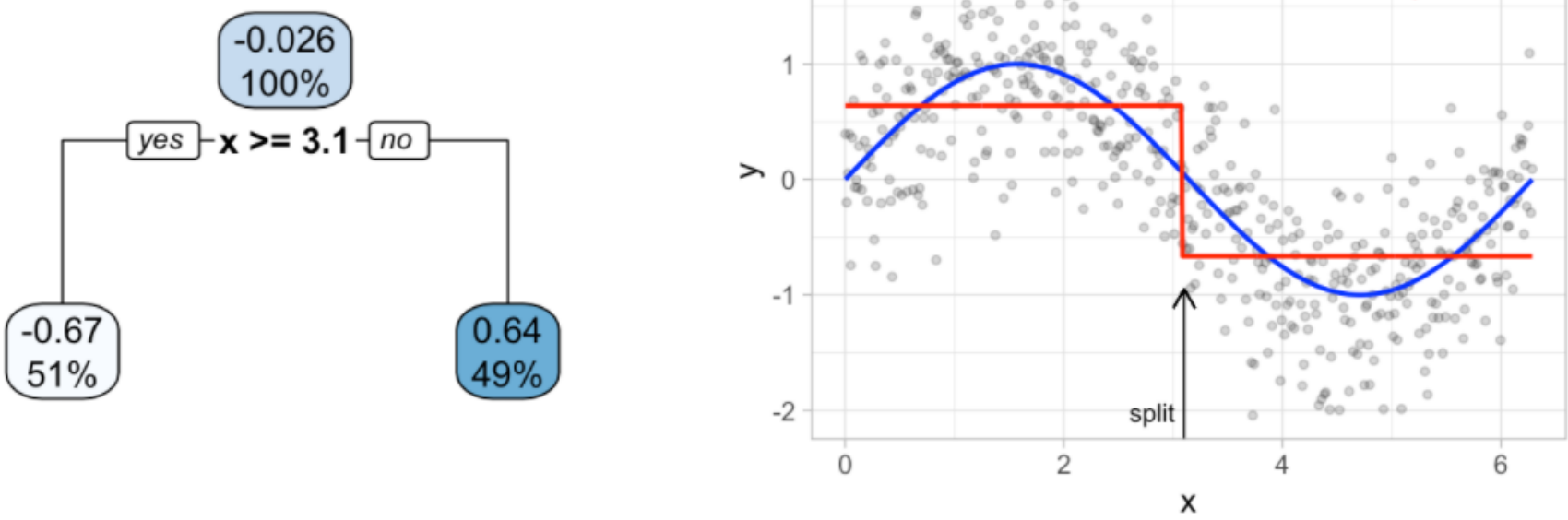


Figure 9.3: Decision tree illustrating the single split on feature x (left). The resulting decision boundary illustrates the predicted value when x < 3.1 (0.64), and when x > 3.1 (-0.67) (right).

