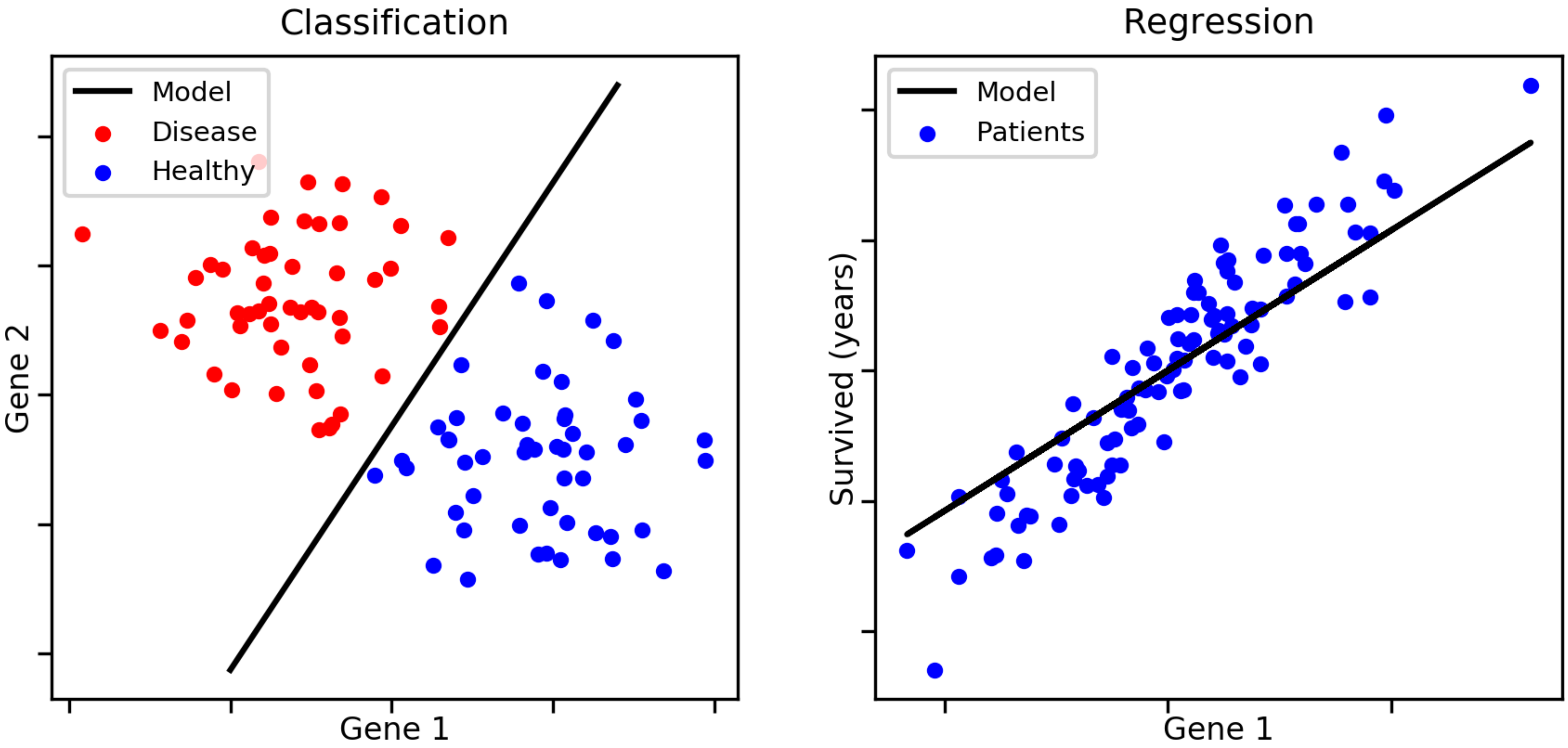
Machine Learning

Lecture 3 Classification

Supervised Learning



Source: https://aldro61.github.io/microbiome-summer-school-2017/sections/basics/

Machine Learning Process

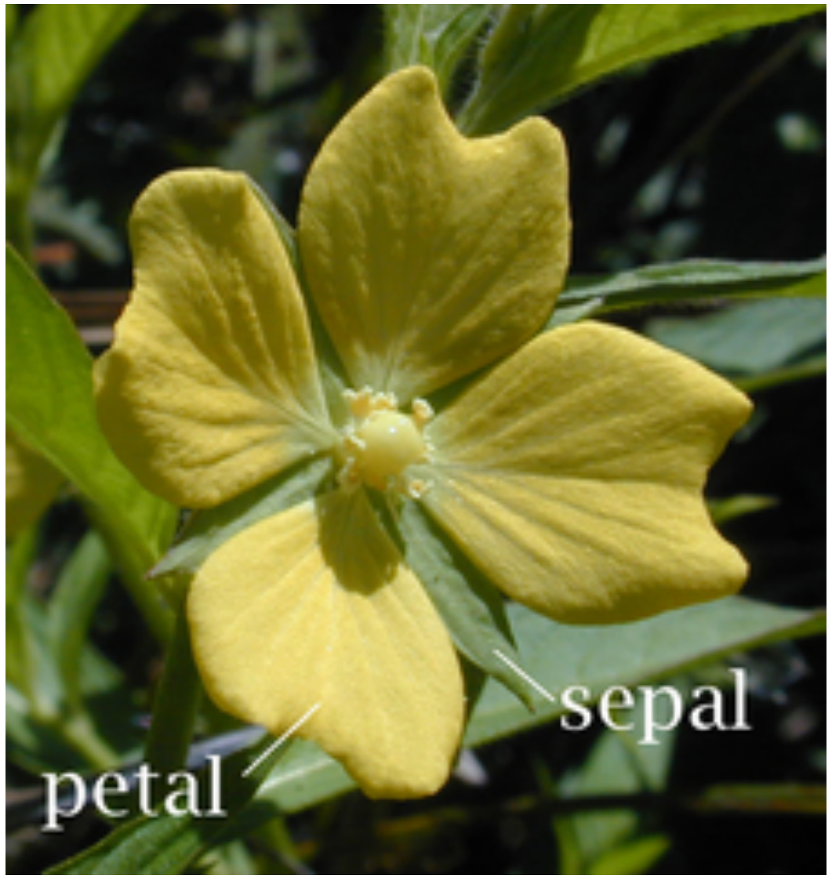
Iris Data Set (鳶尾花卉數據集)

https://archive.ics.uci.edu/ml/datasets/Iris

https://www.kaggle.com/uciml/iris

setosa

- 150個樣本，都屬於鳶尾屬下的三個亞屬，分別是山鳶尾、變色鳶尾和維吉尼亞鳶尾。
  versicolor virginica
- 四個特徵: 花萼和花瓣的長度和寬度
  Sepal Petal



