Introduction to Data Mining Lecture

Week 12: scikit-learn Example 1

Joon Young Kim

Assistant Professor, School of AI Convergence
Sungshin Women's University

Overview of Multi-Linear Regression

- Multiple linear regression model

  → 가장 기본적인 예측 모델 (The most popular model)

  → 결과와 예측치 관계도:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_p X_p + \epsilon$$

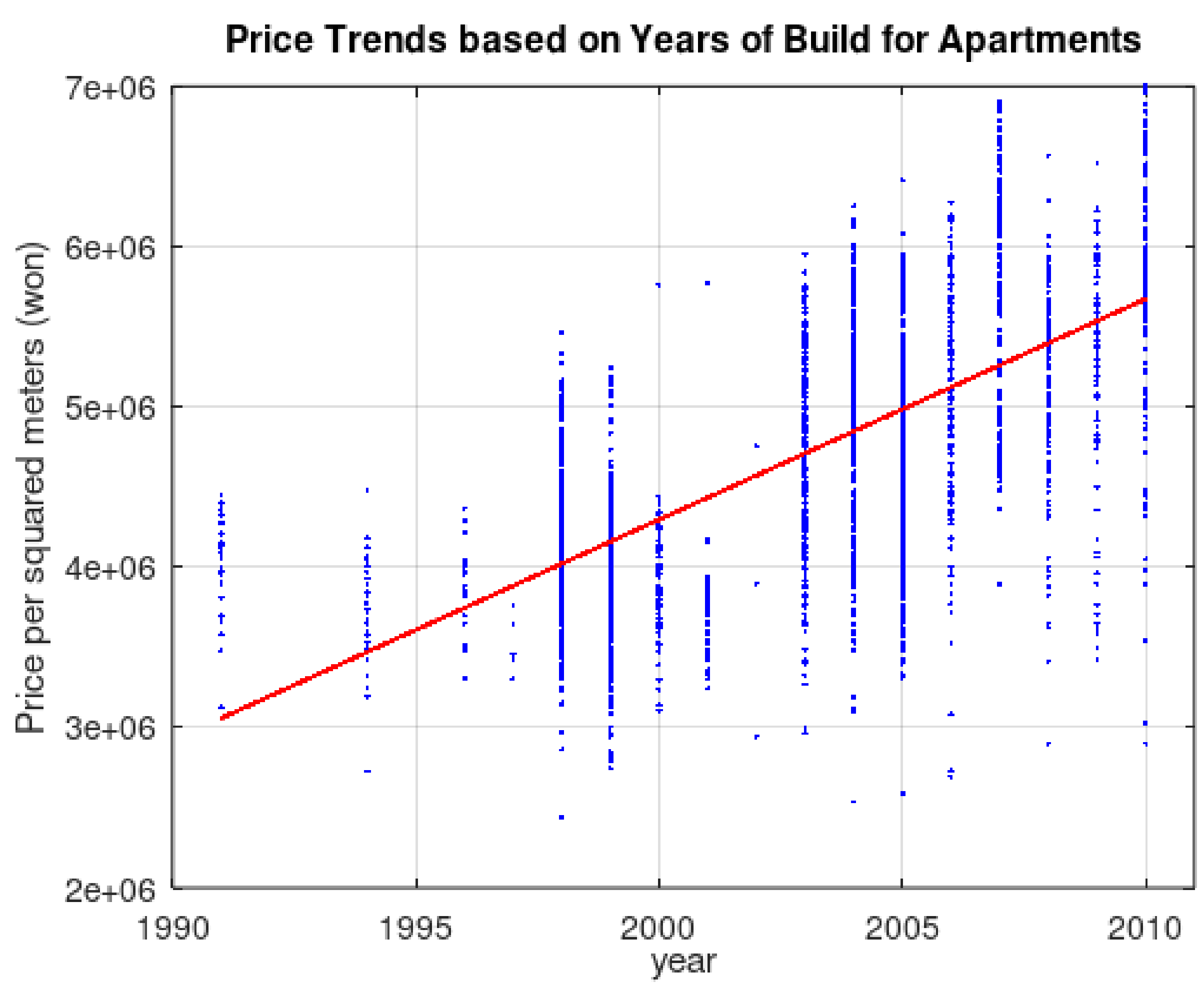
$$\begin{aligned} Y \,&: \text{Quantitative dependent variable} \\ &\quad \text{(The outcome/response variable)} \\ X_1, \cdots, X_p \,&: \text{A set of predictors} \\ &\quad \text{(Independent variables, input variables, regressors, or covariates)} \\ \beta_o, \cdots, \beta_p \,&: \text{Coefficients} \\ \epsilon \,&: \text{Noise} \end{aligned}$$

Overview of Multi-Linear Regression

- Multiple linear regression model

  → 가장 기본적인 예측 모델 (The most popular model)

  → 2011년도 서울시 부동산 거래가



$$Y = \beta_0 + \beta_1 X_1 + \epsilon$$

Y :Price per Squared Meters

X_1 :Year

β_o, β_1 :Coefficients

ϵ :Price Noise

Overview of Multi-Linear Regression

- 예측치 기반의 양적 결과와 연관된 model fitting을 위한 주요 목표는 1) 이해와 2) 예측이다.

  1) 요소들사이의 관계 이해

  Prices per Squared Meters ∝ Year of Build

  2) 새로운 Case의 결과에 대한 예측

  Prices per Squared Meters in 2012?

Overview of Multi-Linear Regression

- 정말 다양한 경우에 Multiple linear regression 적용 가능

인구구조
&
활동 내역

고객 신용카드
사용률

사용률
&
작동 환경

기계 고장
유발 시간

Estimation and Prediction in Regression

- Coefficients(계수)인 $\beta_0, \cdots, \beta_p$ 과 노이즈 표준편차를 통해 데이터의 상관관계를 파악할 수 있다.

- 현재 우리는 샘플만 알고 있다고 가정할때 coefficients인 $\beta_0, \cdots, \beta_p$ 의 값은 unknown이다. 따라서 해당 상관관계를 간략하게 다시 한번 표현하면 하기와 같다.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_p X_p + \epsilon$$

Estimation and Prediction in Regression

- 데이터 예측을 위해서 ordinary least squares (OLS) 기법을 사용한다.

$$
\begin{aligned}
\mathbf{Y} &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_p X_p \\
&= \mathbf{X}\beta
\end{aligned}
$$

$$\rightarrow \mathbf{X^T}\hat{\mathbf{Y}} = \mathbf{X^T}\mathbf{X}\hat{\beta}$$

$$\rightarrow \hat{\beta} = (\mathbf{X^T}\mathbf{X})^{-\mathbf{1}}\mathbf{X^T}\hat{\mathbf{Y}}$$

- 예측값인 $\hat{Y}$ 의 경우 아래와 같이 기술이 가능하다.

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \hat{\beta}_3 X_3 + \cdots + \hat{\beta}_p X_p + \epsilon$$

- 각각의 값들이 비편향적 (unbiased)하다고 고려할시 OLS에 기반한 예측은 가장 좋은 예측이 된다.

