プロジェクト第２回

機械学習・データ解析

今回扱うデータセット

アヤメデータ(Iris dataset)

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

1936年に生物学者Roland Fisherの論文に掲載された3種類の花の品種が含まれているデータセット

サンプル数	150
クラス数	3
特徴量	4つ

今回扱うデータセット

■クラス

3つの品種が含まれている

setosa (セトサ)

versicolor (バージカラー)

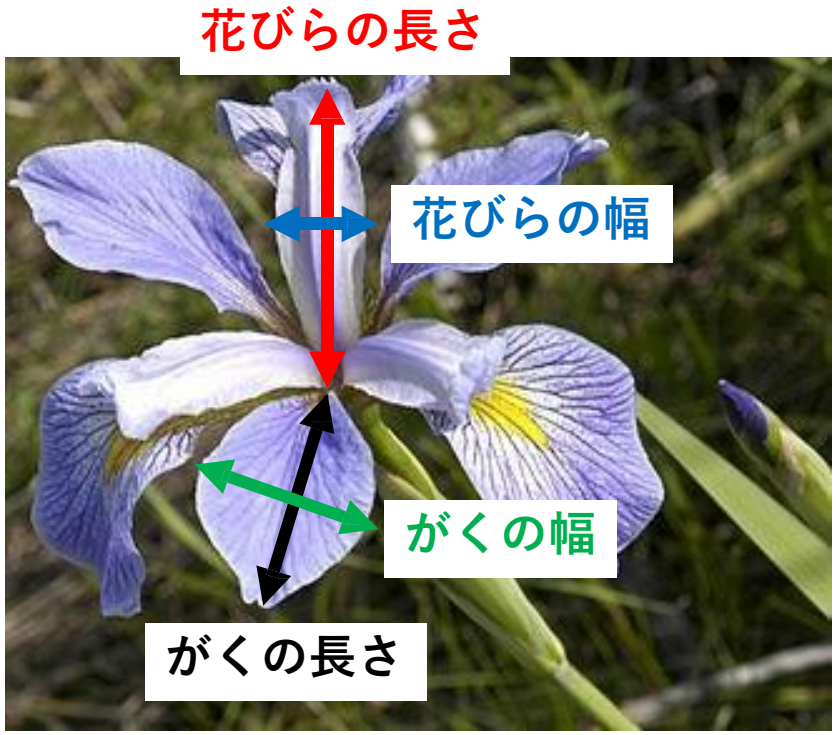
virginica (バージニカ)

今回扱うデータセット

■特徴量（説明変数）

以下の4つの特徴量が含まれている

- がくの長さ
- がくの幅
- 花びらの長さ
- 花びらの幅



Index of /ml/r

- Parent Directory
- Index
- bezdekIris.data
- iris.data
- iris.names

iris (1) - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
```

データセットのインポート

機械学習の便利な機能が使える「scikit-learn」と呼ばれるライブラリを用いてデータセットを呼び出す。

Using keywords: Python scikit-learn

Find the information from Web to read

scikit-learnとは？

オープンソースで公開されているPythonの機械学習ライブラリ、無料で利用可能。

- 機械学習モデルを簡単に構築できる（K近傍法、SVMなど）
- データセットも公開されている（ボストンの住宅不動産の値段データセット、アヤメデータセット、手書き数字データセットなど）

データセットのインポート

以下のコードを実行するとインポートすることができ、データの中身を確認できる。

```
from sklearn.datasets import load_iris


iris = load_iris()
print("入力データ")
print(iris.data)
print("¥n")
print("正解データ")
print(iris.target)
```

```
from sklearn.datasets import load_iris

iris_data = load_iris()
print("入力データ")
print(iris_data.data)
print("¥n")
print("正解データ")
print(iris_data.target)
```

```
入力データ
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]
 [5.4 3.7 1.5 0.2]
```

Pandas

このままだと見にくいためPandasと呼ばれるライブラリを使用する。

Pandasとはデータ解析を容易にする機能を提供するPythonのライブラリ。

Using keywords: Pandas

Find the information from Web to read

AI Academy Media 【初心者向け】データ分析で必須のPandasを入門しよう！
https://aiacademy.jp/media/?p=152（閲覧日：2021/09/30）

Ex1.Pandasを使ったデータの確認