https://bradleyboehmke.github.io/HOML

Decision tree illustrating with depth = 3

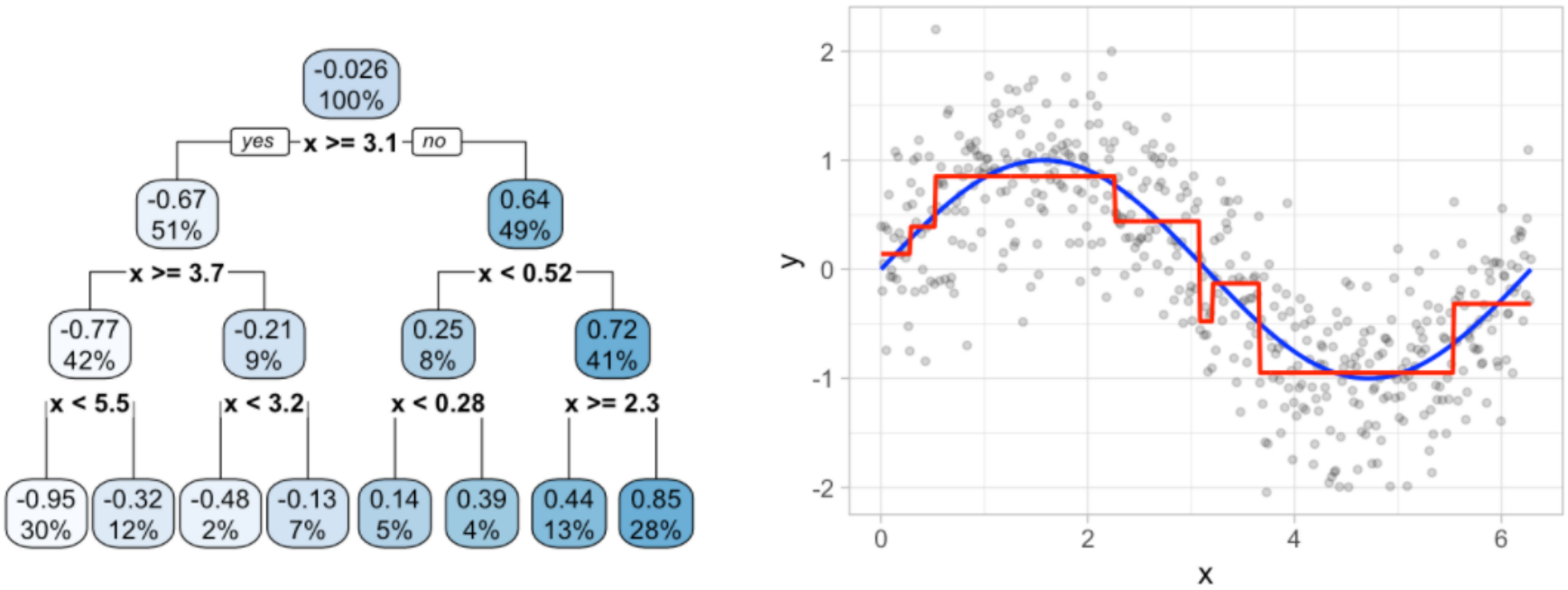


Figure 9.4: Decision tree illustrating with depth = 3, resulting in 7 decision splits along values of feature x and 8 prediction regions (left). The resulting decision boundary (right).

https://bradleyboehmke.github.io/HOML

Properties of decision tree

- No pre-processing requirements (e.g., log)
  - Not required to meet algorithm assumptions (e.g. Gaussian distribution)
  - Outliers typically do not bias the results

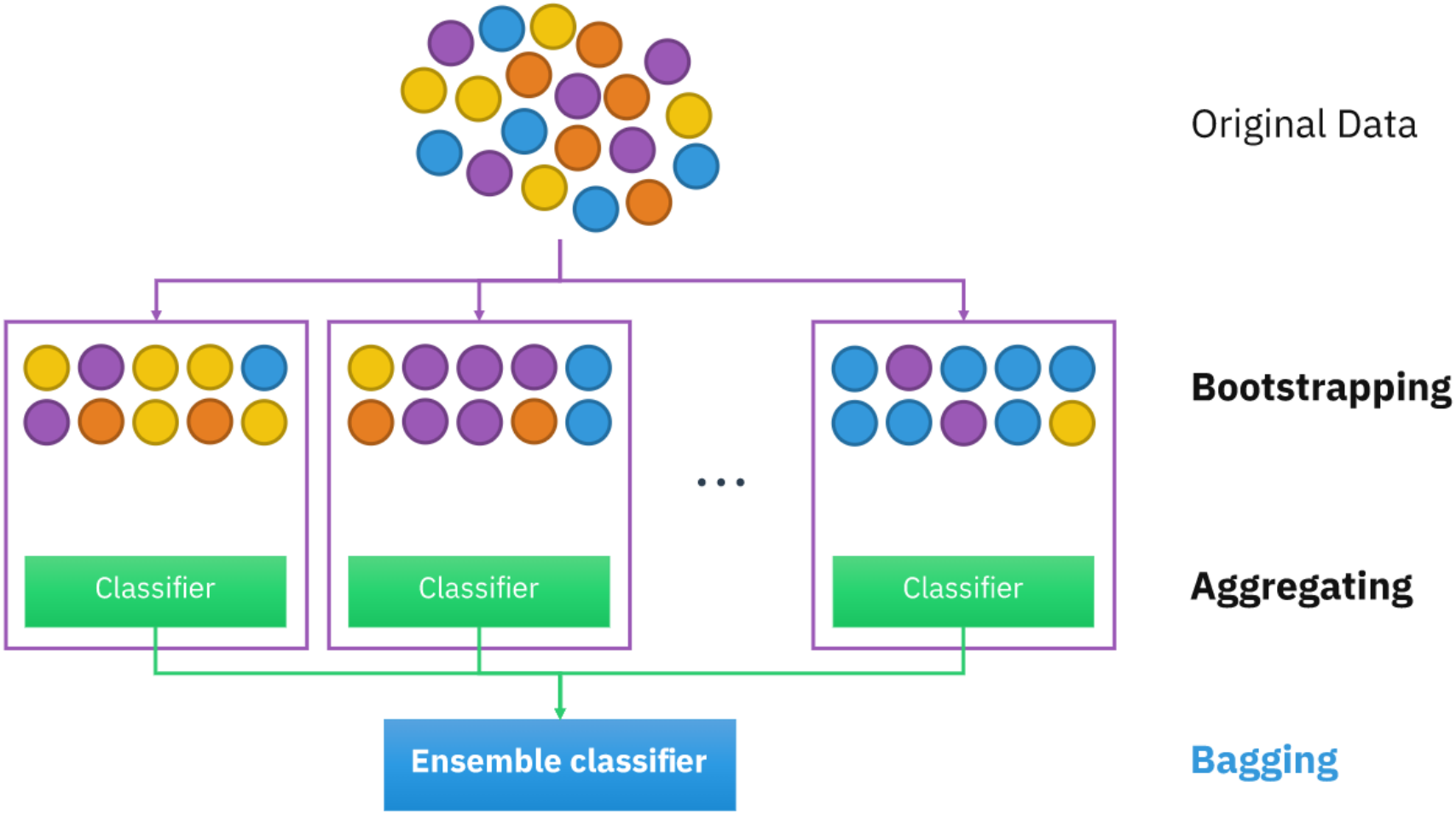
- Easily handle categorical features without preprocessing (e.g., One-hot encoding)
- Easily handle missing values
- Individual decision trees generally often inaccuracy
  - combine multiple trees together into prediction models called ensembles