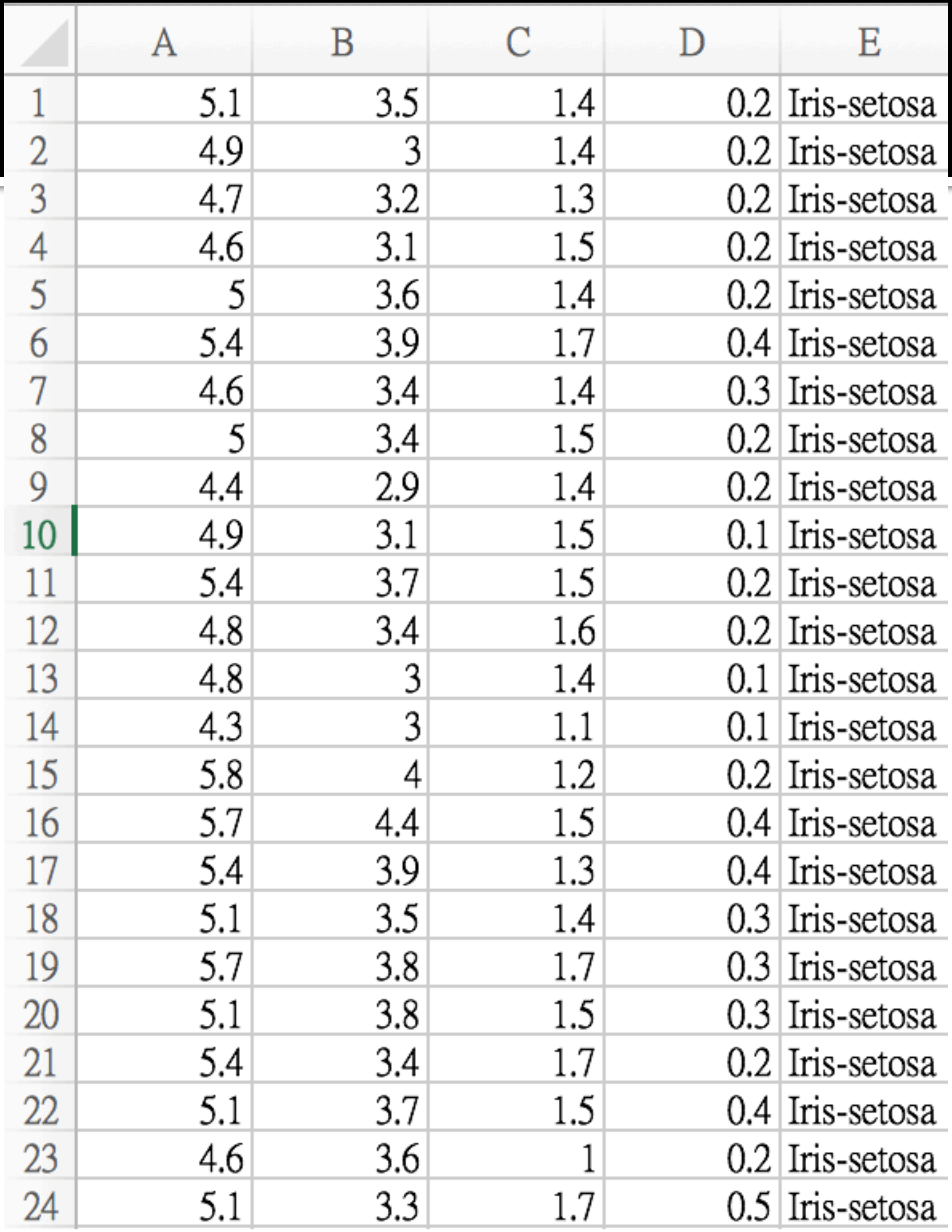
Machine Learning Process

- 準備資料　(包含資料預處理)
- 選擇演算法
- 調整參數
- 評估結果

準備資料

Attribute Information:

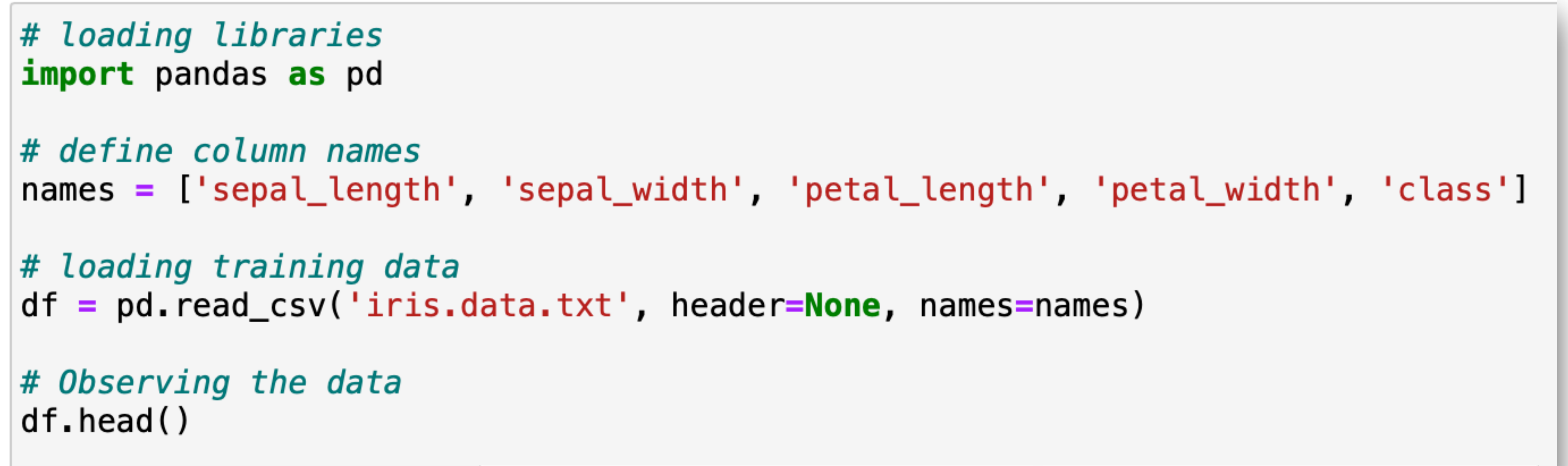
1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:
-- Iris Setosa
-- Iris Versicolour
-- Iris Virginica