Estimation and Prediction in Regression

- 하기와 같은 가정을 기반으로 한다며 이 Regression 기법은 다른 예측치들과 비교할때 가장 적은 평균 제곱 에러를 도출하게 된다.
  - → The noise (혹은 equivalently, dependent variable) 분포는 normal distribution이다.
  - → 선형관계가 정확하다.
  - → 각각의 cases는 독립적이다.
  - → 주어진 예측셋을 위한 Y 값의 변동성은 예측치들의 값에 상관없이 동일하다. (homoskedasticity).

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \hat{\beta}_3 X_3 + \cdots + \hat{\beta}_p X_p + \epsilon$$

$$\begin{array}{c} (x_0[0], x_1[0], \cdots, x_p[0]) \\ (x_0[1], x_1[1], \cdots, x_p[1]) \\ \vdots \end{array}$$

Estimation and Prediction in Regression

- An important and interesting fact for the predictive goal is that **even if we drop the first assumption and allow the noise to follow an arbitrary distribution, these estimates are very good for prediction.**

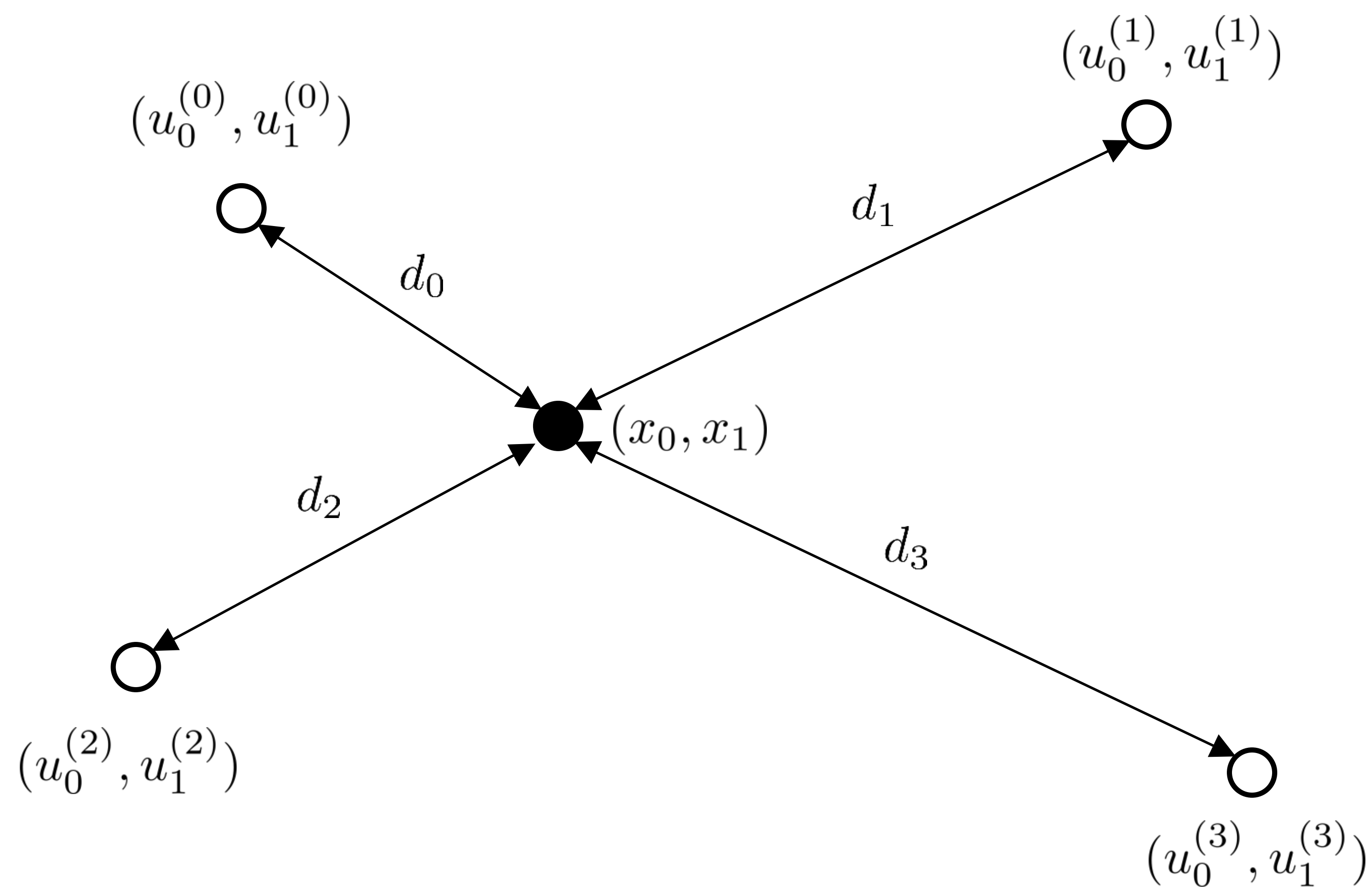
$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_p X_p + \epsilon \quad \not\Rightarrow \quad \epsilon \sim \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Linear Regression Code

- lr.py on LMS

Overview of k-NN Algorithms

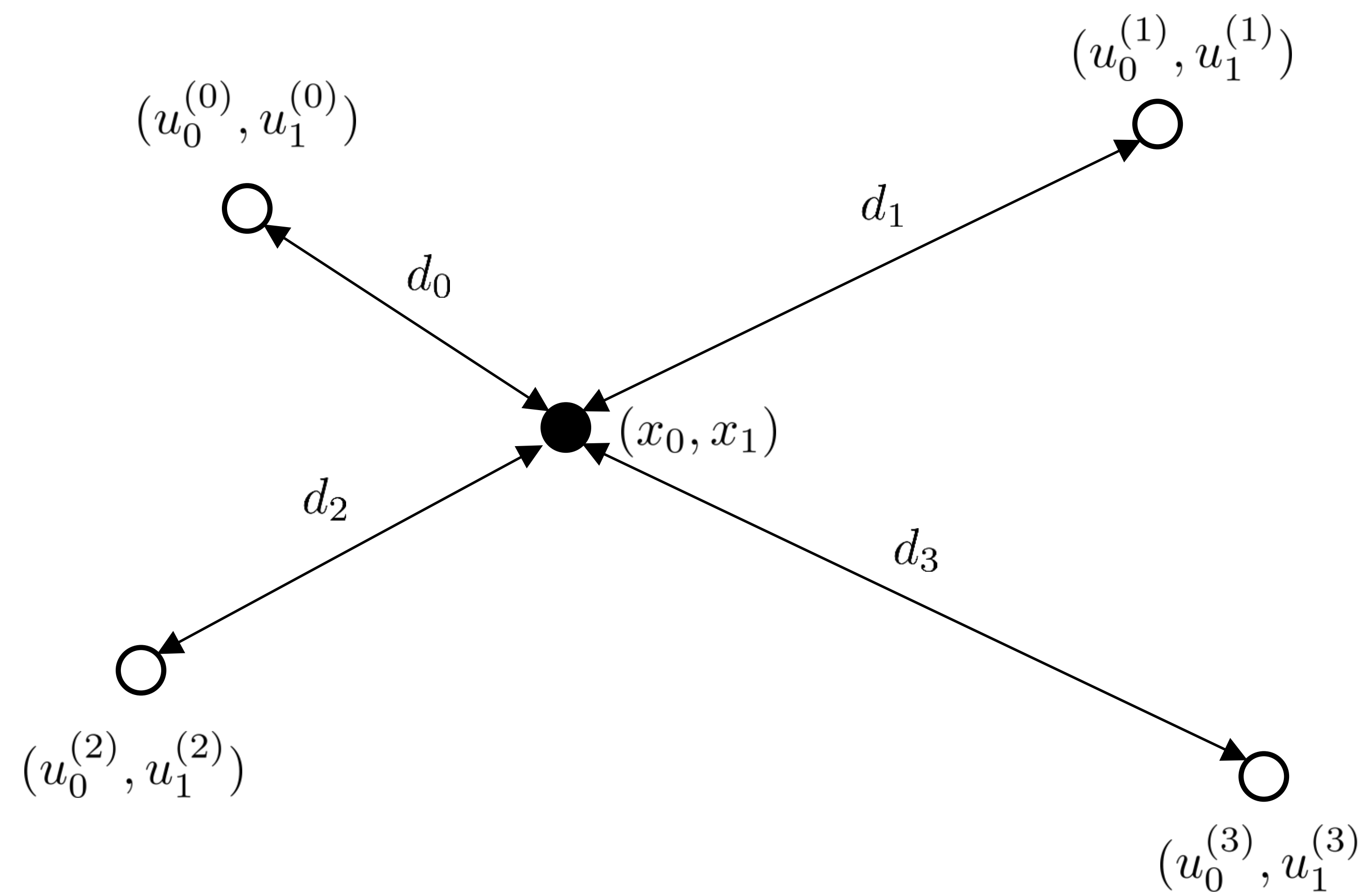
- k-nearest neighbor 기법
  - → 분류를 하고자 하는 새로운 record가 진입시 훈련데이터셋내에서 k 개수의 records를 Identify하는 것