To prevent overfitting

- Early stopping
  - Restrict the tree depth to a certain level
  - Restricting the minimum number of observations allowed in any terminal node
- Pruning
  - $min(SSE + \alpha|T|)$, T = the number of terminal nodes of the tree

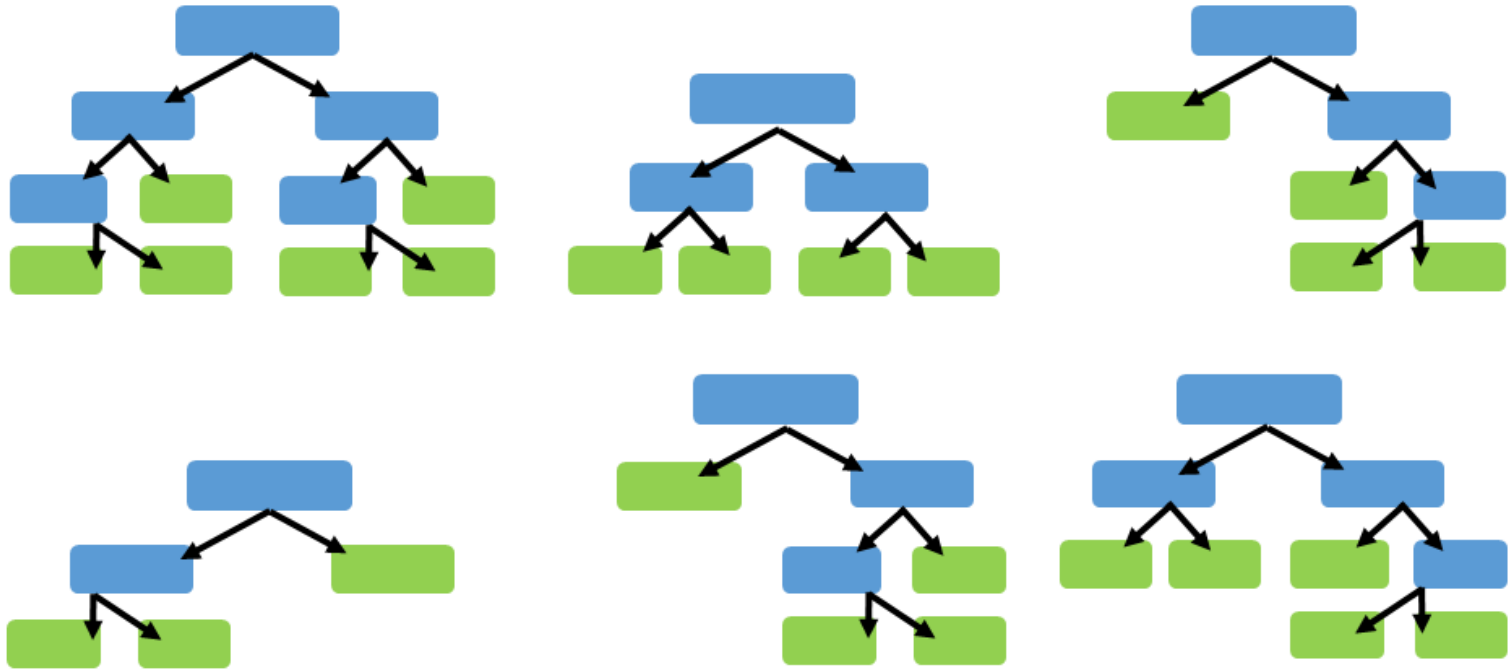
What is Random forest (RF)