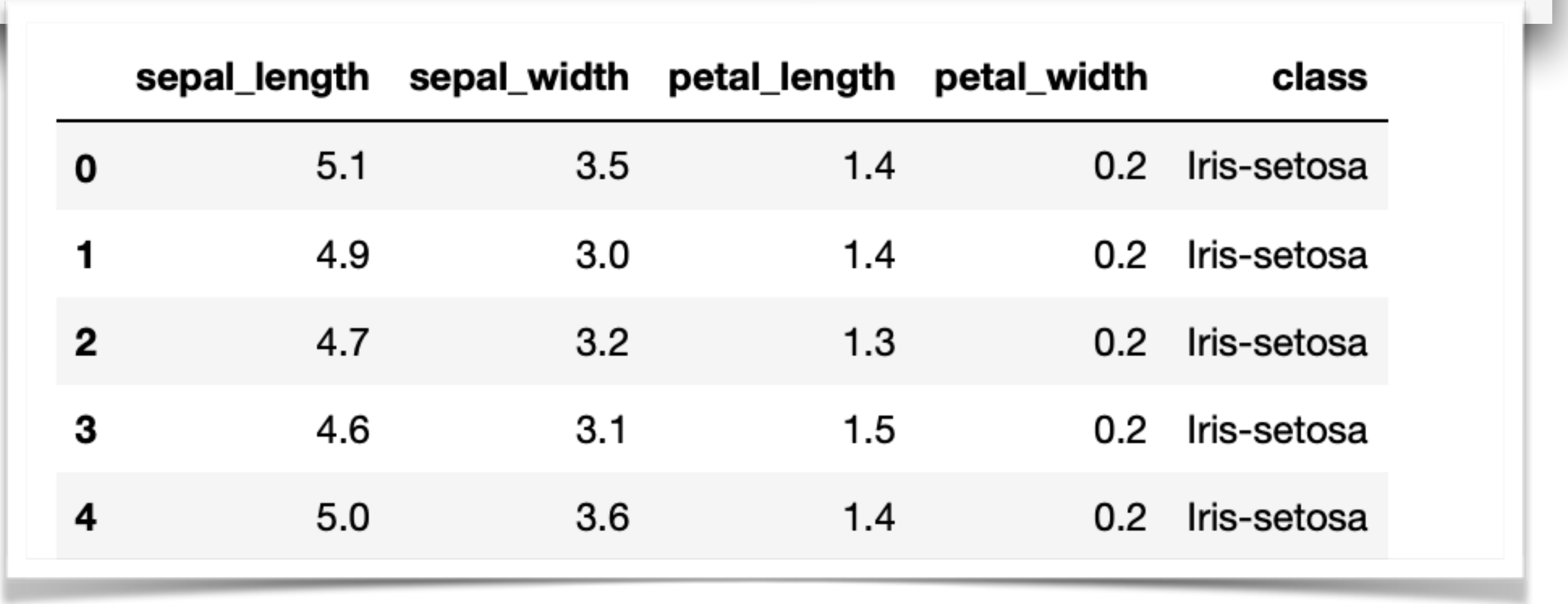
	A	B	C	D	E
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3	1.4	0.1	Iris-setosa
14	4.3	3	1.1	0.1	Iris-setosa
15	5.8	4	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
18	5.1	3.5	1.4	0.3	Iris-setosa
19	5.7	3.8	1.7	0.3	Iris-setosa
20	5.1	3.8	1.5	0.3	Iris-setosa
21	5.4	3.4	1.7	0.2	Iris-setosa
22	5.1	3.7	1.5	0.4	Iris-setosa
23	4.6	3.6	1	0.2	Iris-setosa
24	5.1	3.3	1.7	0.5	Iris-setosa

準備資料

Loading the Data



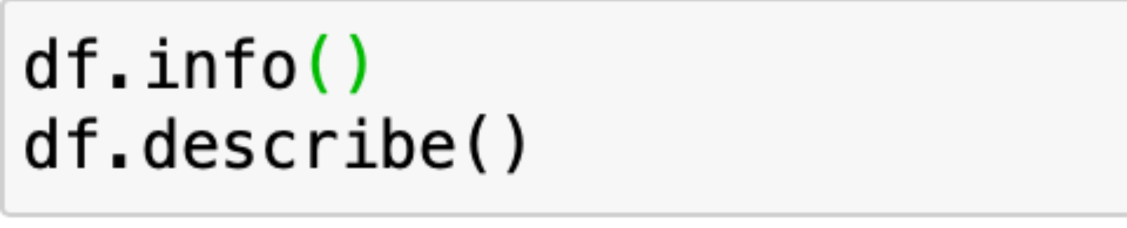
```
# loading libraries
import pandas as pd

# define column names
names = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'class']

# loading training data
df = pd.read_csv('iris.data.txt', header=None, names=names)

# Observing the data
df.head()
```