Overview of k-NN Algorithms

- Neighbors 결정하기
  - → Class Membership: Y
  - → Predictors: $u_1, u_2, \cdots, u_p$
  - → Eculidean distances 측정

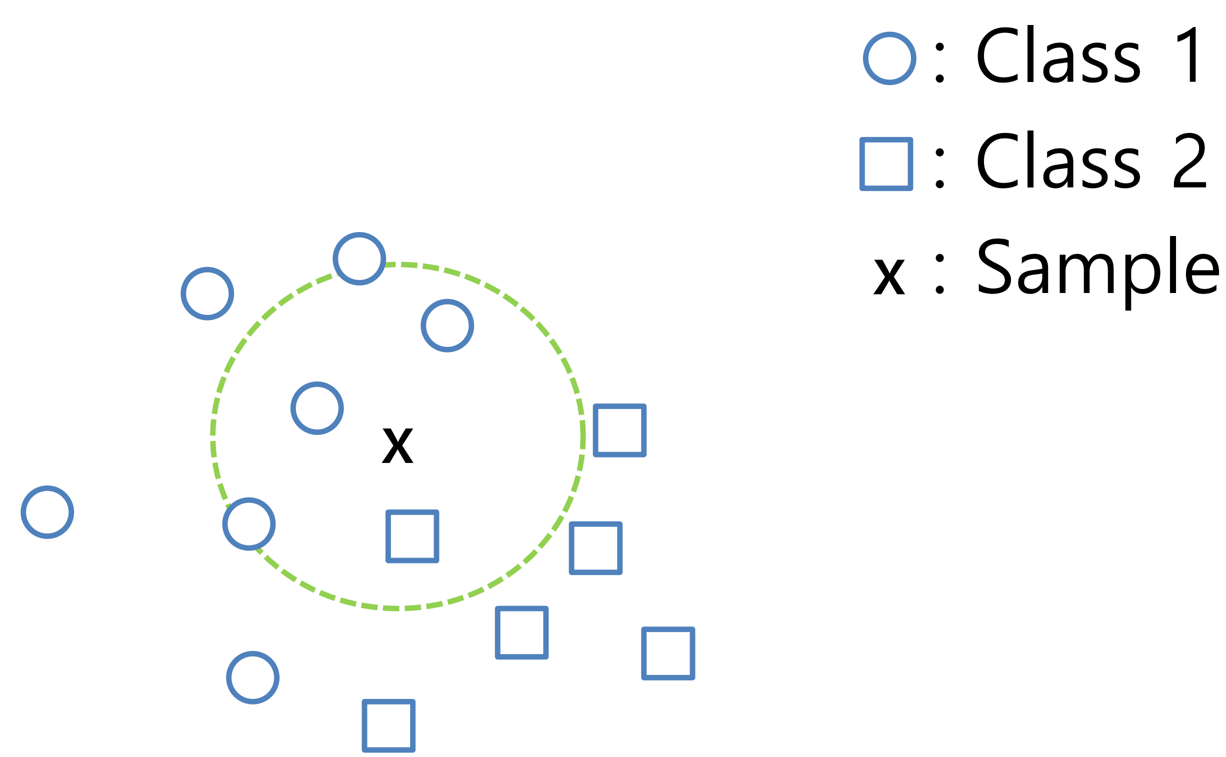
$$d_n = \sqrt{(x_0 - u_0^{(n)})^2 + (x_1 - u_1^{(n)})^2}$$



Overview of k-NN Algorithms

■ Classification Rule

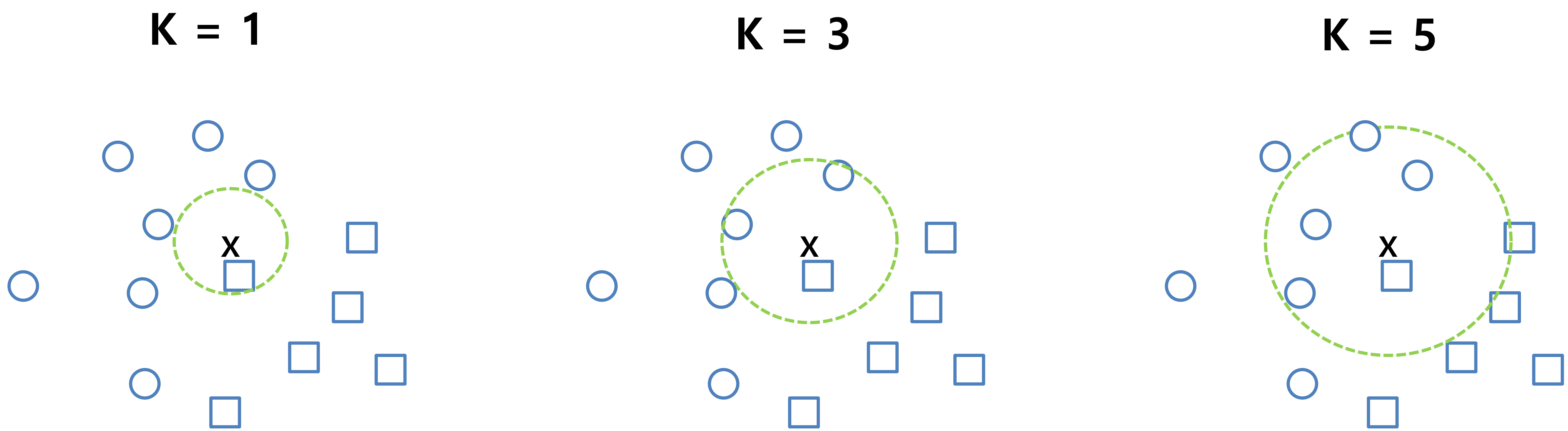
1. 분류하고자 하는 기록에 가장 가까운 k개의 neighbors를 측정한다.
2. 특정 majority class에 분류하는 majority decision rule를 활용하여서 기록을 분류한다.