- Bootstrap aggregating (Bagging) + Split-variable randomization



https://en.wikipedia.org/wiki/Bootstrap_aggregating

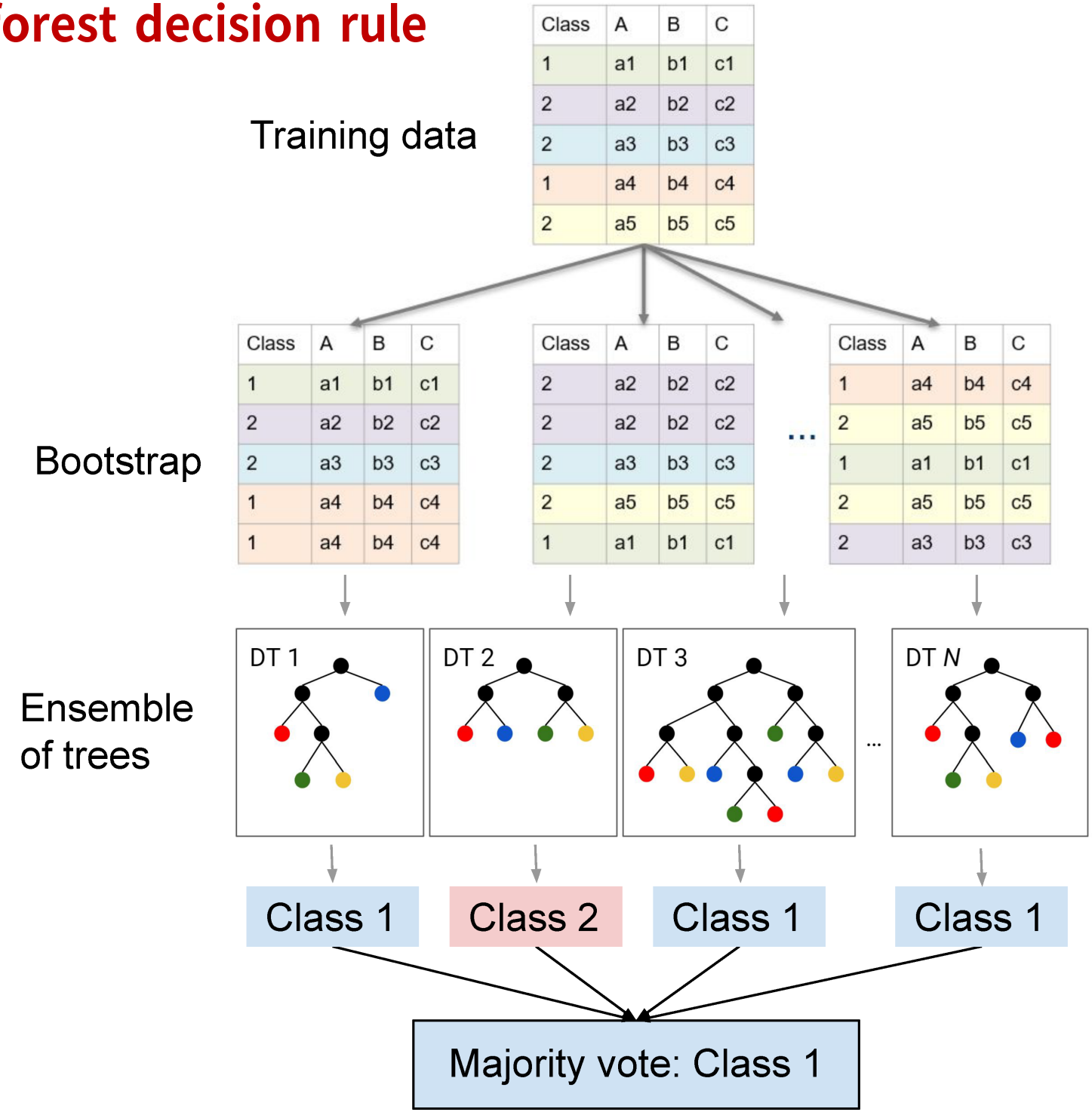
Illustration of RF



https://bradleyboehmke.github.io/HOML

1. Bootstrapped 데이터셋 만들기
2. Random subset of variables 선택
3. Decision tree 생성
4. 위 과정을 반복

Random forest decision rule



