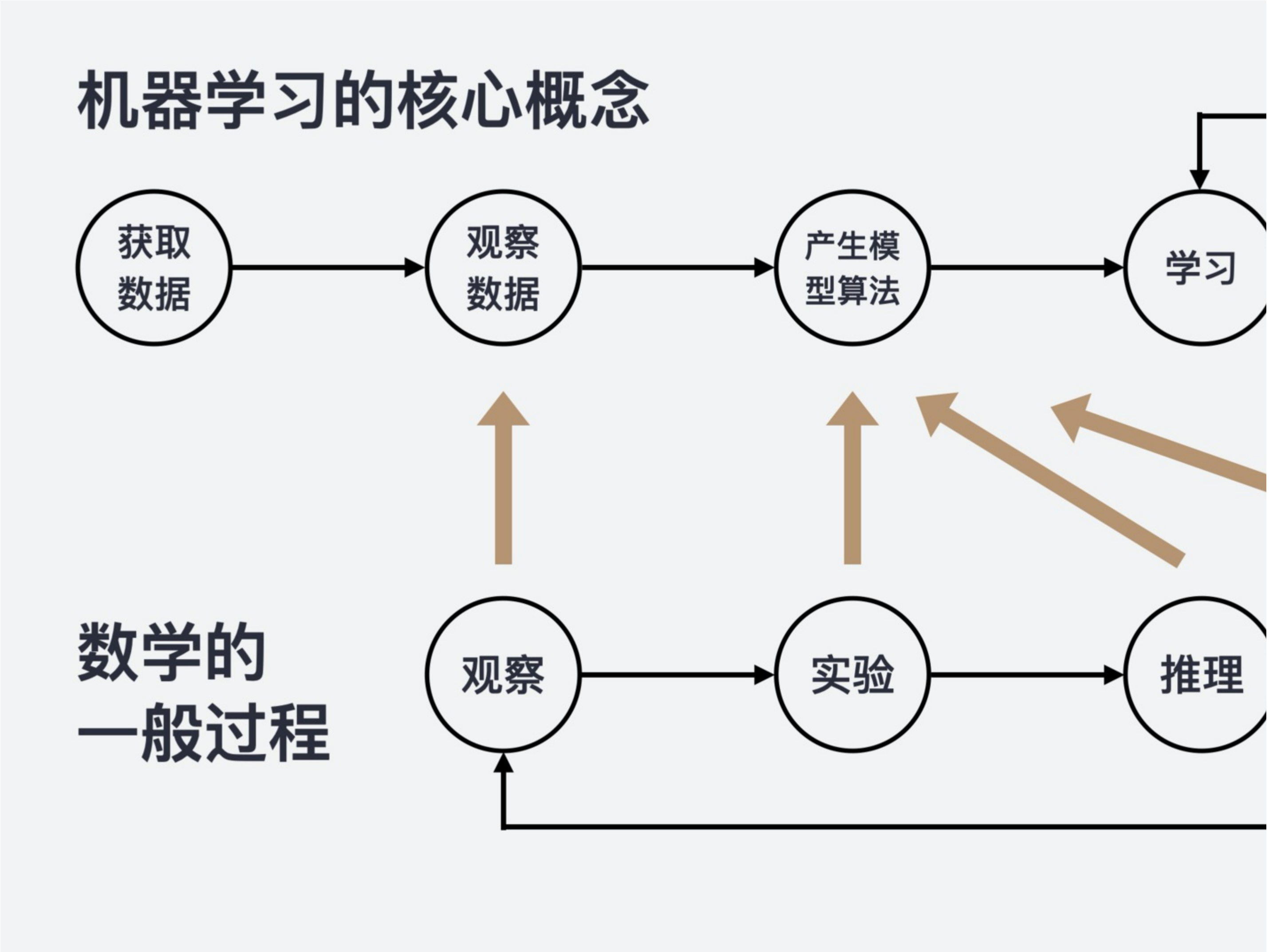
ä½ å¥½ï¼Œæˆ‘æ˜¯æœ±ç»´å˜ã€‚æ¬¢è¿Žä½ è·Ÿæˆ‘ä¸€èµ·é‡å¦çº¿æ€§ä»£æ•°ï¼

åœ¨å¼€ç¯‡è¯¾ä¸ï¼Œæˆ‘å’Œä½ å¤§è‡´è®²è¿‡æˆ‘è‡ªå·±çš„ç»åŽ†ï¼Œä»Ž2006å¹´å¼€å§‹åˆ°çŽ°åœ¨14å¹´çš„æ—¶é—´é‡Œï¼Œæˆ‘éƒ½ä¸“æ³¨äºŽæœºå™¨å¦ä¹ é¢†åŸŸã€‚å¯¹äºŽ**çº¿æ€§ä»£æ•°**åœ¨æœºå™¨å¦ä¹ ä¸çš„åº"ç"¨ï¼Œæˆ‘éžå¸¸äº†è§£ã€‚è€Œè¿™ä¹Ÿæ˜¯çº¿æ€§ä»£æ•°æœ€ä»¤è¦çš„åº"ç"¨åœºæ™¯ä¹‹ä¸€ã€‚å› æ¤ï¼Œä»Šå¤©ç¬¬ä¸€èŠ‚è¯¾ï¼Œæˆ‘æƒ³å…ˆå'Œä½ èŠä¸€èŠï¼Œå¦‚ä½•åœ¨æœºå™¨å¦ä¹ ä¸è¿ç"¨çº¿æ€§ä»£æ•°å·¥å…·ï¼Œåœ¨æˆ‘ä»¬å¼€å§‹è‡ªä¸‹è€Œä¸Šçš„å¦ä¹ ä¹‹å‰ï¼Œå…ˆä»Žä¸Šå±‚æ¥çœ‹ä¸€çœ‹ã€‚