Overview of k-NN Algorithms

■ K 값 선택하기

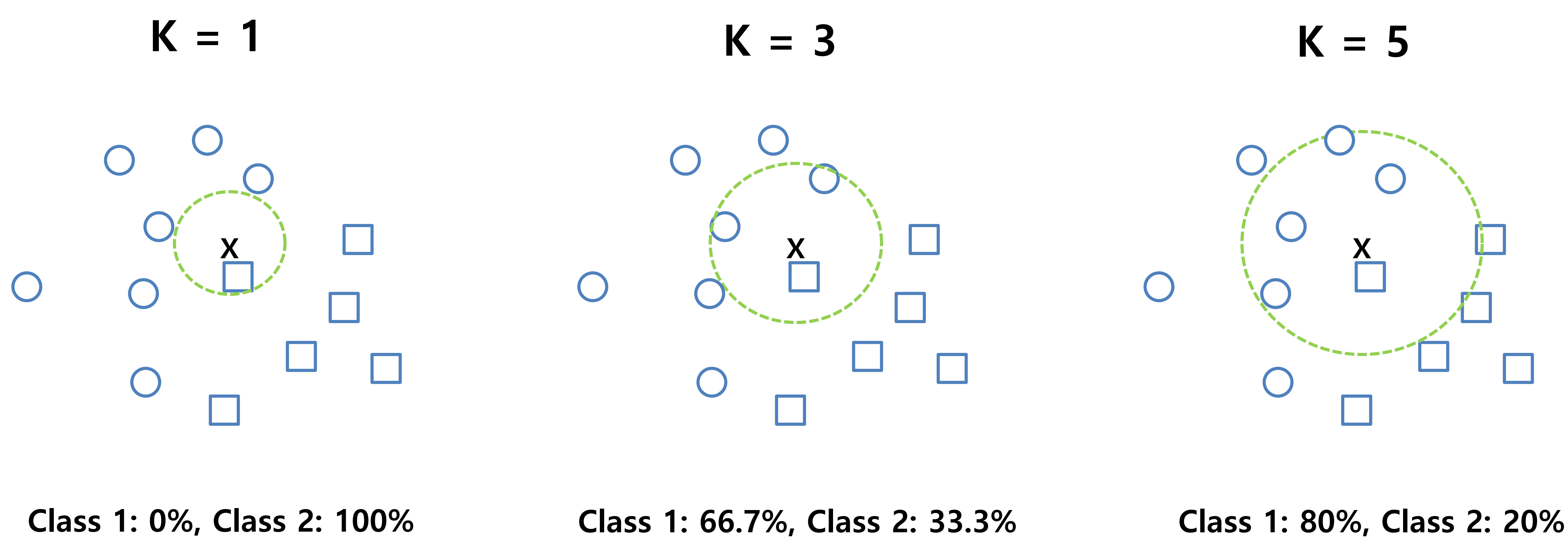
→ K값에 따라, 성능 및 결과는 달라진다.



Overview of k-NN Algorithms

■ Cutoff Value 세팅하기

→ The “majority”의 의미는 class membership probabilities에 기반한 cutoff value와 직접적으로 연관되어 있다.



Overview of k-NN Algorithms

- k-NN with More Than Two Classes
  - → The "majority rule": classified as a member of the majority class of its k neighbors.
  - → use that as an estimate of the probability that the new record belongs to that class, and then refer to a user-specified cutoff value to decide whether to assign the new record to that class.

KNN Code

- 저번 코드인 sk_3.py on LMS

Naive Bayes

- Cutoff Probability Method
  1) 보고자 하는 class 분류를 위해서 cutoff probability를 세팅한다.
  2) 새 record와 같은 전체 훈련 records를 찾는다.
  3) 전체 훈련 records가 본 class에 속할 확률에 대해서 결정한다.
  4) 해당 확률이 cutoff probability보다 높으면 해당 record를 본 class로 지정/분류 한다.

Naive Bayes

- Conditional Probability $P(A|B)$

  → The probability of event A given that event B has occurred

- The Bayesian classifier

  → The only classification or prediction method presented in this book that is especially suited for (and limited to) categorical predictor variables.

$$P(C_i|x_1, \ldots, x_p)$$

$$\text{Class: } C_1, \ldots, C_m$$

$$\text{Predictors: } x_1, \ldots, x_p$$

Bayesian Classifier Design and Operation

■ 현재 발행을 앞두고 있는 회사의 신규 주식

→ 5년후 지수기반 본 주식이 수익을 낼지 손해를 낼지를 분류하고 싶다.

→ 이를 위해서 우리는 수익 확률과 손해 확률에 대한 계산이 필요하다.

→ 수익을 낼 케이스를 C_1, 손해를 낼 케이스를 C_2라고 가정하자.

→ Predictor variabl로써 회사가 IT분야에 속한 경우를 $X = 1$
아닌 경우를 $X = 0$ 로 두었을때 하기와 같은 Pivot Table 생성이 가능하다.

	IT Field	non-IT Field	Total
수익 (Profit)	70	130	200
손해 (Loss)	30	570	600
Total	100	700	800

Bayesian Classifier Design and Operation

■ 현재 발행을 앞두고 있는 회사의 신규 주식

	IT Field	non-IT Field	Total
수익 (Profit)	70	130	200
손해 (Loss)	30	570	600
Total	100	700	800

$$P(\text{Profit},\, C_1) = \frac{200}{800} = \frac{1}{4}$$
$$P(\text{Loss},\, C_2) = \frac{600}{800} = \frac{3}{4}$$
$$P(\text{IT Field},\, X=1) = \frac{100}{800} = \frac{1}{8}$$
$$P(\text{non-IT Field},\, X=0) = \frac{700}{800} = \frac{7}{8}$$

$$P(C_1|X=1) = \frac{70}{100} = \frac{7}{10}$$
$$P(C_2|X=1) = \frac{30}{100} = \frac{3}{10}$$
$$P(C_1|X=0) = \frac{130}{700} = \frac{13}{70}$$
$$P(C_2|X=0) = \frac{570}{700} = \frac{57}{70}$$

Bayesian Classifier Design and Operation

■ Using the “Assign to the Most Probable Class” Method

$\rightarrow$ 대중적인 선택

$\rightarrow$ 가장 확률 높은 Class를 선택

$\rightarrow$ 여기에서는 회사가 IT분야일때 5년내 수익을 낼 확률이 손해 확률보다 높기 때문에 해당 케이스에는 수익으로 분류한다.

$$\boxed{P(C_1|X=1) = \frac{70}{100} = \frac{7}{10}}$$

해당 회사 주식은 5년후 수익 예상

$$P(C_2|X=1) = \frac{30}{100} = \frac{3}{10}$$

Bayesian Classifier Design and Operation

■ Using the Cutoff Probability Method

→ 구체적인 Cutoff Probability을 설정해놓고 이를 기반으로 수익 및 손해 분류를 한다.