The number of trees vs RMSE

- The number of trees needs to be sufficiently large to stabilize the error rate
- A good rule of thumb is to start with 10 times the number of features

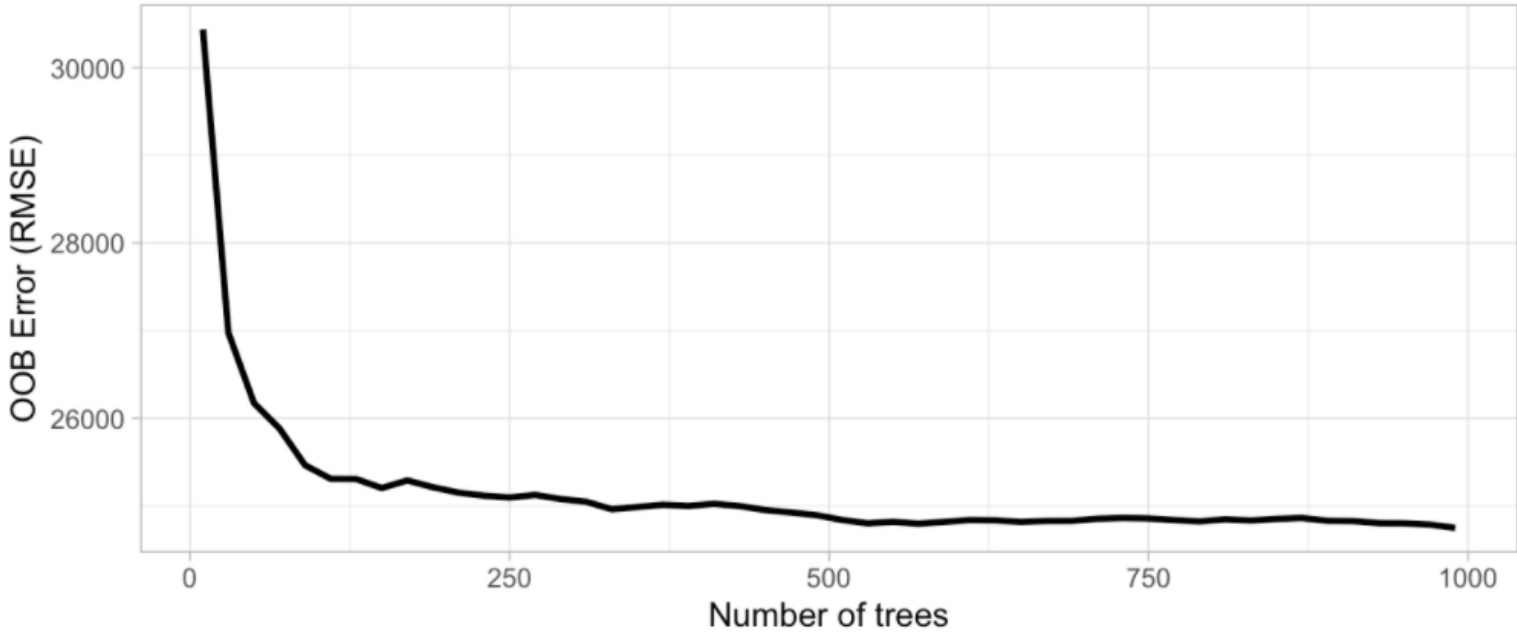


Figure 11.1: The Ames data has 80 features and starting with 10 times the number of features typically ensures the error estimate converges.