æˆ‘ä»¬éƒ½çŸ¥é"ï¼Œâ€œæ•°æ®â€æ˜¯æœºå™¨å¦ä¹ çš„å‰æï¼Œæœºå™¨å¦ä¹ çš„ç¬¬ä¸€æ¥å°±æ˜¯è¦èŽ·å–æ•°æ®çš„æ¶¶é›†ç¾¤ï¼Œå¤„ç†å'Œç‰¹å¾æå–ï¼‰è€Œ**æ¨¡åž‹**å°±æ˜¯é€šè¿‡æ•°æ®æ¥å¦ä¹ çš„ç®—æ³•ï¼›**å¦ä¹ **åˆ™æ˜¯ä¸€ä¸ªå¾ªçŽ¯è¿‡ç¨‹ï¼Œä¸€ä¸ªè‡ªåŠ¨åœ¨æ•°æ®ä¸å¯»æ‰¾æ¨¡å¼ï¼Œå¹¶ä¸åœè°ƒæ•´æ¨¡åž‹å‚æ•°çš„è¿‡ç¨‹ã€‚é‚£æˆ‘ä»¬å°±ä»Žæœºå™¨å¦ä¹ çš„ä¸‰ä¸ªæ ¸å¿ƒæ¦‚å¿µï¼šæ•°æ®ã€æ¨¡åž‹å'Œå¦ä¹ è¯´èµ·ã€‚



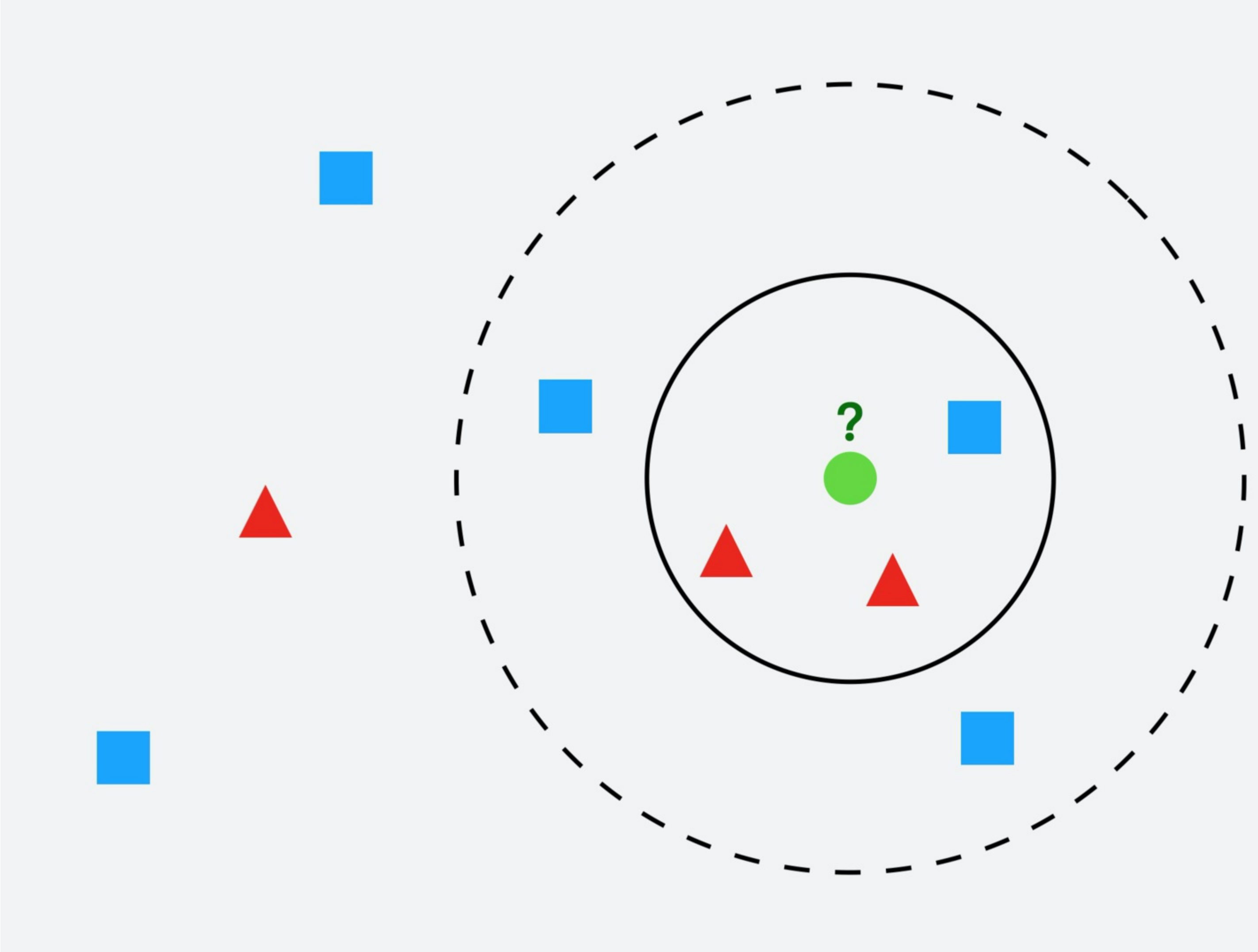
ä½ çœ‹ï¼Œä¸è®ºæ˜¯æ¨¡åž‹ï¼Œè¿˜æ˜¯å¦ä¹ ï¼Œéƒ½æ¶‰åŠæ•°æ®ï¼Œè€Œæ•°æ®åŠ ä¸Šæ¨¡åž‹å'Œå¦ä¹ ï¼Œå°±æ˜¯æ•°å¦çš„ä¸€èˆ¬è¿‡ç¨‹äº†ï¼Œä¹Ÿå°±æ˜¯ï¼šè§‚å¯Ÿã€å®žéªŒã€æŽ¨ç†å'ŒæŠ½è±¡ã€‚æ‰€ä»¥ï¼Œæˆ‘è®¤ä¸ºå¦å¥½æ•°å¦ï¼Œä¸ä»…æœ‰åˆ©äºŽç†è§£å¤æ‚çš„æœºå™¨å¦ä¹ ç³»ç»Ÿï¼Œè¿˜èƒ½è°ƒä¼˜ç®—æ³•å‚æ•°ï¼Œç"šè‡³èƒ½å¸®åŠ©ä½ åˆ›å»ºæ–°çš„æœºå™¨å¦ä¹ è§£å†³æ–¹æ¡ˆã€‚