```
from sklearn.datasets import load_iris
import pandas as pd

iris = load_iris()

#全部の行を表示させる
pd.set_option('display.max_rows', None)

#データフレームの作成
data = pd.DataFrame(iris.data, columns=iris.feature_names) #属性
target = pd.DataFrame(iris.target, columns=['花の種類'])  #ラベル

#花の種類の行と特徴量を表す行を結合させた表を表示
pd.concat([data,target], axis=1)
```

Ex1.Pandasを使ったデータの確認

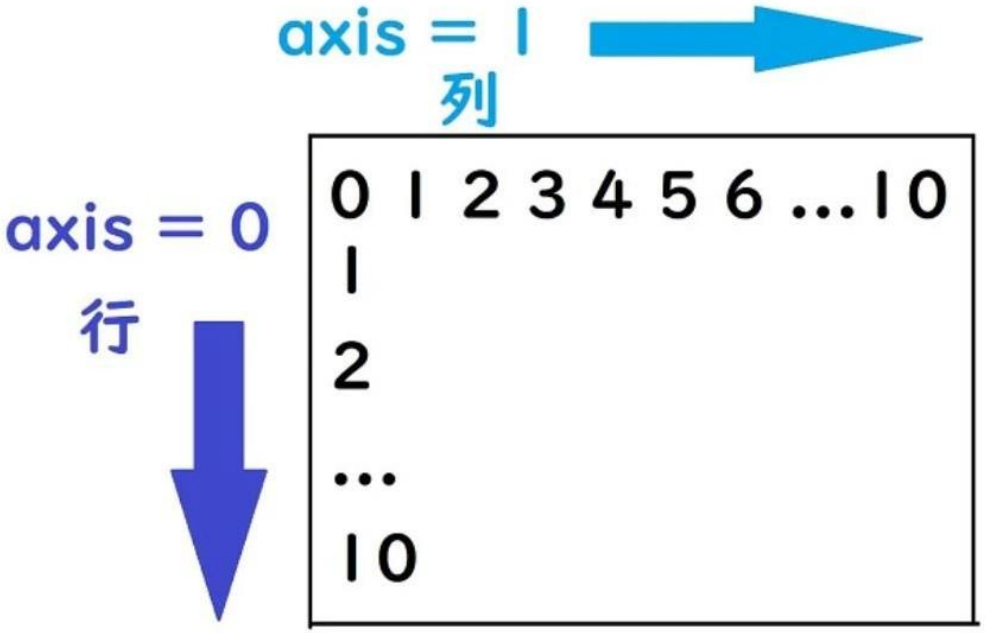
実行すると見やすい表が出てくる。

```
from sklearn.datasets import load_iris
import pandas as pd


iris_data = load_iris()

#全部の行を表示させる
pd.set_option('display.max_rows', None)

data = pd.DataFrame(iris_data.data, columns=iris_data.feature_names)
target = pd.DataFrame(iris_data.target, columns=["花の種類"])

#花の種類の行と特徴量を表す行を結合させた表を表示
pd.concat([data,target], axis=1)
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	花の種類
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
5	5.4	3.9	1.7	0.4	0
6	4.6	3.4	1.4	0.3	0
7	5.0	3.4	1.5	0.2	0
8	4.4	2.9	1.4	0.2	0