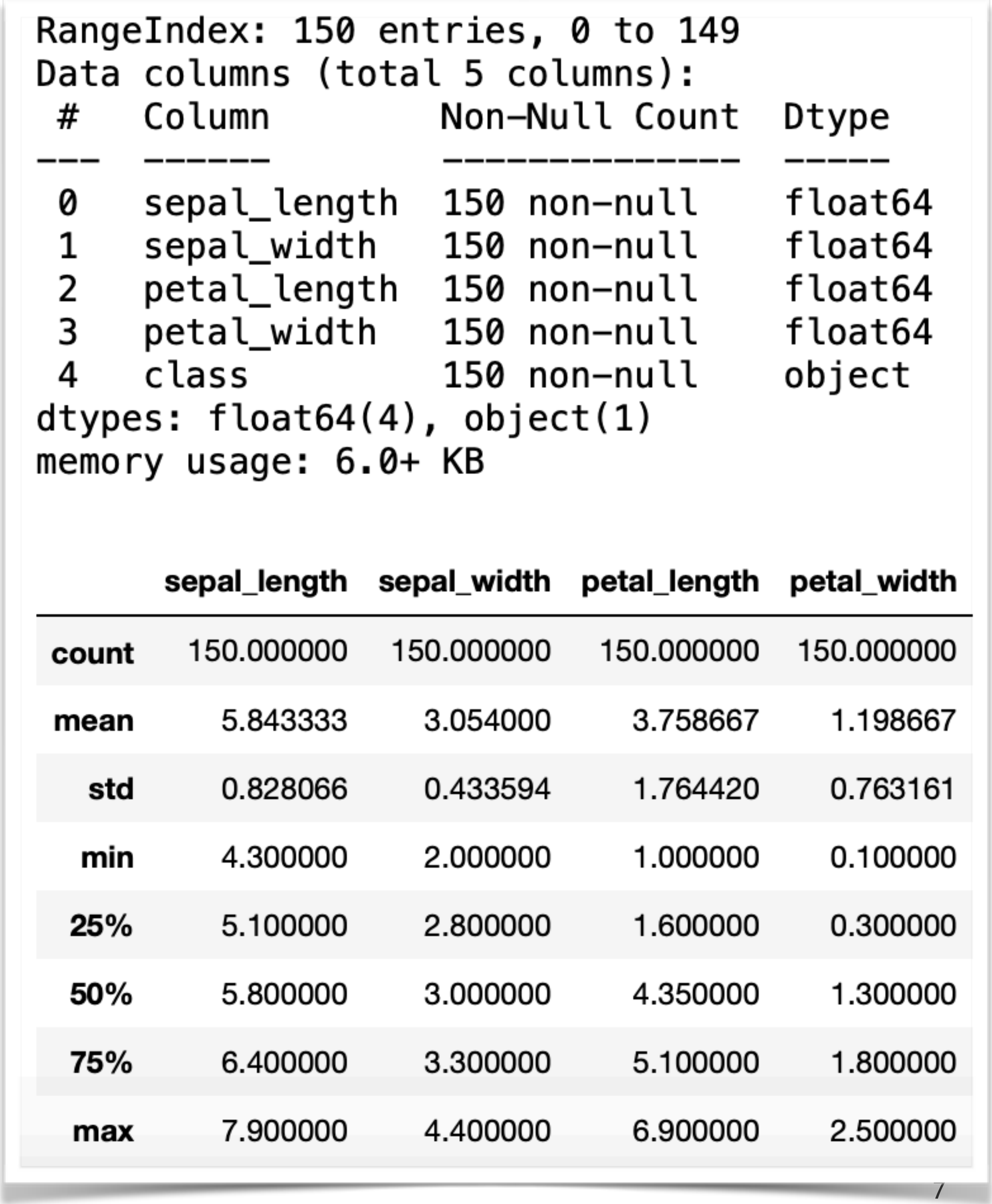
	sepal_length	sepal_width	petal_length	petal_width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

準備資料

- Observing the data



```
df.info()
df.describe()
```

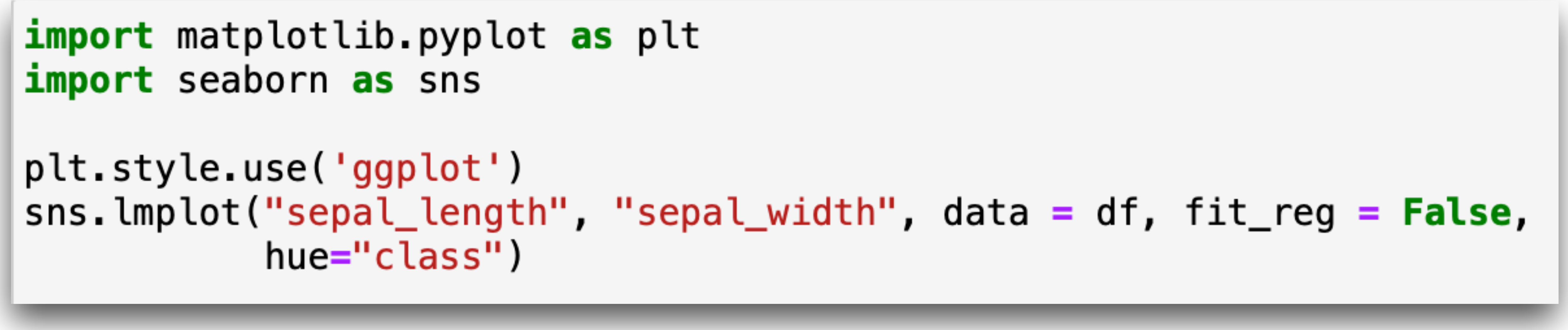


```
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   class         150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

準備資料

- Plotting graph

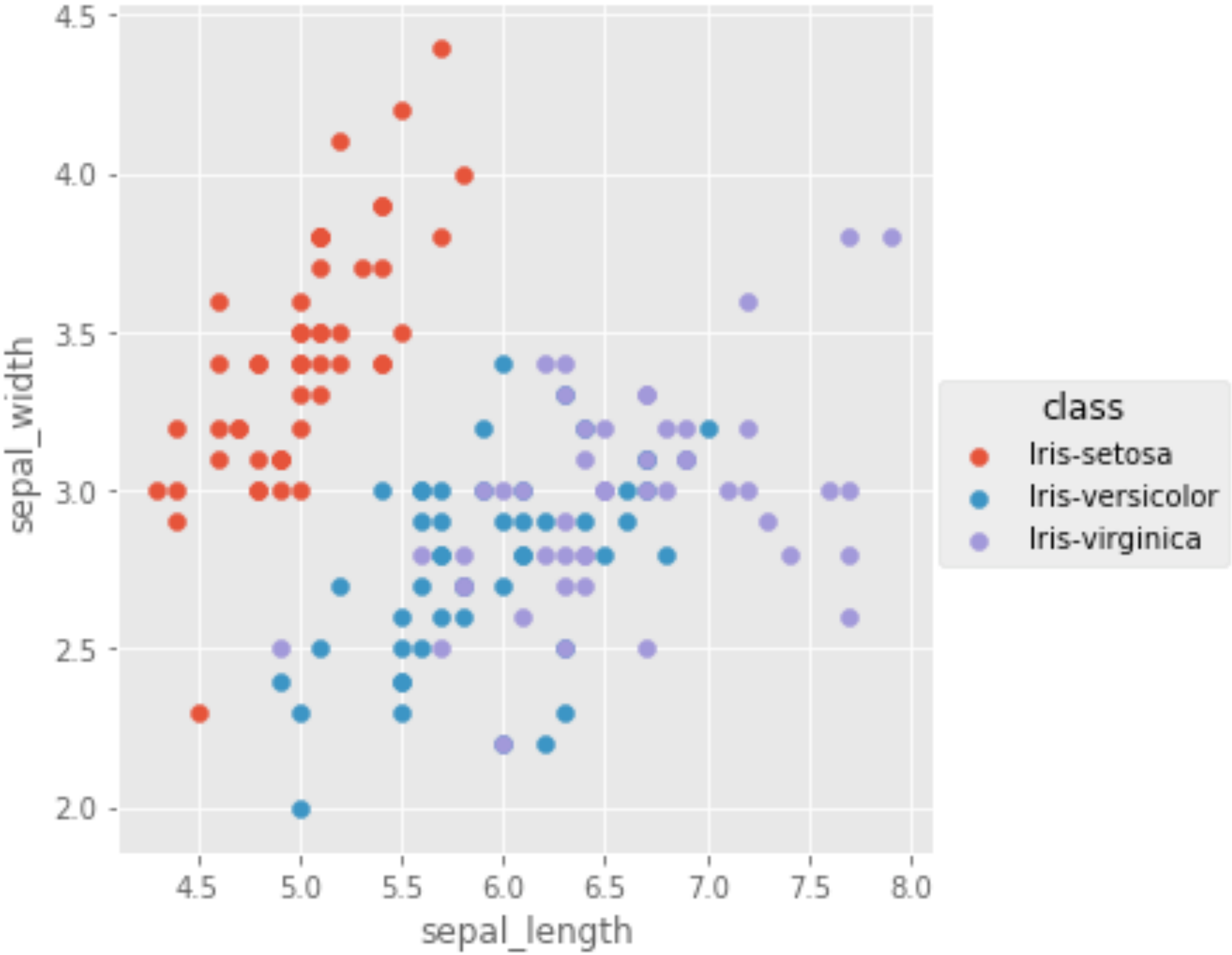