ä»Žæœºå™¨å¦ä¹ åˆ°çº¿æ€§ä»£æ•°

é‚£æœºå™¨å¦ä¹ å'Œçº¿æ€§ä»£æ•°ä¹‹é—´åˆ°åº•æœ‰ç€æ€Žæ ·çš„å…³ç³»å'¢ï¼Ÿæˆ‘æƒ³ï¼Œç"¨ä¸€ä¸ªå®žé™…çš„æœºå™¨å¦ä¹ ç®—æ³•çš„ä¾‹åæ¥è§£é‡Šï¼Œä½ å¯èƒ½æ›´å®¹æ˜“æŽ¥å—ã€‚æŽ¥ä¸‹æ¥ï¼Œæˆ‘ä½¿ç"¨KNNï¼ˆK-Nearest Neighborï¼ŒKæœ€è¿'é‚»åˆ†ç±»ç®—æ³•ï¼‰æ¥è®©ä½ ç®€å•äº†è§£ä¸€ä¸‹æœºå™¨å¦ä¹ ï¼Œä»¥åŠå®ƒå'Œçº¿æ€§ä»£æ•°ä¹‹é—´çš„å…³ç³»ã€‚

ä¹‹æ‰€ä»¥é€‰KNNåˆ†ç±»ç®—æ³•ï¼Œå› ä¸ºå®ƒæ˜¯ä¸€ä¸ªç†è®ºä¸Šæ¯"è¾ƒæˆç†Ÿçš„æ–¹æ³•ï¼Œä¹Ÿæ˜¯æœ€ç®€å•çš„æœºå™¨å¦ä¹ ç®—æ³•ä¹‹ä¸€ã€‚è¿™ä¸ªæ–¹æ³•çš„æ€è·¯æ˜¯ï¼šå¦‚æžœä¸€ä¸ªæ ·æœ¬åœ¨ç‰¹å¾ç©ºé—´ä¸çš„Kä¸ªæœ€ç›¸ä¼¼ï¼ˆå³ç‰¹å¾ç©ºé—´ä¸æœ€é‚»è¿'ï¼‰çš„æ ·æœ¬ä¸çš„å¤§å¤šæ•°å±žäºŽæŸä¸€ä¸ªç±»åˆ«ï¼Œåˆ™è¯¥æ ·æœ¬ä¹Ÿå±žäºŽè¿™ä¸ªç±»åˆ«ã€‚

è¿™é‡Œæœ‰ä¸ªå‰æï¼ŒKNNç®—æ³•ä¸ï¼Œæ‰€é€‰æ‹©çš„é‚»å±…éƒ½æ˜¯å·²ç»æ£ç¡®åˆ†ç±»çš„å¯¹è±¡ã€‚KNNåˆ†ç±»ç®—æ³•åœ¨åˆ†ç±»å†³ç–ä¸Šåªä¾æ®æœ€é‚»è¿'çš„ä¸€ä¸ªæˆ–è€…å‡ ä¸ªæ ·æœ¬çš„ç±»åˆ«ï¼Œæ¥å†³å®šå¾…åˆ†æ ·æœ¬æ‰€å±žçš„ç±»åˆ«ã€‚æˆ'ä»¬é€šè¿‡å›¾æ¥ç†è§£çš„è¯æ›´è®¤æ›´å®¹æ˜"ä¸€äº›ã€‚



å‡è®¾å›¾ç‰‡ä¸é‚£ä¸ªç»¿è‰²åœ†å½¢è¦æ˜¯æˆ‘ä»¬è¦å†³ç–çš„å¯¹è±¡ï¼Œé‚£ä¹ˆæ ¹æ®KNNç®—æ³•å®ƒå±žäºŽå“ªä¸€ç±»ï¼Ÿæ˜¯çº¢è‰²ä¸‰è§'å½¢è¿˜æ˜¯è"è‰²å››æ–¹å½¢ï¼Ÿ

å¦‚æžœK=3ï¼ˆå®žçº¿åœ†ï¼‰ï¼Œä¹Ÿå°±æ˜¯åŒ…å«ç¦»ç»¿è‰²åœ†æœ€è¿'çš„3ä¸ªï¼Œç"±äºŽçº¢è‰²ä¸‰è§'å½¢æ‰€å æ¯"ä¾‹ä¸º2/3ï¼Œç»¿è‰²åœ†å°±å±žäºŽçº¢è‰²ä¸‰è§'å½¢é‚£ä¸ªç±»ã€‚ä½†å¦‚æžœK=5ï¼ˆè™šçº¿åœ†ï¼‰ï¼Œå°±æ˜¯åŒ…å«ç¦»ç»¿è‰²åœ†æœ€è¿'çš„5ä¸ªï¼Œç"±äºŽè"è‰²å››æ–¹å½¢æ¯"ä¾‹ä¸º3/5ï¼Œç»¿è‰²åœ†å°±å±žäºŽè"è‰²å››æ–¹å½¢é‚£ä¸ªç±»ã€‚

é¸¢å°¾èŠ±åˆ†ç±»é—®é¢˜çš„ç‰¹æ€§ä»£æ•°

é€šè¿‡å‰é¢è¿™ä¸ªå°ä¾‹åï¼Œä½ å"è¯¥å·²ç»ç†è§£äº†KNNç®—æ³•çš„æ¦‚å¿µã€‚é‚£ä¹ˆæŽ¥ä¸‹æ¥ï¼Œæˆ'ä»¬å°±è¯•ç€ä½¿ç"¨KNNåœ¨ç»™å®šé¸¢å°¾èŠ±ç‰¹å¾å€¼çš„æƒ…å†µä¸‹ï¼Œç»™é¸¢å°¾èŠ±åšèŠ±ç§åˆ†ç±»ï¼Œå¸¦ä½ æ¥å®žé™…çœ‹ä¸€ä¸‹ç‰¹æ€§ä»£æ•°åœ¨è¿™é‡Œèµ·åˆ°çš„ä½œç"¨ã€‚