→ 손해 분류 Cutoff Probability를 0.6으로 구분할때 회사가 Non-IT 분야일때 5년내 손실에 대한 확률은 81.42%이다.

→ 따라서 해당 회사의 손익 경우 손실로 분류한다.

$$P(C_1|X=0)=\frac{130}{700}=\frac{13}{70}$$

$$\boxed{P(C_2|X=0)=\frac{570}{700}=\frac{57}{70}>0.6}$$

해당 회사 주식은 5년후 손실 예상

Bayesian Classifier Design and Operation

- Complete (Exact) Bayes 절차에 대한 현실적 난제
  - → **Finding all the records in the sample that are exactly like the new record** to be classified in the sense that **the predictor values are all the same**
  - → 작은 샘플의 경우는 괜찮다.(e.g. one predictor).
  - → predictors 수가 증가할수록 문제 (more than 20)
    - ▶ 많은 records 들이 exact matches 가 없다.

Bayesian Classifier Design and Operation

- Solution: Naive Bayes

  → naive Bayes solution에서 match되는 기록을 위한 계산을 할 필요가 없다. 대신 우리는 전체 데이터셋을 활용한다.

  → 수정된 기본 naive Bayes 절차:

  1) class 1를 위해서 각 predictor별 class1에 들어갈 확률을 구한다.
  2) 각각의 확률들을 곱하고 나서 종합적으로 클래스 1에 속할 확률을 곱한다.
  3) 모든 클래스 대상으로 step 1과 2를 반복한다.
  4) Class i에 들어갈 확률을 Step 2에서 구한 값 나누기 전체 클래스 값을 더한것으로 나눈다.
  5) 가장 높은 확률을 보이는 클래스로 record를 지정한다.

Bayesian Classifier Design and Operation

- Naive Bayes Formula
  - → Conditional Probability와 Unconditional Probability를 활용
  - → 각 Predictors들이 Independent할 수록 실제 Complete 일때의 값과 일치해짐

$$P(C_i|x_1,\ldots,x_p) = \frac{P(C_1)[P(x_1|C_1)P(x_2|C_1)\cdots P(x_p|C_1)]}{P(C_1)[P(x_1|C_1)P(x_2|C_1)\cdots P(x_p|C_1)] + \cdots + P(C_m)[P(x_1|C_m)P(x_2|C_m)\cdots P(x_p|C_m)]}$$

$$\text{Class: } C_1,\ldots,C_m$$

$$\text{Predictors: } x_1,\ldots,x_p$$

Bayesian Classifier Design and Operation

- <u>**rare class of special interest**</u>를 위한 절차:
  1) **class of interest를 위한 Cutoff probability를 지정한다.**
  2) class of interest 를 위해서 각 predicto별 해당 class에 들어갈 확률을 구한다.
  3) 각각의 확률들을 곱하고 나서 종합적으로 해당 클래스 에 속할 확률을 곱한다.
  4) 모든 클래스 대상으로 step 2과 3를 반복한다.
  5) Class i에 들어갈 확률을 Step 3에서 구한 값 나누기 전체 클래스 값을 더한 것으로 나눈다.
  6) 이때 확률 값이 cutoff 보다 높으면 해당 기록을 class of interest에 지정한다. 반대로 낮으면 지정하지 않는다.
  7) **필요시 the cutoff value를 모델 파라미터로 취급하면서 조정한다.**

Bayesian Classifier Design and Operation

- Example

Field	Size	Predict
IT	Small	Profit
IT	Large	Loss
Non-IT	Small	Profit
IT	Small	Loss
Non-IT	Large	Loss
IT	Large	Profit
IT	Large	Loss
Non-IT	Small	Loss
IT	Small	Profit
Non-IT	Small	Loss

Bayesian Classifier Design and Operation

Complete (Exact) Bayes Calculations

$$\boxed{P(C_2|X=1,Y=1)=\frac{1}{3}}$$

$$P(C_2|X=0,Y=1)=\frac{2}{3}$$

$$P(C_2|X=1,Y=0)=\frac{2}{3}$$

$$P(C_2|X=0,Y=0)=\frac{1}{1}=1$$

Naive Bayes Calculations

$$\boxed{P(C_2|X=1,Y=1)=}$$

$$\frac{P(X=1|C_2)P(Y=1|C_2)P(C_2)}{P(X=1|C_1)P(Y=1|C_1)P(C_1)+P(X=1|C_2)P(Y=1|C_2)P(C_2)}=$$

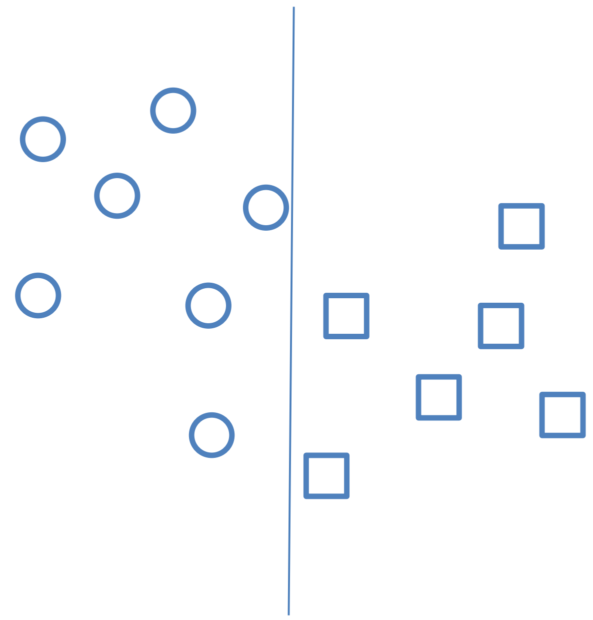
$$\frac{\frac{3}{6}\cdot\frac{3}{6}\cdot\frac{6}{10}}{\frac{3}{4}\cdot\frac{3}{4}\cdot\frac{4}{10}+\frac{3}{6}\cdot\frac{3}{6}\cdot\frac{6}{10}}=\frac{\frac{3}{20}}{\frac{9}{40}+\frac{3}{20}}=\boxed{\frac{2}{5}}$$