補足：axisとは

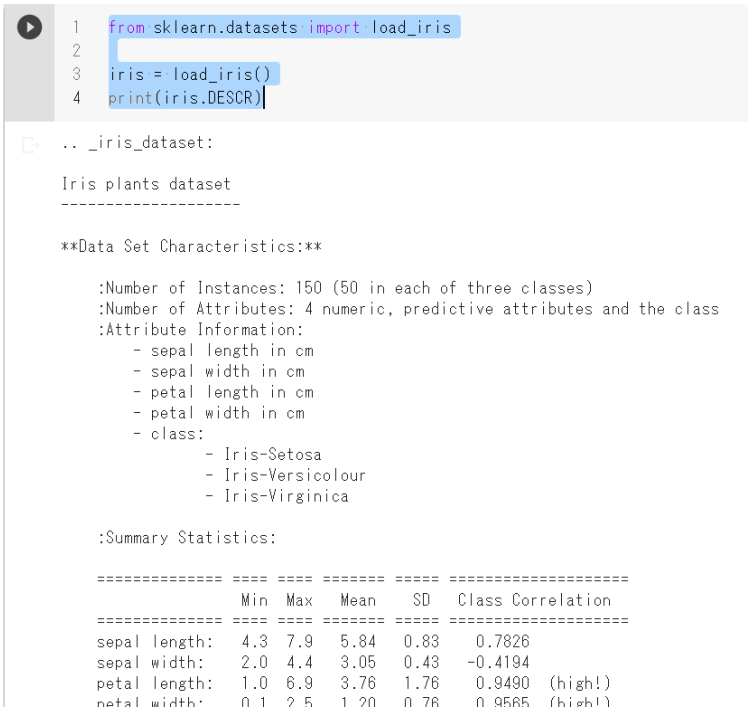
pd.concat([data,target], axis=1)で使用したaxisは行列の軸を指定する引数。axis=0は行、1は列を指定できる。



データセットの詳細

以下のコードを実行するとデータセットについての説明を確認できる。

```
from sklearn.datasets import load_iris

iris = load_iris()
print(iris.DESCR)
```



```
.. _iris_dataset:

Iris plants dataset
--------------------

**Data Set Characteristics:**

    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, predictive attributes and the class
    :Attribute Information:
        - sepal length in cm
        - sepal width in cm
        - petal length in cm
        - petal width in cm
        - class:
                - Iris-Setosa
                - Iris-Versicolour
                - Iris-Virginica

    :Summary Statistics:

    ============== ==== ==== ======= ===== ====================
                    Min  Max   Mean    SD   Class Correlation
    ============== ==== ==== ======= ===== ====================
    sepal length:   4.3  7.9   5.84   0.83    0.7826
    sepal width:    2.0  4.4   3.05   0.43   -0.4194
    petal length:   1.0  6.9   3.76   1.76    0.9490  (high!)
    petal width:    0.1  2.5   1.20   0.76    0.9565  (high!)
```