ç‰¹åˆ«è¯´æ˜Žä¸€ä¸‹ï¼Œ**é¸¢å°¾èŠ±åˆ†ç±»é—®é¢˜**æ˜¯ä¸€ä¸ªå›½é™…ä¸Šé€šç"¨çš„æ¡ˆä¾‹ï¼Œä¸€èˆ¬éƒ½è¢«ä½œä¸ºæœºå™¨å¦ä¹ å…¥é—¨æ¥ä½¿ç"¨ï¼Œæ‰€ä»¥å®ƒçš„æ•°æ®é›†ä¹Ÿæ˜¯å…¬å¼€çš„ã€‚

1.æ•°æ®é›†çš„æ”¶é›†ã€åŠ è½½å'Œåˆ†æž

é¦–å…ˆï¼Œæˆ'ä»¬è¦åšçš„æ˜¯æ•°æ®é›†çš„æ"¶é›†ã€åŠ è½½å'Œåˆ†æžï¼Œä½ ä¹Ÿå¯ä»¥ç‚¹å‡»è¿™é‡Œä¸‹è½½åŽŸå§‹æ•°æ®é›†ï¼ŒæŽ¥ç€ç œåŽŸå§‹æ•°æ®é›†ä¸ä»¥æ ·ï¼Œä¸‹é¢æ˜¯èŽ·å–å'ŒåŠ è½½æ•°æ®çš„ä»£ç ï¼Œsklearnæ•°æ®é›†å·²ç»åŒ…å«æ ·æœ¬æ•°æ®ï¼Œä½ å¯ä»¥ç›´æŽ¥ç"¨ã€‚

```
import pandas as pd

from sklearn import datasets
iris = datasets.load_iris()

species = [iris.target_names[x] for x in iris.target]

iris = pd.DataFrame(iris['data'], columns = ['Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width'])

iris['Species'] = species
```

ä»Žæ˜¾ç¤ºçš„ç»"æžœï¼Œæˆ'ä»¬èƒ½å¤Ÿçœ‹å‡ºé¸¢å°¾èŠ±æœ‰å››ä¸ªç‰¹å¾ï¼šèŠ±è¼çš„é•¿ã€å®½å'ŒèŠ±ç"£çš„é•¿ã€å®½ã€‚æˆ'ä»¬æ¥çœ‹ä¸‹è¿™å››ä¸ªç‰¹å¾çš„æ•°æ®ç±»åž‹ï¼š

```
iris.dtypes
Sepal_Length    float64
Sepal_Width     float64
Petal_Length    float64
Petal_Width     float64
Species          object
dtype: object
```

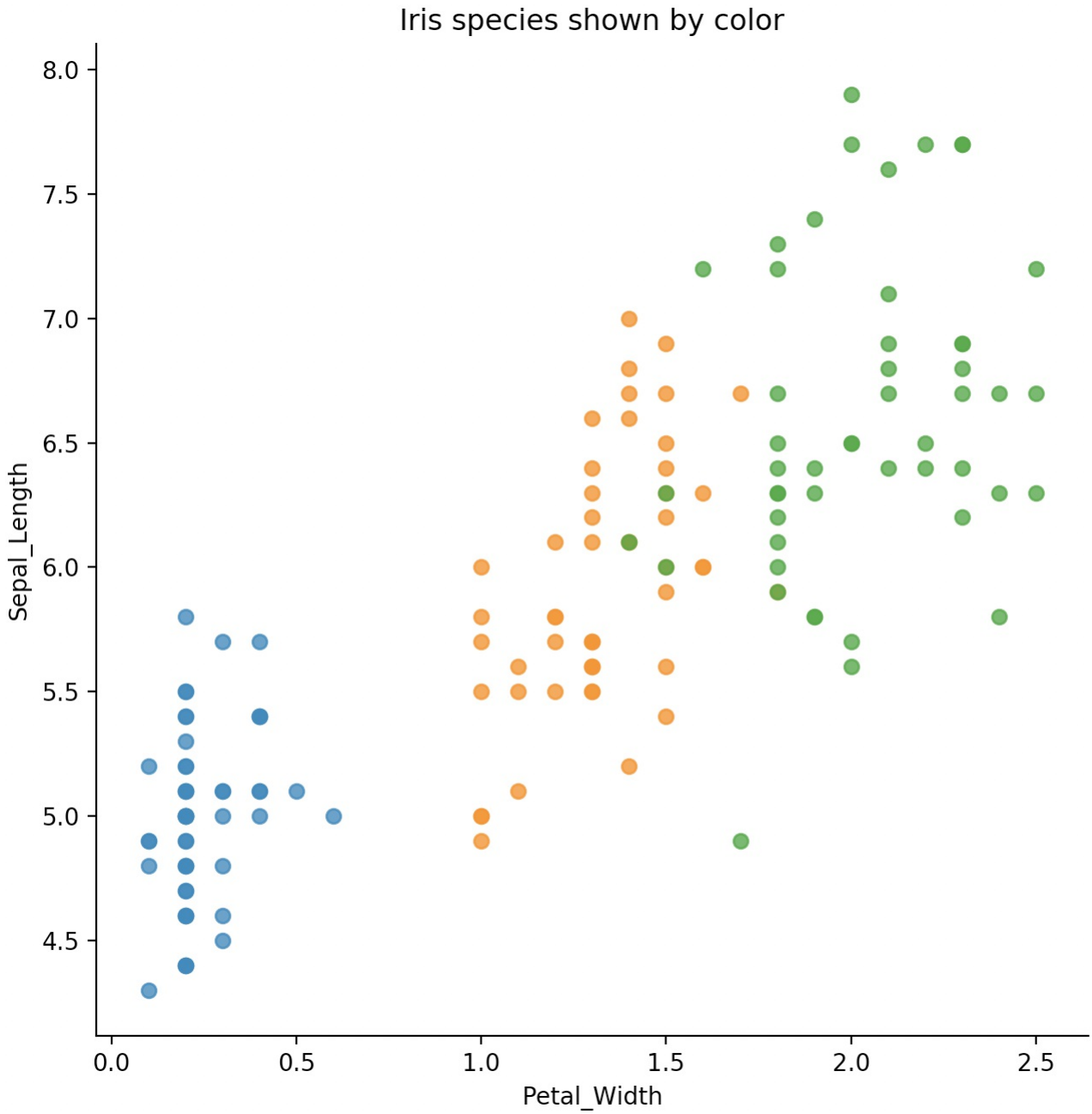
è¿™äº›ç‰¹å¾éƒ½æ˜¯æ•°å€¼åž‹ï¼Œè€Œä¸"æ ‡ç¾Speciesè¡¨ç¤ºçš„æ˜¯èŠ±ç§ï¼Œæ˜¯ä¸€ä¸ªå—ç¬¦ä¸²ç±»åž‹çš„å˜é‡ã€‚æˆ'ä»¬ç»§ç»çœ‹ä¸€ä¸‹é¸¢å°¾èŠ±çš„åˆ†ç±»ç»Ÿè®¡ï¼š