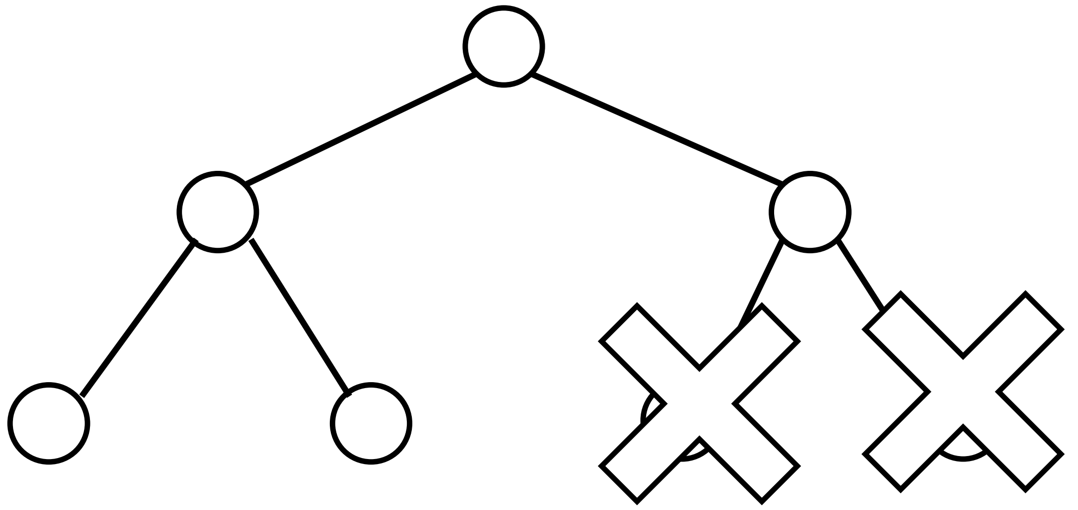
Naive Bayes Code

- nb.py on LMS

Classification Trees

- 쉽게 이해할수 있는 분류 규칙을 포함함
  (easily understandable classification rules)
- 두가지 주요 아이디어
  → Predictor 변수 공간을 회귀적 분리진행 (Recursive partitioning)
  → Validation Data를 이용하여 가지 쳐내기 (Pruning)