Ex2.特徴量の可視化

matplotlibと呼ばれるライブラリを用いて特徴量の分布を可視化する。matplotlibはグラフを描画させるためのpythonライブラリ。

※Ex2,3のコードで使用している関数の説明はAppendixに載せています。

Ex2.特徴量の可視化

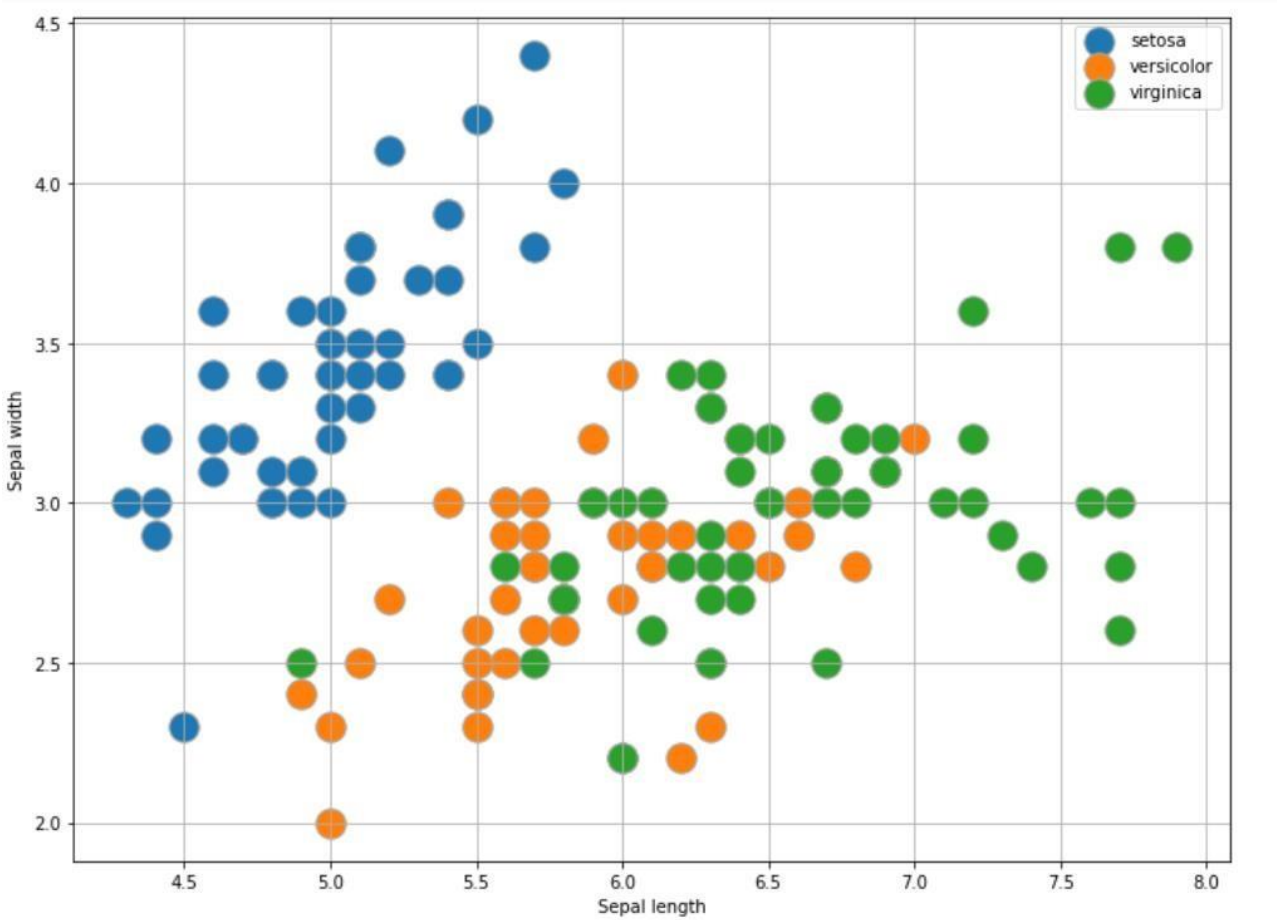
以下のコードを実行してみる（がくの長さと幅の特徴量）。

```
import matplotlib.pyplot as plt

plt.figure(figsize=(12,9))
plt.clf()
class_num=3
class_label=['setosa','versicolor','virginica']

#散布図の描画
def plt_scatter():
  for i_class in range(class_num):
    plt.scatter(
      data[target["花の種類"]==i_class]['sepal length (cm)'],
      data[target["花の種類"]==i_class]['sepal width (cm)'],
      s=300,
      cmap=plt.cm.Set2,
      label=class_label[i_class],
      edgecolor='darkgray'
      )
```

```
#インデントに注意
plt_scatter()
plt.xlabel('Sepal length')
plt.ylabel('Sepal width')
plt.legend()

plt.grid(True)
```