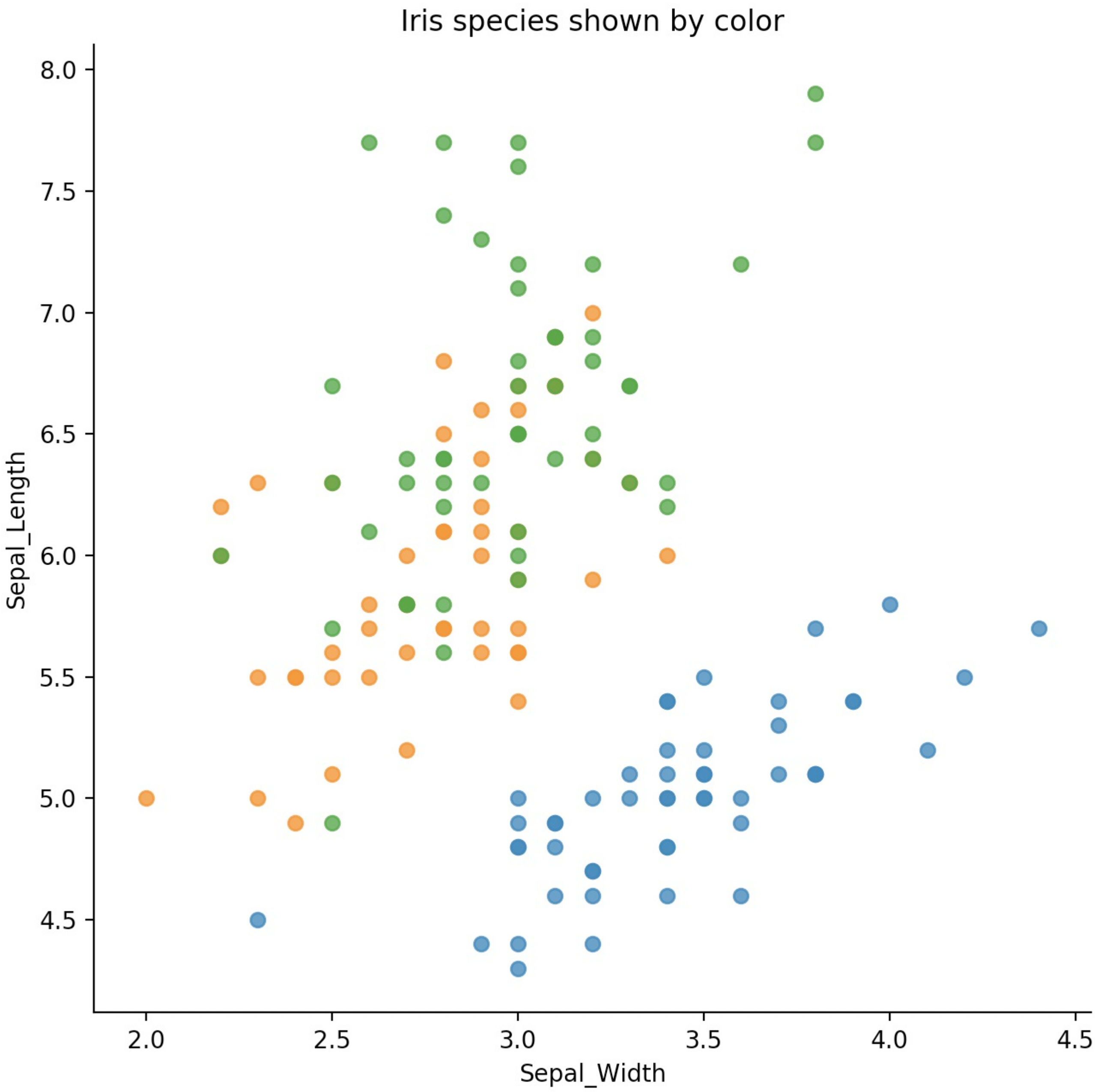
```
iris['count'] = 1
iris[['Species', 'count']].groupby('Species').count()
```

Species	count
setosa	50
versicolor	50
virginica	50

è¿™é‡Œæˆ‘ä»¬ç›´æŽ¥è¿ƒ½å¤Ÿçœ‹åˆ°ï¼Œé¸¢å°¾èŠ±æœ‰ä¸‰ä¸ªèŠ±ç§ï¼Œæ¯ä¸ªç§ç±»æœ‰50ä¸ªå®žä¾‹ï¼Œæˆ–è€…è¯´50æ¡æ•°æ®ï¼Œæˆ‘ä»¬å†ç”¨å›¾æ¥æ›´ç›´è§‚åœ°æ˜¾ç¤ºè¿™ä¸‰ç§é¸¢å°¾èŠ±ã€‚

```
%matplotlib inline

def plot_iris(iris, col1, col2):
    import seaborn as sns
    import matplotlib.pyplot as plt

    sns.lmplot(x = col1, y = col2,
               data = iris,
               hue = "Species",
               fit_reg = False)

    plt.xlabel(col1)

    plt.ylabel(col2)

    plt.title('Iris species shown by color')

    plt.show()

plot_iris(iris, 'Petal_Width', 'Sepal_Length')

plot_iris(iris, 'Sepal_Width', 'Sepal_Length')
```


Iris species shown by color
Sepal_Length
Petal_Width
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
0.0
0.5
1.0
1.5
2.0
2.5