```
import matplotlib.pyplot as plt
import seaborn as sns

plt.style.use('ggplot')
sns.lmplot("sepal_length", "sepal_width", data = df, fit_reg = False,
           hue="class")
```

準備資料

藉由圖形來輔助我們判斷 class 及其他特徵的關係

- Plotting graph - 花萼 長 v.s. 寬



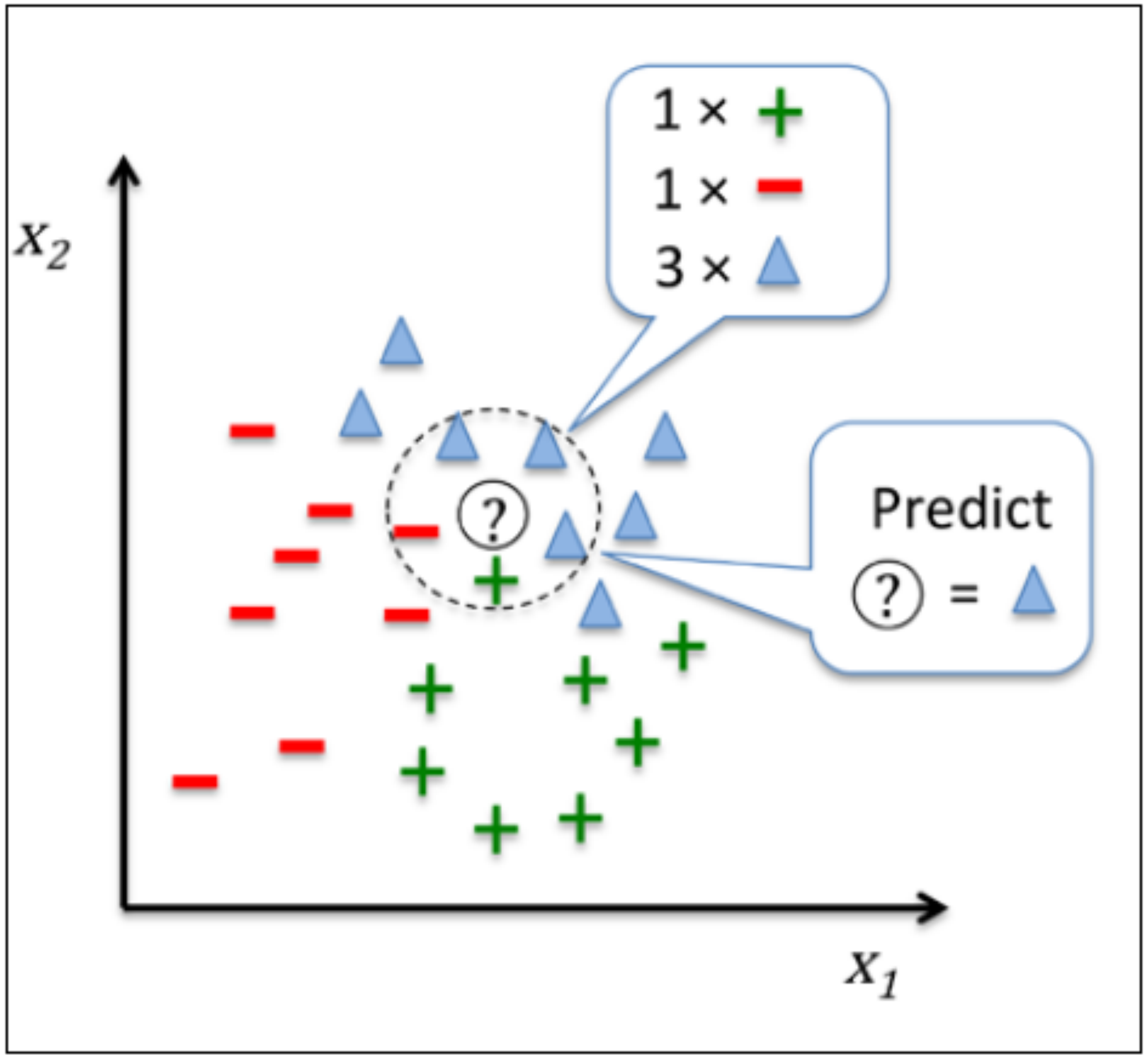
Machine Learning Process

- 準備資料　（包含資料預處理）
- 選擇演算法
- 調整參數
- 評估結果

K-NN (K-nearest neighbor classifier)

$$d(a,b)=\sqrt{(a_1-b_1)^2+(a_2-b_2)^2+\cdots+(a_n-b_n)^2}$$