https://bradleyboehmke.github.io/HOML

Gradient boosting machine (GBM) algorithm

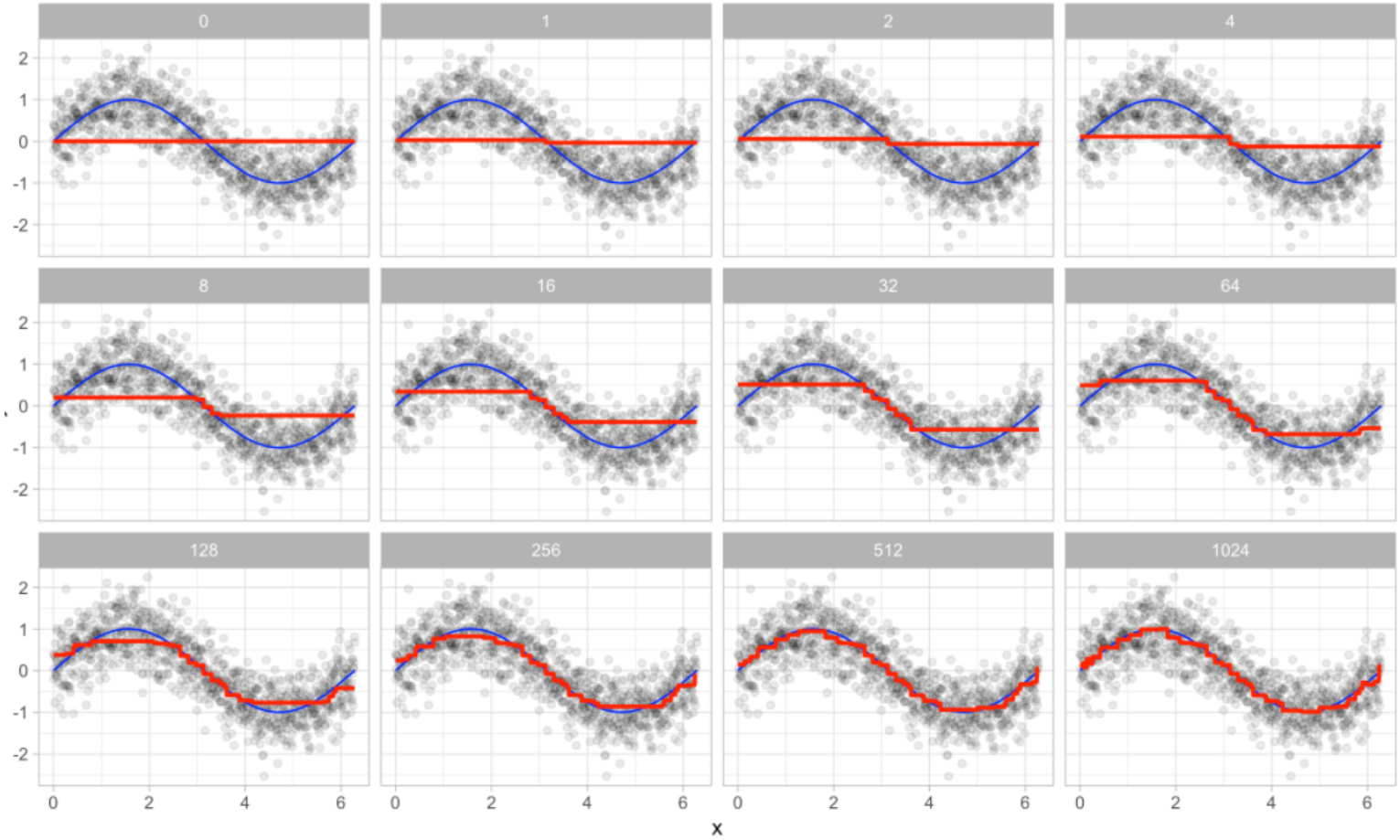
1. Fit a decision tree to the data: $F_1(x) = y$,
2. We then fit the next decision tree to the residuals of the previous: $h_1(x) = y - F_1(x)$,
3. Add this new tree to our algorithm: $F_2(x) = F_1(x) + h_1(x)$,
4. Fit the next decision tree to the residuals of F_2: $h_2(x) = y - F_2(x)$,
5. Add this new tree to our algorithm: $F_3(x) = F_2(x) + h_2(x)$,
6. Continue this process until some mechanism (i.e. cross validation) tells us to stop.

The final model here is a stagewise additive model of *b* individual trees:

$$f(x) = \sum_{b=1}^{B} f^b(x)$$

https://bradleyboehmke.github.io/HOML

Gradient boosting



https://bradleyboehmke.github.io/HOML