è“ã€é»‚ã€ç»¿ï¼Œè¿™ä¸‰ç§é¢œè‰²åˆ†åˆ«ä»£è¡¨äº†ä¸‰ç§é¸¢å°¾èŠ±ï¼Œæ˜¾ç¤ºè¿˜æ˜¯å¾ˆæ¸…æ¥šçš„ã€‚

2.æ•°æ®é›†çš„å‡†å¤‡

æŽ¥ä¸‹æ¥çš„ç¬¬äºŒæ¥å°±æ˜¯æ•°æ®é›†çš„å‡†å¤‡äº†ã€‚åœ¨è®ç»ƒä»»ä½•æœºå™¨å¦ä¹ æ¨¡åž‹ä¹‹å‰ï¼Œæ•°æ®å‡†å¤‡éƒ½æ˜¯å¿…ä¸å¯å°‘çš„ä¸€æ¥ï¼Œè¿™é‡Œä¹Ÿè¦æ‰¾ä¸ªå¥½å‡†å¤‡ã€‚

ç¬¬ä¸€æ¥ï¼Œç‰¹å¾æ•°å€¼æ ‡å‡†åŒ–ã€‚å¦‚æžœæˆ‘ä»¬ä¸åšæ ‡å‡†åŒ–ï¼ŒåŽæžœå°±æ˜¯å¤§æ•°å€¼ç‰¹å¾ä¼šä¸»å¯¼æ¨¡åž‹è®ç»ƒï¼Œè¿™ä¼šå¯¼è‡´
´æ›´æœ‰æ„ä¹‰çš„å°æ•°å€¼ç‰¹å¾è¢«å¿½ç•¥ã€‚è¿™é‡Œæˆ‘ä»¬ç”¨Z Scoreæ ‡å‡†åŒ–ï¼Œä½¿æ¯ä¸€ç±»ç‰¹å¾çš„å‡å€¼ä¸º0ï¼Œæ–¹å·®ä¸º1.0ï¼Œæˆ‘ä»¬å¯ä»¥é€šè¿‡ä»£ç å®žçŽ°æ¥çœ‹ä¸‹æ•ˆæžœã€‚

```
from sklearn.preprocessing import scale

import pandas as pd

num_cols = ['Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width']

iris_scaled = scale(iris[num_cols])

iris_scaled = pd.DataFrame(iris_scaled, columns = num_cols)

print(iris_scaled.describe().round(3))
```

	Sepal_Length	Sepal_Width	Petal_Length
count	150.000	150.000	150.000
mean	-0.000	-0.000	-0.000
std	1.003	1.003	1.003
min	-1.870	-2.434	-1.568
25%	-0.901	-0.592	-1.227
50%	-0.053	-0.132	0.336
75%	0.675	0.559	0.763
max	2.492	3.091	1.786

ä½ å¯ä»¥çœ‹åˆ°ï¼Œæ¯ä¸€åˆ—å¹³å‡å€¼ä¸º0ï¼Œæ ‡å‡†å·®å¤§çº¦æ˜¯1.0ã€‚ä¸ºäº†åˆ†ç±»éœ€è¦ï¼Œæˆ‘ä»¬ç”¨å—å…¸æŠŠè§ç§ä»Žå—ç¬¦ä¸²ç±»åž‹è½¬æ¢æˆæ•°å—è¡¨ç¤ºã€‚

```
levels = {'setosa':0, 'versicolor':1, 'virginica':2}

iris_scaled['Species'] = [levels[x] for x in iris['Species']]

iris_scaled.head()
```

	Sepal_Length	Sepal_Width	Petal_Length	Petal_W
0	-0.900681	1.019004	-1.340227	-1.315
1	-1.143017	-0.131979	-1.340227	-1.315
2	-1.385353	0.328414	-1.397064	-1.315
3	-1.506521	0.098217	-1.283389	-1.315
4	-1.021849	1.249201	-1.340227	-1.315

ç¬¬äºŒæ¥ï¼ŒæŠŠæ•°æ®é›†éšæœºåˆ†å‰²æˆä¸¤ä¸ªæ•°æ®é›†ï¼šè®ç»ƒé›†å'Œè¯•é›†ï¼Œè®ç»ƒé›†ç"¨æ¥è®ç»ƒKNNæ¨¡åž‹ï¼Œè¯•é›†ç"¨æ¥æµ‹è¯•å'Œè¯„ä¼°KNNçš„åˆ†ç±»æ•ˆæžœã€‚

```
from sklearn.model_selection import train_test_split

import numpy as np

np.random.seed(3456)

iris_split = train_test_split(np.asmatrix(iris_scaled), test_size = 75)

iris_train_features = iris_split[0][:, :4]

iris_train_labels = np.ravel(iris_split[0][:, 4])

iris_test_features = iris_split[1][:, :4]

iris_test_labels = np.ravel(iris_split[1][:, 4])

print(iris_train_features.shape)

print(iris_train_labels.shape)

print(iris_test_features.shape)

print(iris_test_labels.shape)
```

é€šè¿‡ä»£ç ï¼Œæˆ'ä»¬å¾—åˆ°äº†è¿™æ ·çš„ç»"æžœã€‚

```
(75, 4)
(75,)
(75, 4)
(75,)
```

3.è®ç»ƒæ¨¡åž‹

æ•°æ®å‡†å¤‡å¥½åŽï¼Œå°±æ˜¯ç¬¬ä¸‰æ¥è®ç»ƒæ¨¡åž‹äº†ã€‚è¿™é‡Œæˆ'ä»¬ä½¿ç"¨K=3æ¥è®ç»ƒKNNæ¨¡åž‹ï¼Œå½“ç„¶ä½ ä¹Ÿå¯ä»¥è°ƒæ•´è¿™ä¸ªå‚æ•°æ¥è¿›è¡Œè§‚å¯Ÿå'Œè°ƒè¯•ã€‚

```
from sklearn.neighbors import KNeighborsClassifier

KNN_mod = KNeighborsClassifier(n_neighbors = 3)

KNN_mod.fit(iris_train_features, iris_train_labels)
```