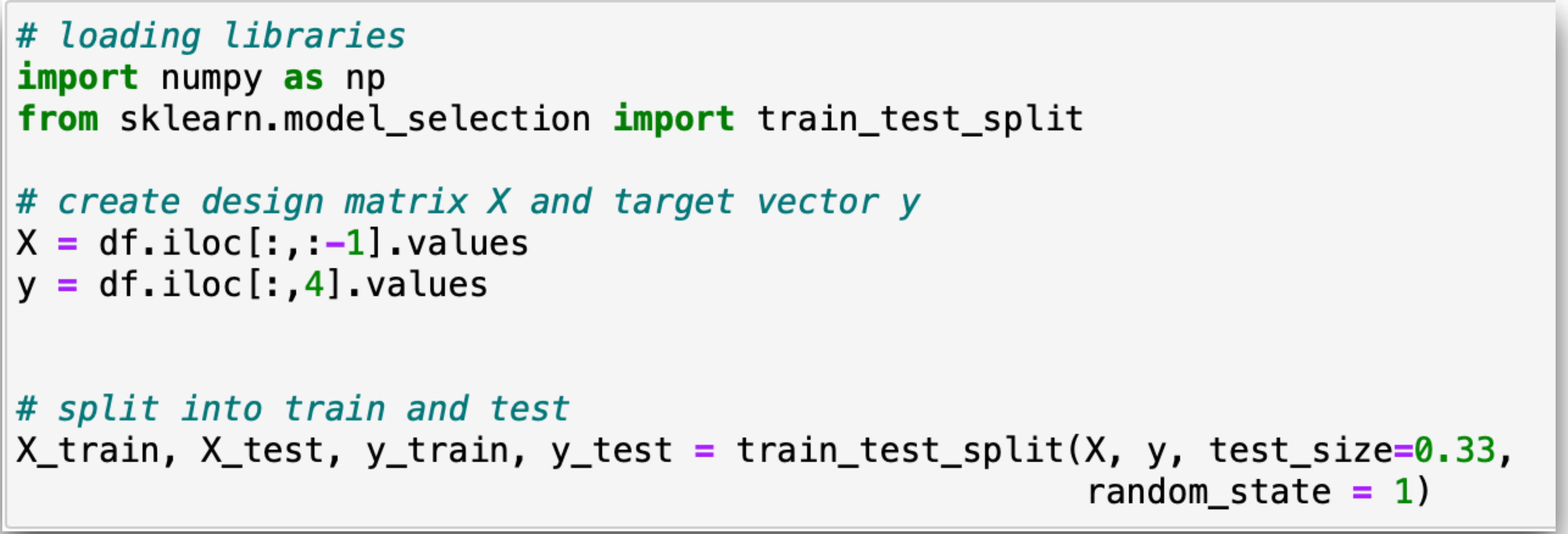
1. **確定k大小和距離度量 (k = 5)。**
2. **對於測試集中的一個樣本，找到訓練集中和它最近的k個樣本。**
3. **將這k個樣本的投票結果作為測試樣本的類別 (多數決)。**



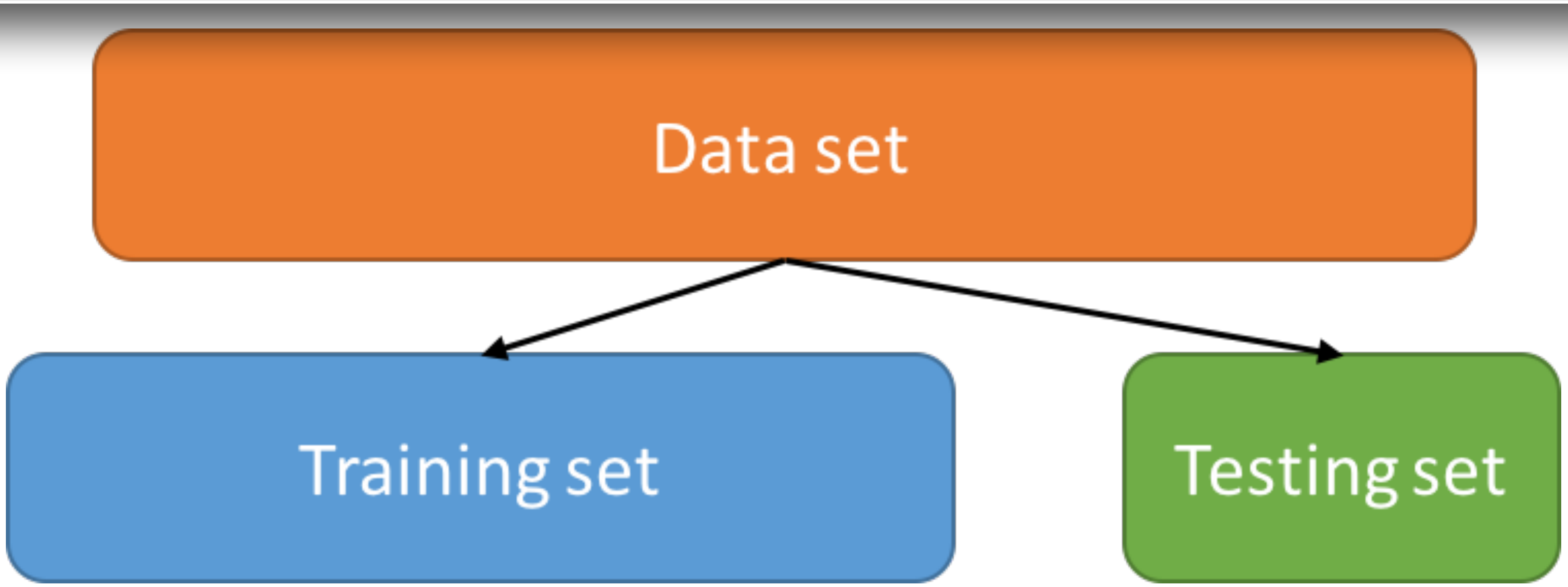
https://ljalphabeta.gitbooks.io/python-/content/knn.html

選擇演算法

- Split into train and test



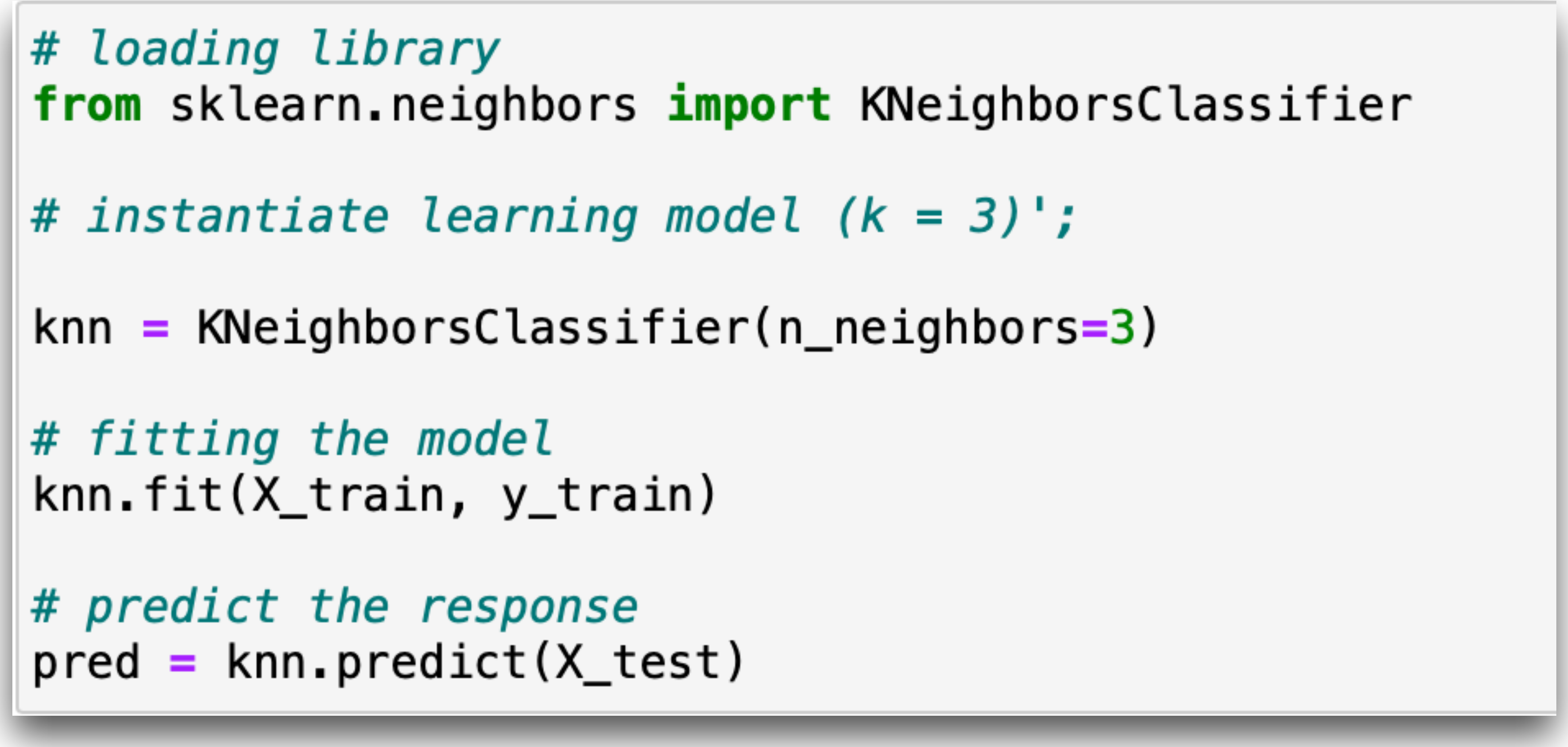
```python
# loading libraries
import numpy as np
from sklearn.model_selection import train_test_split