Ex2.特徴量の可視化

Versicolorとvirginicaが混在している。



Ex3.特徴量の可視化

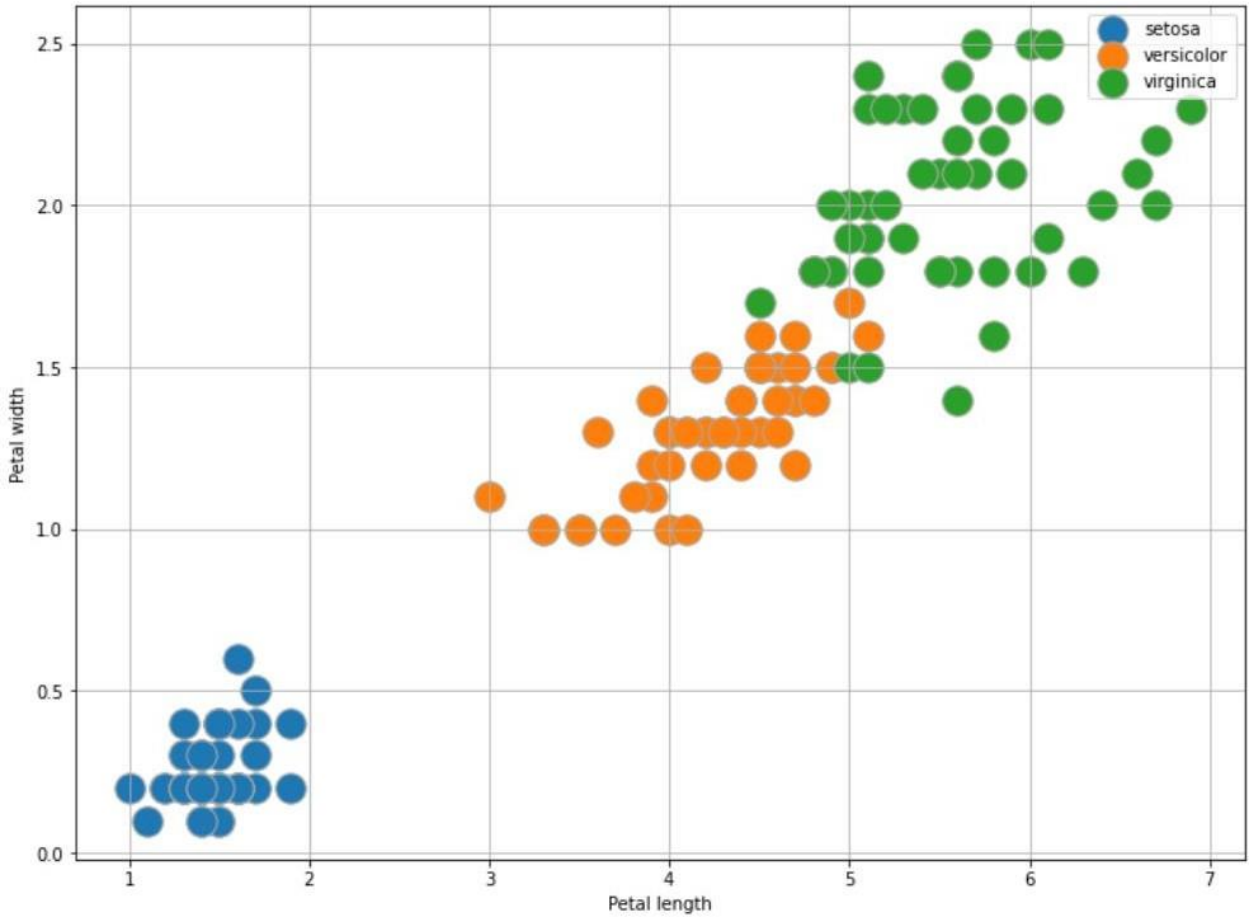
以下のコードを実行してみる（花びらの長さと幅の特徴量）。

```
import matplotlib.pyplot as plt

plt.figure(figsize=(12,9))
plt.clf()
class_num=3
class_label=['setosa','versicolor','virginica']

#散布図の描画
def plt_scatter():
  for i_class in range(class_num):
    plt.scatter(
        data[target["花の種類"]==i_class]['petal length (cm)'],
        data[target["花の種類"]==i_class]['petal width (cm)'],
        s=300,
        cmap=plt.cm.Set2,
        label=class_label[i_class],
        edgecolor='darkgray'
        )
```

```
#インデントに注意
plt_scatter()
plt.xlabel('Petal length')
plt.ylabel('Petal width')
plt.legend()

plt.grid(True)
```

