



# Logical Reasoning & Knowledge Based System

---

Practice3

# 실습 1. Python Library

## ■ Numpy

- Rank, Zeros, Ones, Identity, Random
- Array Indexing
- Data type
- Array Math

## Pandas Exercise

### 1. Load Packages

In [1]:

### 2. Import DataSet

In [2]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/10 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/10 8:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/10 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/10 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/10 8:26	3.39	17850.0	United Kingdom

### 3. Create a bar graph with the 10 countries that have the most 'Quantity' ordered except UK

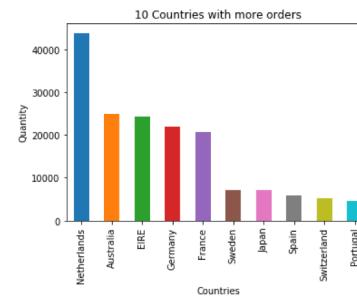
In [3]: # group by the Country

# sort the value and get the first 10 after UK

# create the plot

# Set the title and labels

# show the plot



## ■ Pandas

- Object creation
- Viewing Data
- Import csv dataset
- Selection
- Setting
- Operation
- merge

## ■ Matplotlib

- pyplot

# 실습 2. KNN (scikit-learn)

- Iris data set classification



[setosa]

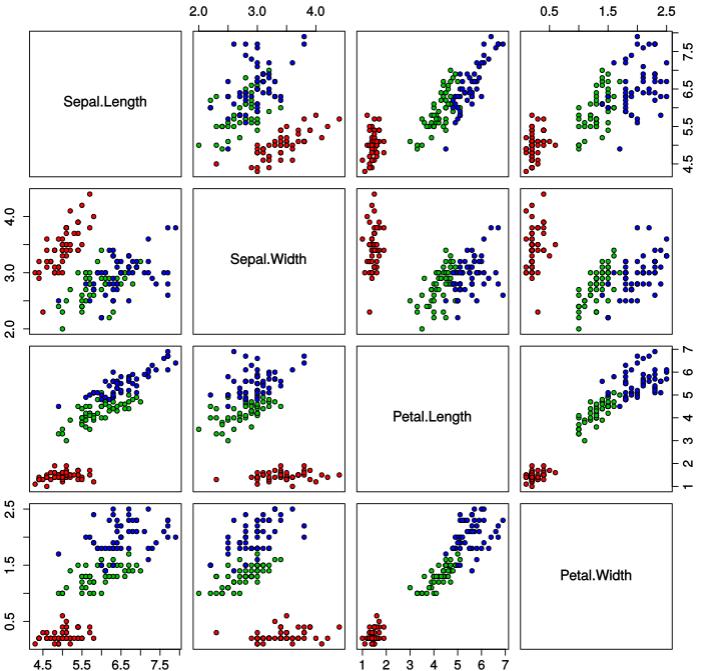


[versicolor]



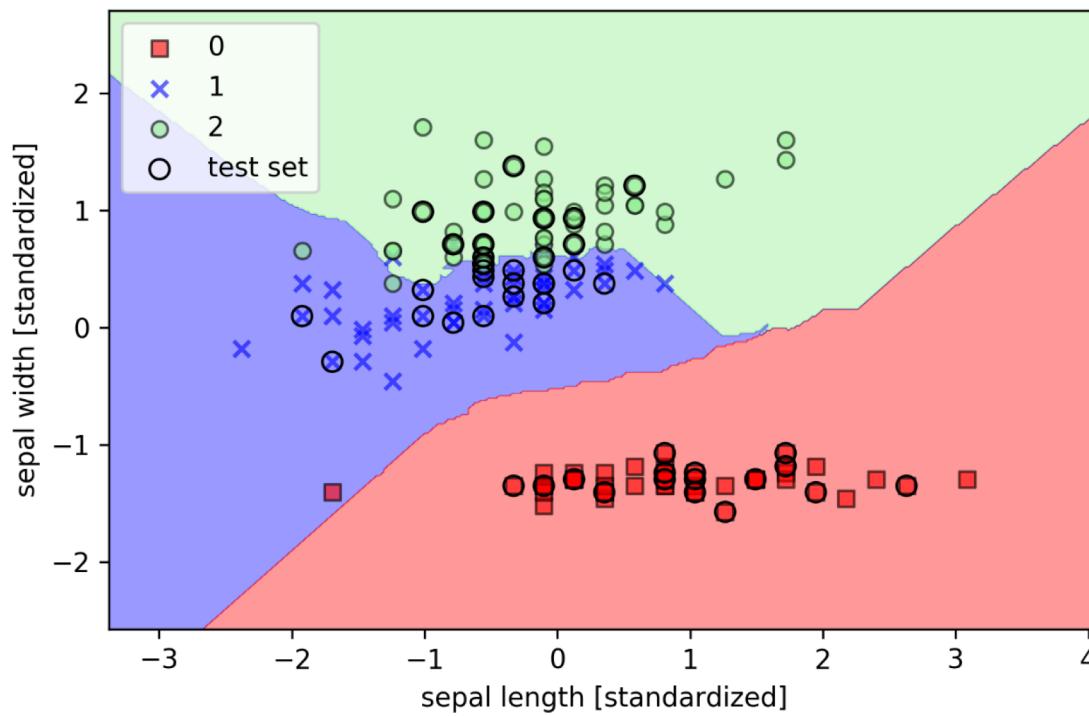
[virginica]