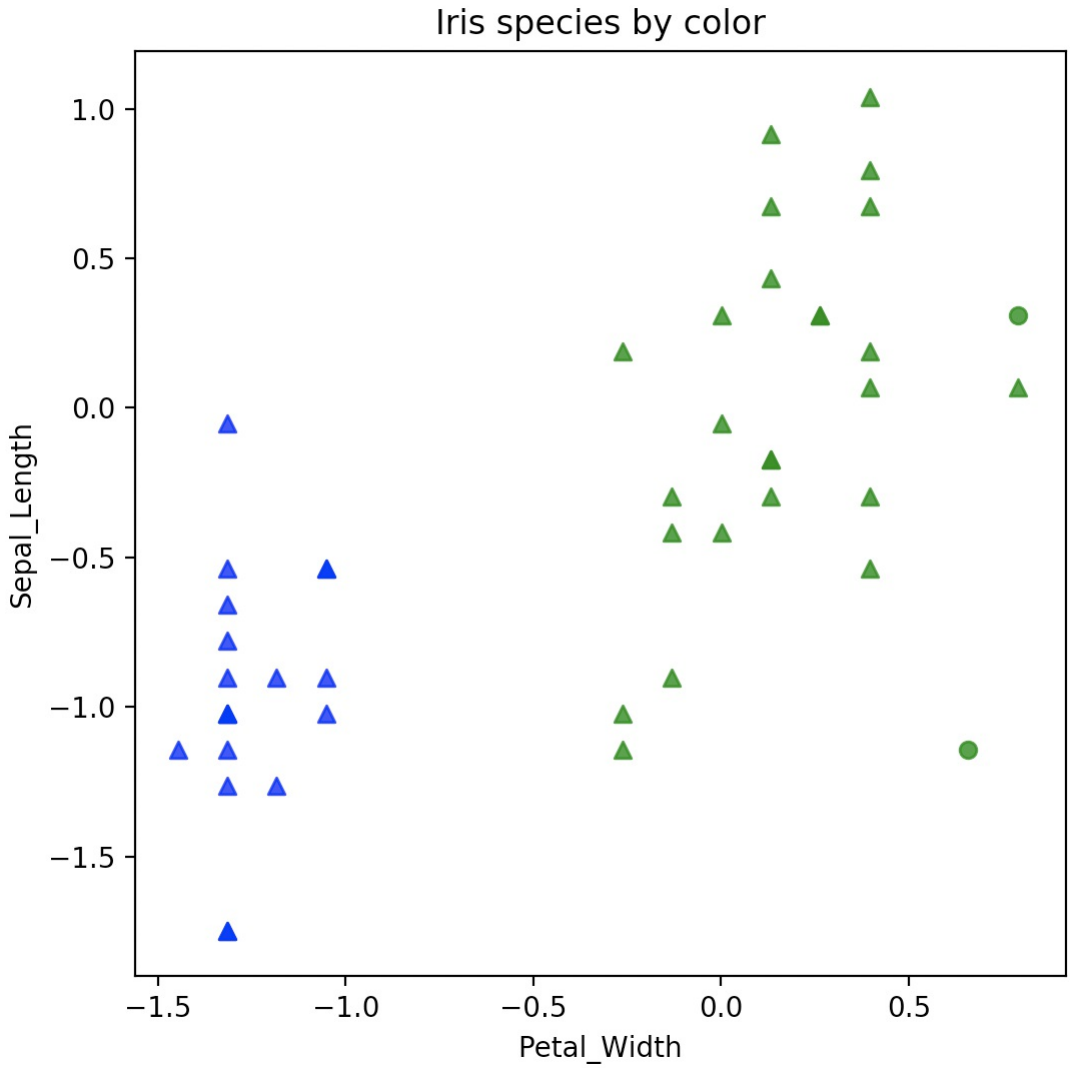
4.æ¨¡åž‹æµ‹è¯•

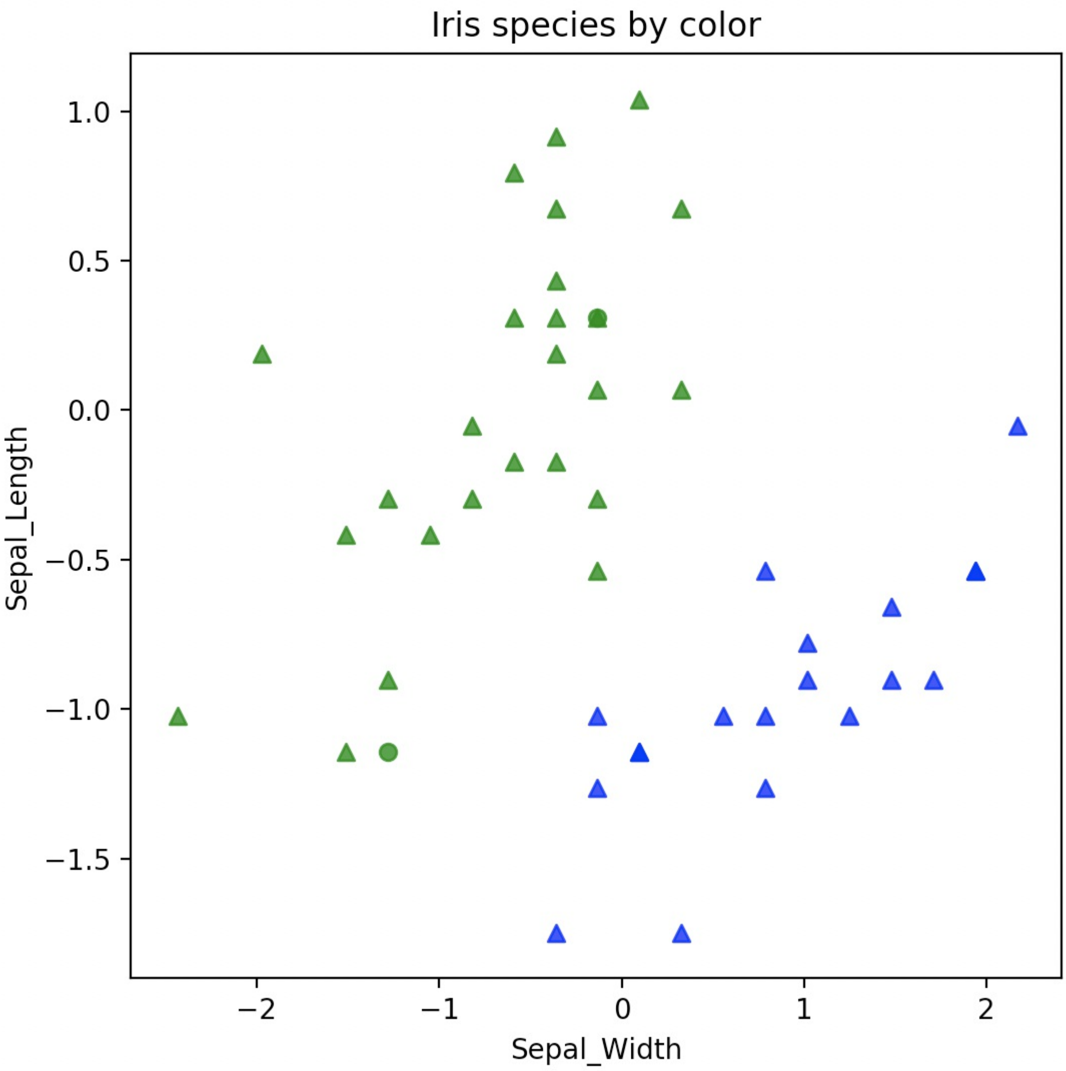
æ‰§è¡ŒKNNè®ç»ƒåŽï¼Œæˆ'ä»¬æ¥åˆ°äº†æœ€åŽä¸€æ¥ï¼Œæ¨¡åž‹æµ‹è¯•ï¼Œè¿™é‡Œæˆ'ä»¬ä½¿ç"¨æµ‹è¯•é›†æ¥æµ‹è¯•æ¨¡åž‹ã€‚

```
iris_test = pd.DataFrame(iris_test_features, columns = num_cols)

iris_test['predicted'] = KNN_mod.predict(iris_test_features)

iris_test['correct'] = [1 if x == z else 0 for x, z in zip(iris_test['predicted'], iris_test_labels)]

accuracy = 100.0 * float(sum(iris_test['correct'])) / float(iris_test.shape[0])

print(accuracy)

96.0
```

æœ€ç»ˆï¼Œæˆ'ä»¬å¾—åˆ°çš„å‡†ç¡®çŽ‡æ˜¯96.0ï¼Œè¯´æ˜Žäº†KNNçš„è®ç»ƒæ¨¡åž‹ä¸é"™ï¼Œé€‚ç"¨è¿™ç±»åœºæ™¯ã€‚æˆ'ä»¬é€šè¿‡ä»£ç æŠŠå…¶ä¸çš„ä¸¤ä¸ªåˆ†ç±»setosaå'Œversicoloræ‰"å°å‡ºæ¥çœ‹çœ‹ã€‚

```
levels = {0:'setosa', 1:'versicolor', 2:'virginica'}

iris_test['Species'] = [levels[x] for x in iris_test['predicted']]

markers = {1:'^', 0:'o'}

colors = {'setosa':'blue', 'versicolor':'green',}

def plot_shapes(df, col1,col2,  markers, colors):
    import matplotlib.pyplot as plt
    import seaborn as sns

    ax = plt.figure(figsize=(6, 6)).gca() # define plot axis

    for m in markers: # iterate over marker dictioary keys
        for c in colors: # iterate over color dictionary keys