Ex3.特徴量の可視化

花びらの長さと幅の特徴量の方が、3クラス綺麗に分かれているので境界線が引きやすい（分類しやすい）のではないかと推測できる。

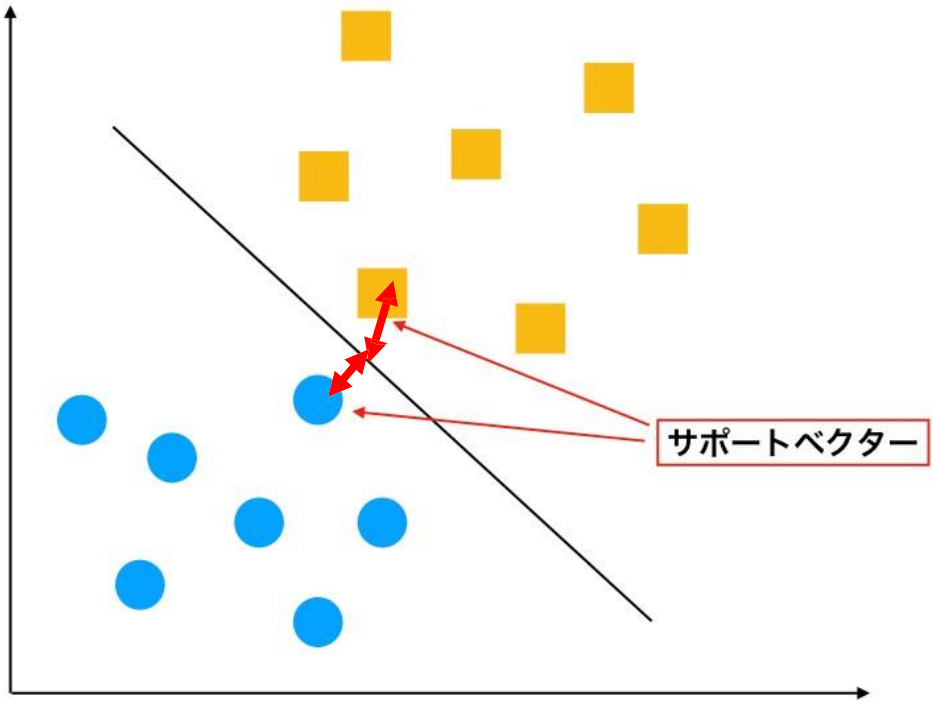


使用する機械学習モデル

今回はサポートベクターマシンと呼ばれる機械学習モデルを使用して、アヤメの識別器を作成する。

サポートベクターマシン(SVM)

機械学習の１つで各クラスのデータ点との距離が最大となるように線を引き、分類を行う。



AI Academy Media　SVM（サポートベクターマシン）とは
https://aiacademy.jp/media/?p=248(閲覧日：2021/09/30)

演習 1

- Ex1～Ex3までを実行してみる。また他の特徴量の組み合わせを可視化してみる（がくの幅と花びらの幅、がくの長さと花びらの長さなど）。

演習 1

Ex2,3で特徴量を指定する場合、 plt.scatterの引数にある以下の**赤字部分**をそれぞれ変更して取り出す。 i_classの部分はクラスラベルの数字(setosaが０、versicolorが１、virginicaが２)が含まれている。

```
plt.scatter(
data[target["花の種類"]==i_class]['sepal length (cm)']
data[target["花の種類"]==i_class]['sepal width (cm)']
…
)
```

■特徴量一覧

特徴量名称	対応する名前
がくの長さ	**'sepal length (cm)'**
がくの幅	**'sepal width (cm)'**
花びらの長さ	**'petal length (cm)'**
花びらの幅	**'petal width (cm)'**

演習を早く終わった人 (optional)

アヤメデータ以外のデータセット（scikit-learnのサンプルデータなど）を調査してみる。

appendix

Ex2.特徴量の可視化

plt.figure(figsize=(12,9))
図の初期設定、figsizeでサイズを指定できる。

plt.clf()
図を初期化する

Ex2.特徴量の可視化

plt.scatter

散布図を描画

引数	説明
x,y	使用するデータ（Ex2ならがくの長さと幅）
s	散布図の出力サイズ
cmap	カラーマップ
label	ラベルの名前
edgecolor	線の色

Ex2.特徴量の可視化

data[target["花の種類"]==i_class]['sepal length (cm)']

dataはpandasのDataframeであるので、data[target[“花の種類”]==i_class]と指定することでi_class(アヤメのクラス)に該当するデータを取得し、['sepal length (cm)']とすることでがくの長さを取得できる。

data[(target==i_class).values]['sepal length (cm)']

i_classのvalues(特徴量)を取得し、さらに'sepal length (cm)'だけ表示。

演習１補足

Iris_dataからも特徴量を取り出せる。特徴量を変えるときは以下の**赤印部分**を変更する。0が「がくの長さ」、1が「がくの幅」、2は「花びらの長さ」、3は「花びらの幅」。

```
#0番目と1番目の特徴量が取り出せる
first_two_features=iris.data[:, [0,1]]
```

```
#2番目と3番目の特徴量が取り出せる
last_two_features=iris.data[:, [2,3]]
```