# create design matrix X and target vector y
X = df.iloc[:,:-1].values
y = df.iloc[:,4].values


# split into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,
                                                    random_state = 1)
```



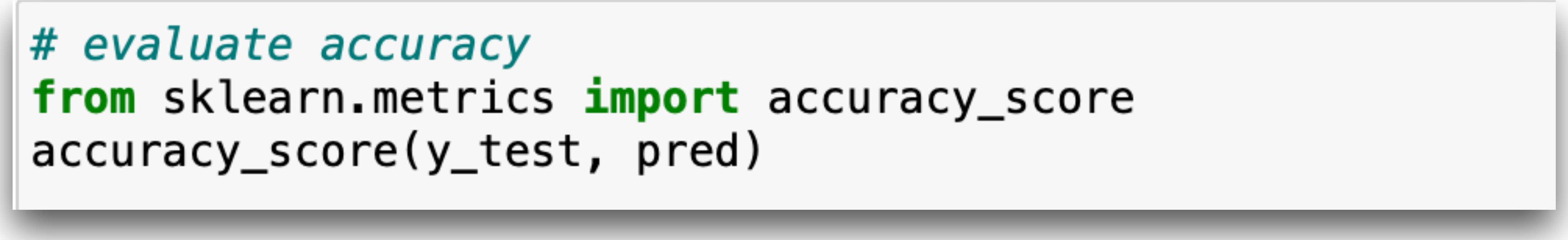
Source: https://aldro61.github.io/microbiome-summer-school-2017/sections/basics/

選擇演算法

- K-NN



```
# loading library
from sklearn.neighbors import KNeighborsClassifier

# instantiate learning model (k = 3)';

knn = KNeighborsClassifier(n_neighbors=3)