            df_temp = df[(df['correct'] == m)  & (df['Species'] == c)]
            sns.regplot(x = col1, y = col2,
                        data = df_temp,
                        fit_reg = False,
                        scatter_kws={'color': colors[c]},
                        marker = markers[m],
                        ax = ax)
    plt.xlabel(col1)
    plt.ylabel(col2)
    plt.title('Iris species by color')
    return 'Done'

plot_shapes(iris_test, 'Petal_Width', 'Sepal_Length', markers, colors)
plot_shapes(iris_test, 'Sepal_Width', 'Sepal_Length', markers, colors)
```





ä»Žæ˜¾ç¤ºçš„æ•ˆæžœæ¥è¯´ï¼Œå†ç±»è¿˜æ˜¯æŒºæ˜Žæ˜¾çš„ï¼Œç†Ÿæ‚‰äº†æœ€åŸºç¡€çš„æœºå™¨å¦ä¹ è¿‡ç¨‹åŽï¼Œä½ å¯èƒ½ä¼šé—®ï¼Œè®²äº†åŠå¤©ï¼Œç®—æ³•ä»£ç 在哪里呢？å…³é”®å°±在KNeighborsClassifieræ¨¡å—上，这个模型的实现，其实也不复杂。

é¦–å…ˆï¼Œå›¾ä¸è‰²彩相近的实际上就是距离更近的点，分类的结果主要取决于与最近邻点的类别。

æŽ¥ç€ï¼Œé‡‡æ ·æœ¬之间的距离计算，常用的距离计算方法是欧氏距离，两点之间的欧氏距离，即为直线距离：

$$d=\sqrt{\left(x_{1}-x_{2}\right)^{2}+\left(y_{1}-y_{2}\right)^{2}}$$

æ‰©å±•åˆ°多维空间同理。

ä½ çœ‹ï¼Œè¿™å°±是机器学习的基本流程。KNN是一种简单而有效的算法……

çŽ°åœ¨ï¼Œä½ æ˜¯ä¸æ˜¯已经大致理解了机器学习的基本原理了呢？

æœ¬èŠ‚å°ç»“

å¥½äº†ï¼Œåˆ°这里，我们本节的内容就讲完了。

è¿™ä¸€èŠ‚æˆ‘ä»¬é€šè¿‡ä¸€ä¸ªç®€å•çš„KNNï¼Œå¦ä¹ 了机器学习的基本流程……

æœºå™¨å¦ä¹ ä¸ï¼Œæ•°æ®é¢„å¤„ç†……

åŒæ—¶ï¼Œæˆ‘ä»¬ä¹Ÿè¦æ³¨æ„……