Iris Data (red=setosa,green=versicolor,blue=virginica)



# 실습 2. KNN (scikit-learn)

- Plot Decision Regions



# 실습 3. Weighted KNN Implementation

## ■ k-Nearest Neighbor method

### 1. Define distance functions

#### ■ Example

- $d_{\text{sex}}(A, B) = |A - B|$  (female:0, male:1)
- $d_{\text{age}}(A, B) = |A - B| / \text{max difference}$
- $d = d_{\text{sex}} + d_{\text{age}}$
  
- $d_{\text{sum}}(c5, c1) = |0 - 1| + |20 - 40| / 20 = 2.0$
- $d_{\text{sum}}(c5, c2) = |0 - 1| + |20 - 20| / 20 = 1.0$
- $d_{\text{sum}}(c5, c3) = |0 - 0| + |20 - 30| / 20 = 0.5$
- $d_{\text{sum}}(c5, c4) = |0 - 1| + |20 - 30| / 20 = 1.5$

# 실습 3. Weighted KNN Implementation

## 2. Predict value from neighbors

- Weighted average of neighbor values  $f_i$  ( $Y = +1$ ,  $N = -1$ )

$$\text{Let } w_i = \frac{1}{d_{ij}}, \quad W = \sum w_i$$

$$f_j = \sum f_i \cdot \frac{w_i}{W}$$

- Example

- 3-NN  $\rightarrow c_3, c_2, c_4$

$c_3: f_3 = -1(N), d_{35} = 0.5, w_3 = 2.0$

$c_2: f_2 = +1(Y), d_{25} = 1.0, w_2 = 1.0$

$c_4: f_4 = +1(Y), d_{45} = 1.5, w_4 = 0.67$

- $f_5 = [(-1*2.0) + (1*1.0) + (1*0.67)] / 3.67 = -0.09 \rightarrow N$

# 실습 3. Weighted KNN Implementation

```
def getPredictionsWeightedKNN(trainingSet, testSet, k):
    predictions = []
    # for each instances(vectors) in testSet
    for i in range(len(testSet)):
        # get K neighbors from trainingSet
        neighbors = getNeighbors(trainingSet, testSet[i], k, distance=distance)

        # Weighted KNN
        # get W & # get f_j
        weights = []
        fi_wi = []
        for neighbor in neighbors:
            # set d_ij(distance)
            d_ij = (5)
            label = (6)
            f_i = function_i(label)

            # Let w_i = (1 / d_ij)
            (7)
            # f_i * w_i
            (8)
            # W = Sum([w_i])
            (9)
            # f_j = Sum(fi_wi) / W
            (10)
            # determine label by f_j
            if f_j (11):
                prediction = 0
            elif f_j (12):
                prediction = 1

            # put result in the predictions
            (13)

    return predictions
```

# 실습 4. Propositional Calculus

- 명제논리 추론규칙들을 적용하여 아래 a.의 4 개 문장들로부터 b.의 문장을 얻는 과정을 상세히 기술하시오. 매 단계에 적용한 추론 규칙을 명시하시오.

a.  $A \Rightarrow B \wedge C$

$$D \Rightarrow E \vee \neg B \vee F$$

$$\neg C \vee D$$

$$A$$

b.  $E \vee F$

# 실습 5. Predicate Calculus

- 다음의 문장을 보고 답하시오.

S1. “고열이 있고 저위험군이면 Aspirin을 처방한다”  
S2. “고열이 있고 고위험군이면 Tamiflu를 처방한다“  
S3. “심장병이 있으면 고위험군이다”  
S4. “나이가 많고 고혈압이면 고위험군이다”  
S5. “고위험군이 아니면 저위험군이다”  
S6. “Kim씨는 고열이 있다”  
S7. “Kim씨는 고혈압이 있다”  
S8. “Kim씨는 나이가 많다”

- a. S1 ~ S8을 모두 술어논리 문장으로 나타내시오. 다음과 같은 이름을 쓰시오.

**prescript(X, Y)** (Y: aspirin 또는 tamiflu), **group(X, Y)** (Y: high\_risk 또는 low\_risk),  
**old(X)**, **young(X)**, **high\_bp(X)**, **high\_fever(X)**, **heart\_disease(X)**

- b. S1 ~ S8로부터 forward chaining으로 다음의 문장을 추론하는 과정을 보이시오.

S9: **prescript(kim, tamiflu)**

- c. 다음의 문장을 backward chaining으로 추론하여 처방할 약을 결정하는 과정을 보이시오.

S9: **prescript(kim, X)**