# fitting the model
knn.fit(X_train, y_train)

# predict the response
pred = knn.predict(X_test)
```



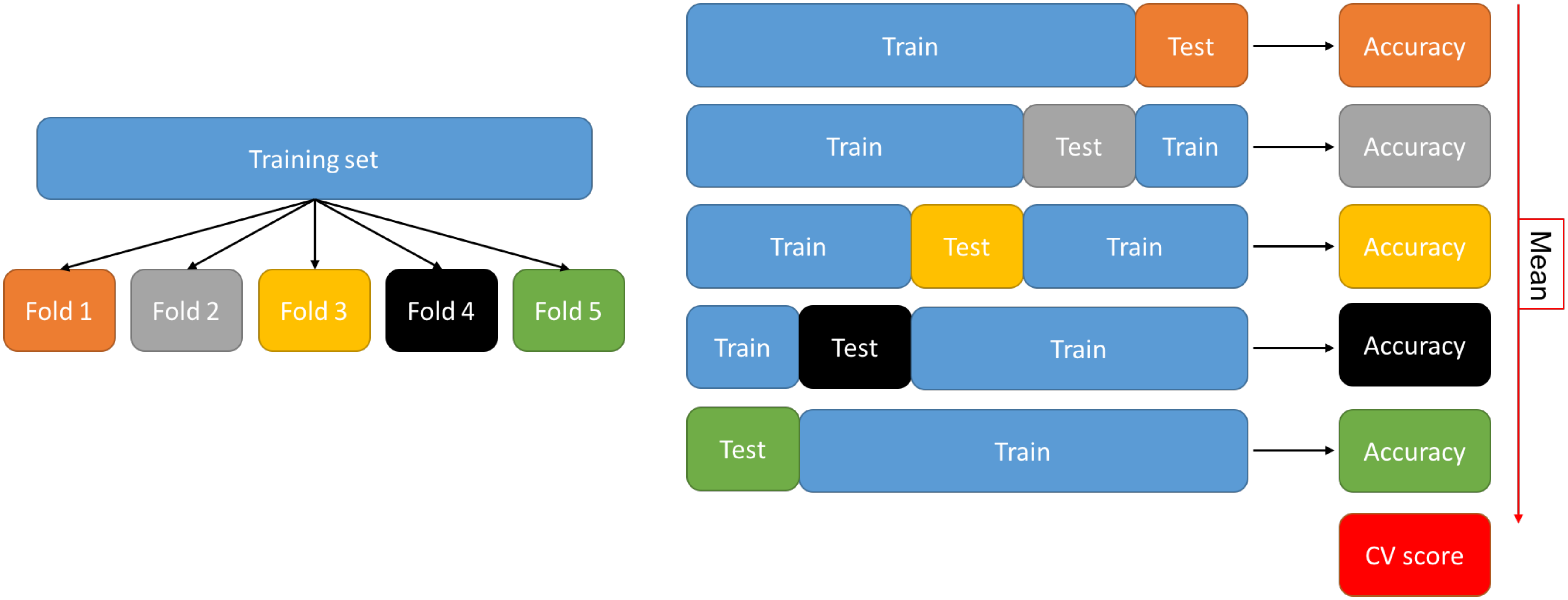
```
# evaluate accuracy
from sklearn.metrics import accuracy_score
accuracy_score(y_test, pred)
```

Machine Learning Process

- 準備資料　（包含資料預處理）
- 選擇演算法
- 調整參數
- 評估結果

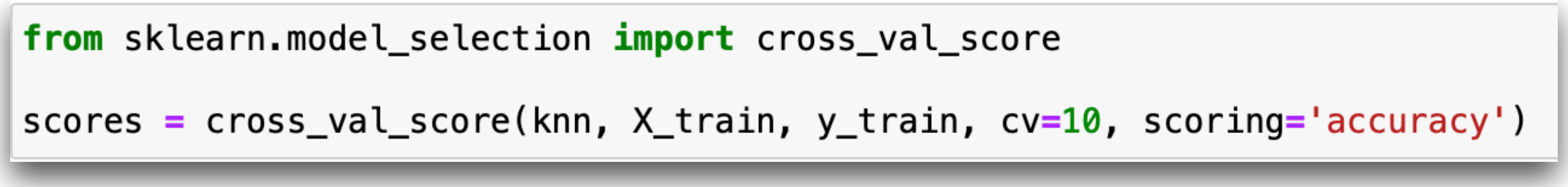
調整參數 & 評估結果

- cross-validation



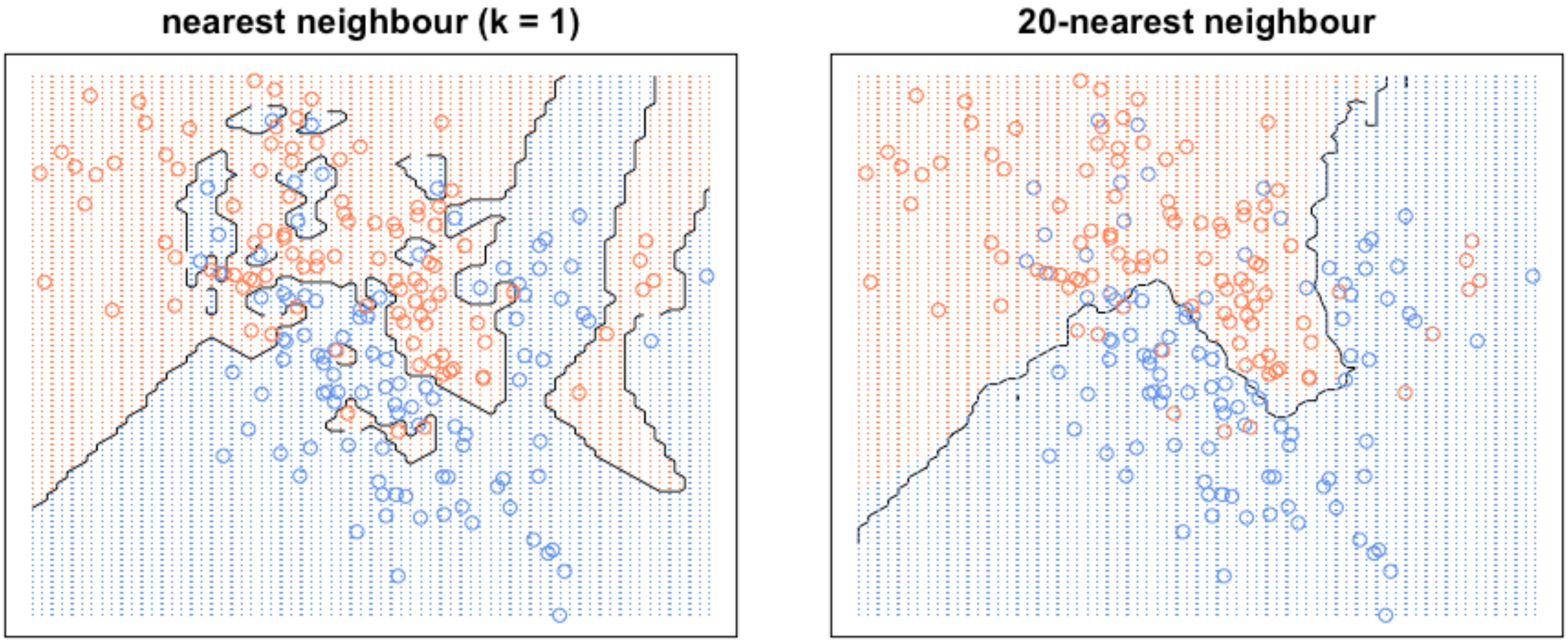
Source: https://aldro61.github.io/microbiome-summer-school-2017/sections/basics/

調整參數 & 評估結果



```
from sklearn.model_selection import cross_val_score

scores = cross_val_score(knn, X_train, y_train, cv=10, scoring='accuracy')
```

Underfitting v.s. Overfitting



nearest neighbour (k = 1)

20-nearest neighbour

https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/#exploring-knn-in-code

- k too small ➔ overfitting. Why?
- k too large ➔ underfitting. Why?