Classification Trees

■ Recursive Partitioning

→ p 차원 공간의 파티셔닝(분리)을 통한 각 공간별 하나의 Class 분류

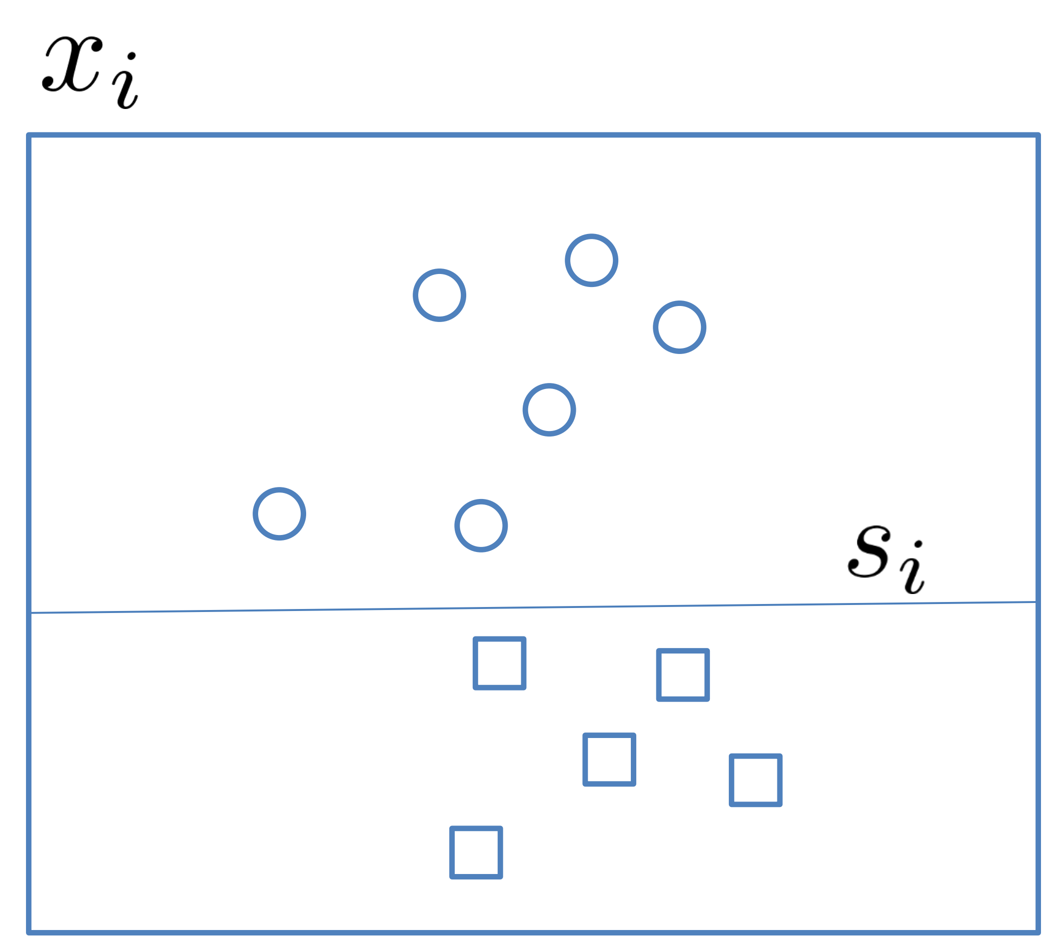
→ 각 predicitor별로 값을 정해서 파티셔닝 진행 가능

$$[x_1, x_2, x_3, \cdots, x_p]$$

$$x_i \leq s_i \qquad\qquad x_i \geq s_i$$

$$\Downarrow \qquad\qquad\qquad \Downarrow$$

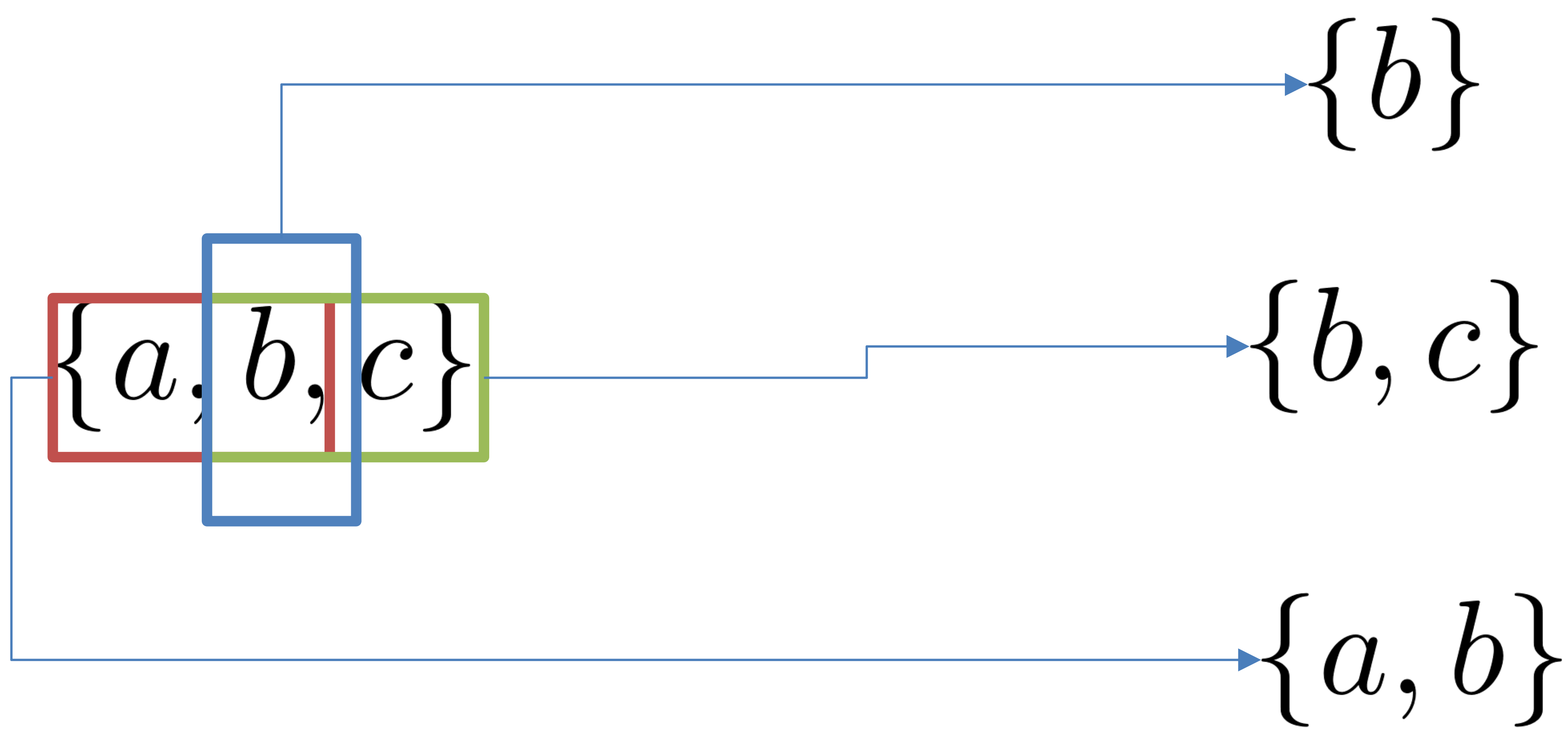
□ ○



Classification Trees

■ Categorical Predictors

→ 해당 predictors의 경우도 소위 파티셔닝이 가능하며 분리 가능



Measurement of Impurity

■ Impurity (오염도)

→ 한마디로 잘못 분류된 경우 (Wrongly classified)

■ Gini impurity index

→ 사각형 A가 있다고 가정할때 해당 A의 Gini Index 공식

$$I(A) = 1 - \sum_{k=1}^{m} p_k^2 \qquad 0 \le I(A) \le \frac{m-1}{m}$$

$k = 1, 2, \ldots, m$

p_k: Proportion of observations to class k

Measurement of Impurity

■ Entropy measure

→ Rectangular A에 대한 엔트로피 계산

$$\text{Entropy}(A) = -\sum_{k=1}^{m} p_k^2 \log_2(p_k)$$

$k = 1, 2, \ldots, m$

p_k: Proportion of observations to class k

$$\Downarrow$$

$$0 \le \text{Entropy}(A) \le \log_2(m)$$

Measurement of Impurity

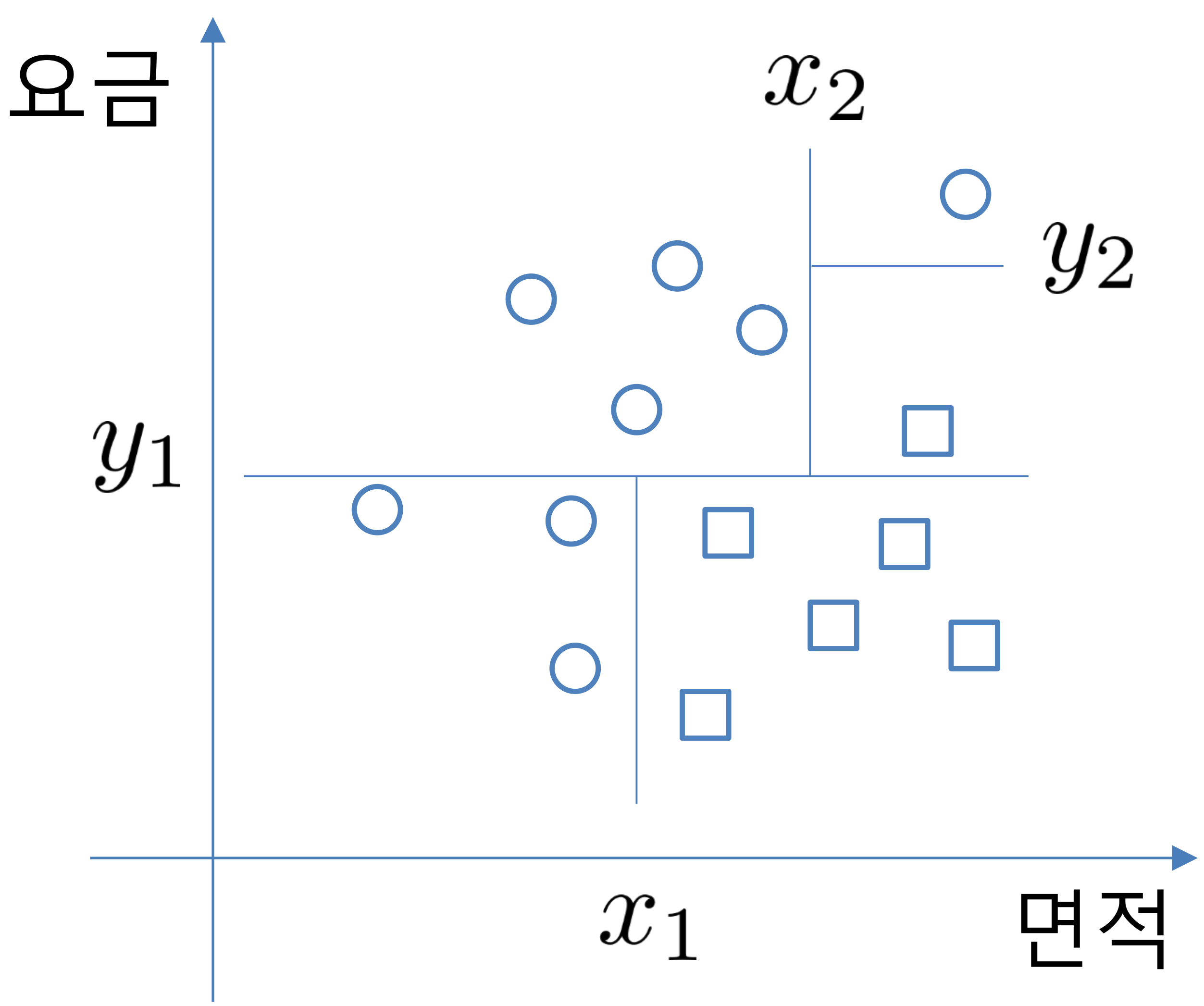
- Example

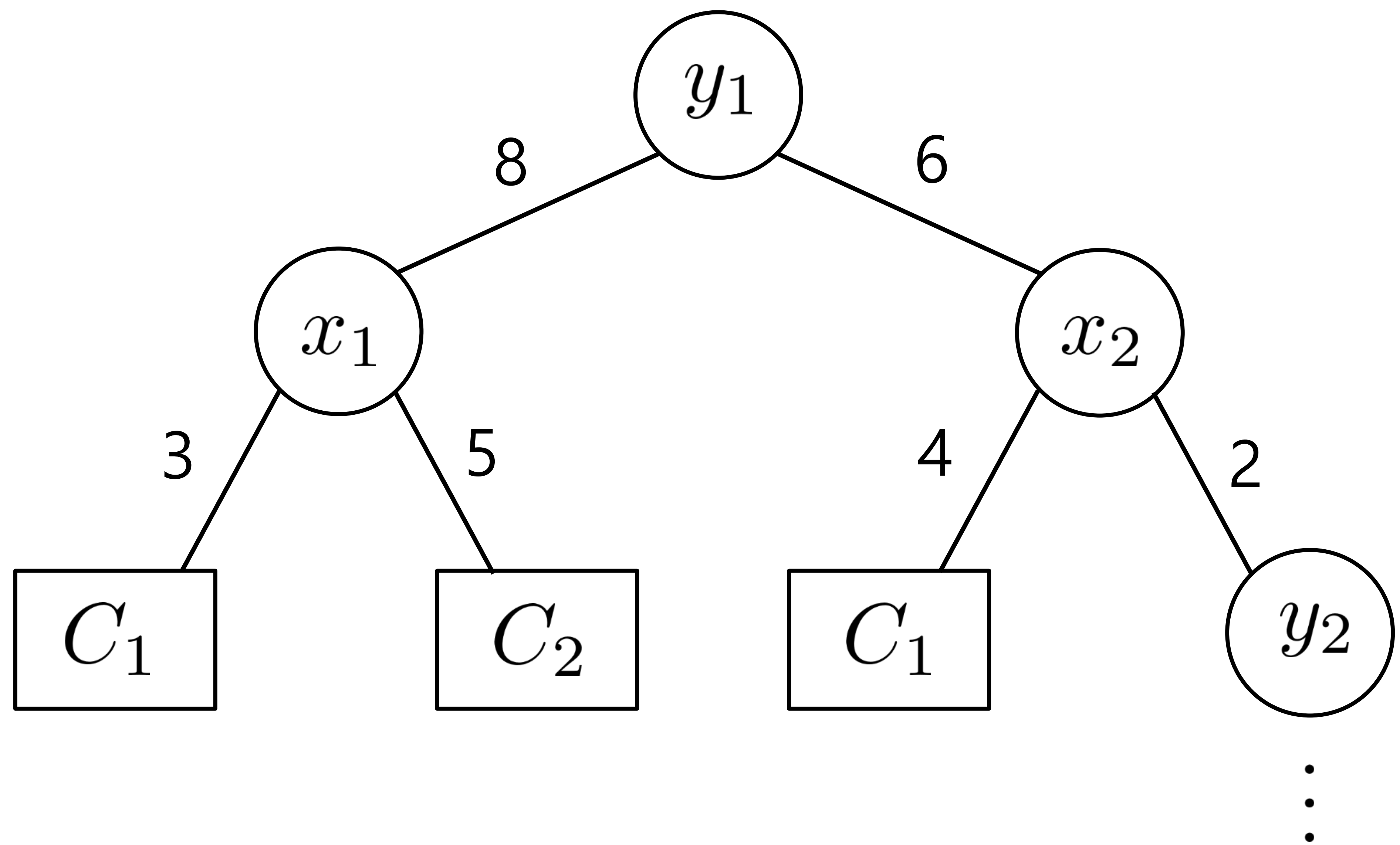
  → Two Class cases with equally distributed.

Measurement of Impurity

■ Tree Structure

→ Split을 통해서 상세화 가능

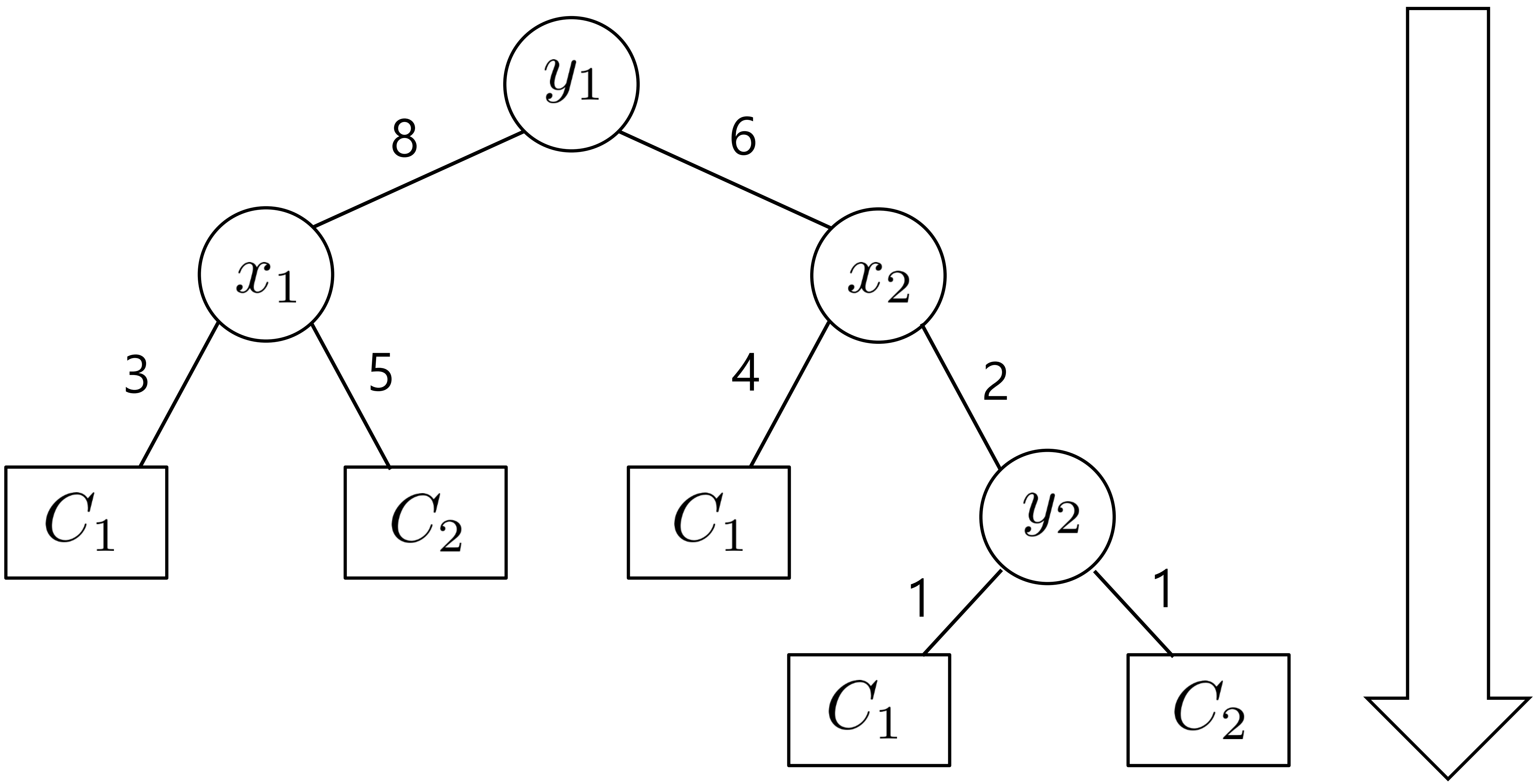




Measurement of Impurity

■ Classifying a New Observation

→ 꼭대기에서부터 밑에까지 내려가면 된다.



Classification Tree Code

- ct.py on LMS