Pythonが5日でわかる本

パイソン

AI基礎編

中島 省吾（メディアプラネット）著

人気No.1プログラミング言語「Python」で「AI」の基礎を学ぶ新人研修を誌上体験しよう

日経ソフトウエア
NIKKEI SOFTWARE
2020年1月号 第2付録

特別付録② Pythonが5日でわかる本 AI基礎編

日経ソフトウエア
NIKKEI SOFTWARE

1日目

外部ライブラリ

Part 1 はじめに

ここは、とある大手水産会社の本社第七会議室。会社は、魚の養殖事業にAI(人工知能)やIoT(モノのインターネット)を活用したいと考え、水産学部出身の新人、小鯖 進(こさば すすむ)と、理学部生物学科出身の新人、栄井 愛(さかい あい)を「AI人材」に育成するため、研修を行うことになった。2人は初夏に行われた「基本編」で、「Pythonの基本」は習得済みである。今回の研修では、いよいよ「AIプログラミング」に挑戦する…。

進 よう、愛ちゃん。ひさしぶりぃ!

愛 小鯖…、君。

進 いよいよ、AIプログラミングの研修やなぁ。

愛 うん。

講師 はい。そろったようなので、始めましょう。おはようございます。講師の中島です。

進 **愛** おはようございます。

講師 お二人には、前回の研修で「Pythonの基本」を習得してもらいましたが、覚えていますか。

愛 はい…。

進 完璧ですわ。

講師 今回の研修テーマは、「AIプログラミング」です。正確には、Pythonによる機械学習プログラミングを体験します。

愛 AIではない…。

講師 AIと機械学習はよく混同されますが、「AI」の最終目的は、コンピュータによって「人間の脳の働きをシミュレーション」することです。この「AI」を開発するために、大量のデータからパターンを導き出し、現実世界の様々なオブジェクトを認識させる技術が「機械学習」です。

進 Pythonは、機械学習もできるんか。

講師 そうです。機械学習の技術はここ数年で大きく発展していて、Pythonには機械学習を行うための「ライブラリ」が、多数用意されています。

進 ライブラリ…って、Pythonが持っているオブジェクトや関数の集まりのことやろ。

愛 モジュールやパッケージのこと…。

講師 Pythonのライブラリは大きく2種類に分けることができます。1つは、Pythonが標準で持っているモジュールの集まりである「標準ライブラリ」。もう1つは、インターネットのサイトからダウンロードして、インストールしてから使う「外部ライブラリ」です。外部ライブラリは「サードパーティライブラリ」とも呼ばれます。

愛 パーティ…。

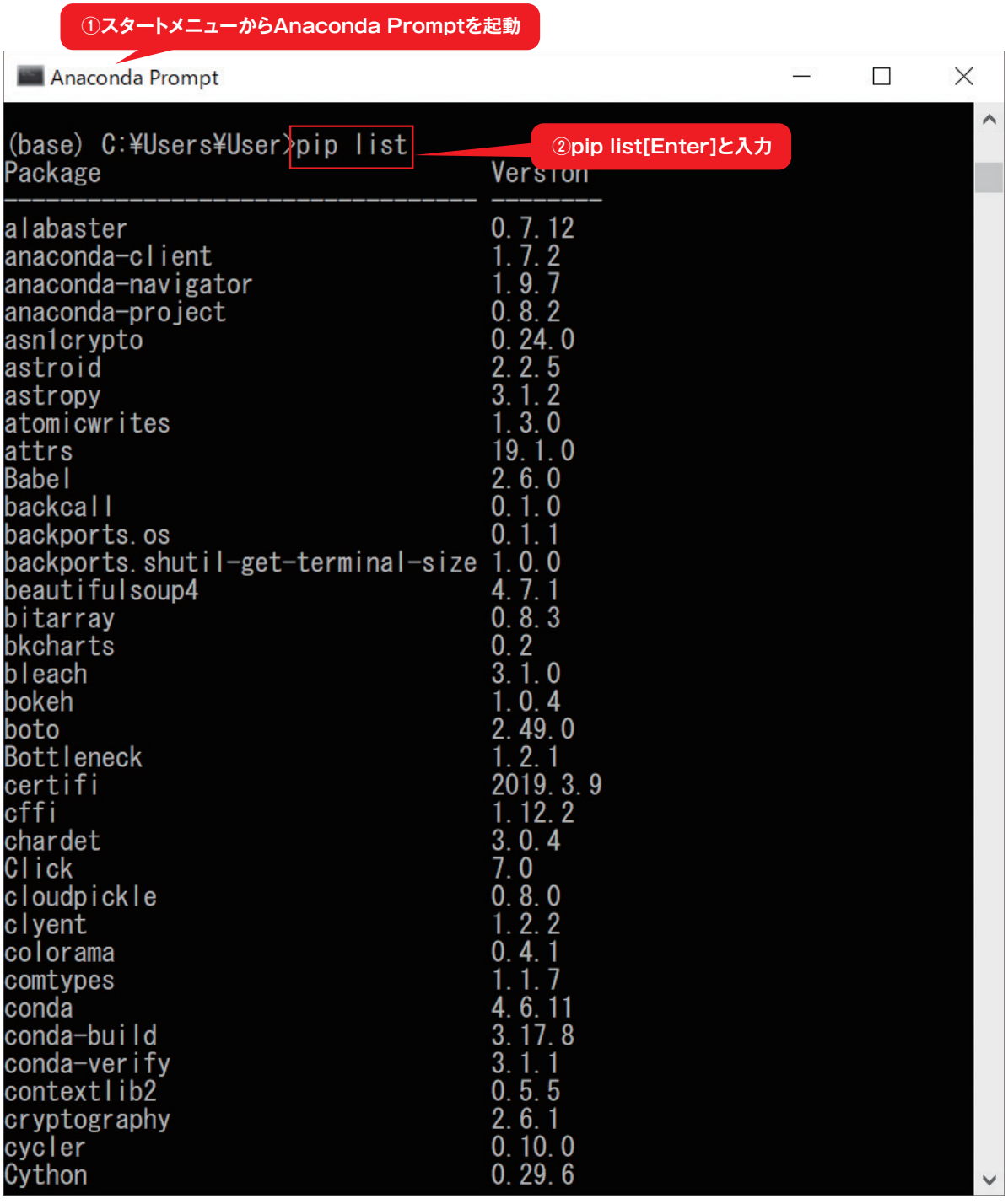
進 なんや、外部ライブラリは面倒やなぁ。

講師 確かに外部ライブラリはパッケージをダウンロードしてインストールする手間があります。ただ、「Anaconda」（アナコンダ）を使うと、そのような手間はだいぶ軽減されます。Anacondaには、有名なモジュールやパッケージがはじめから多数同梱されているので、importするだけでそれらを利用できます。

進 おお、すげー!

講師 お二人のパソコンには、Anacondaがインストール済みですので、スタートメニューからAnaconda Promptを起動して、listオプションを付けてpipコマンドを実行しましょう。利用できるモジュールやパッケージが一覧表示されますよ（**図1**）。

図1●pipが管理しているパッケージ一覧



進 なん…、スゴイ量じゃあ。

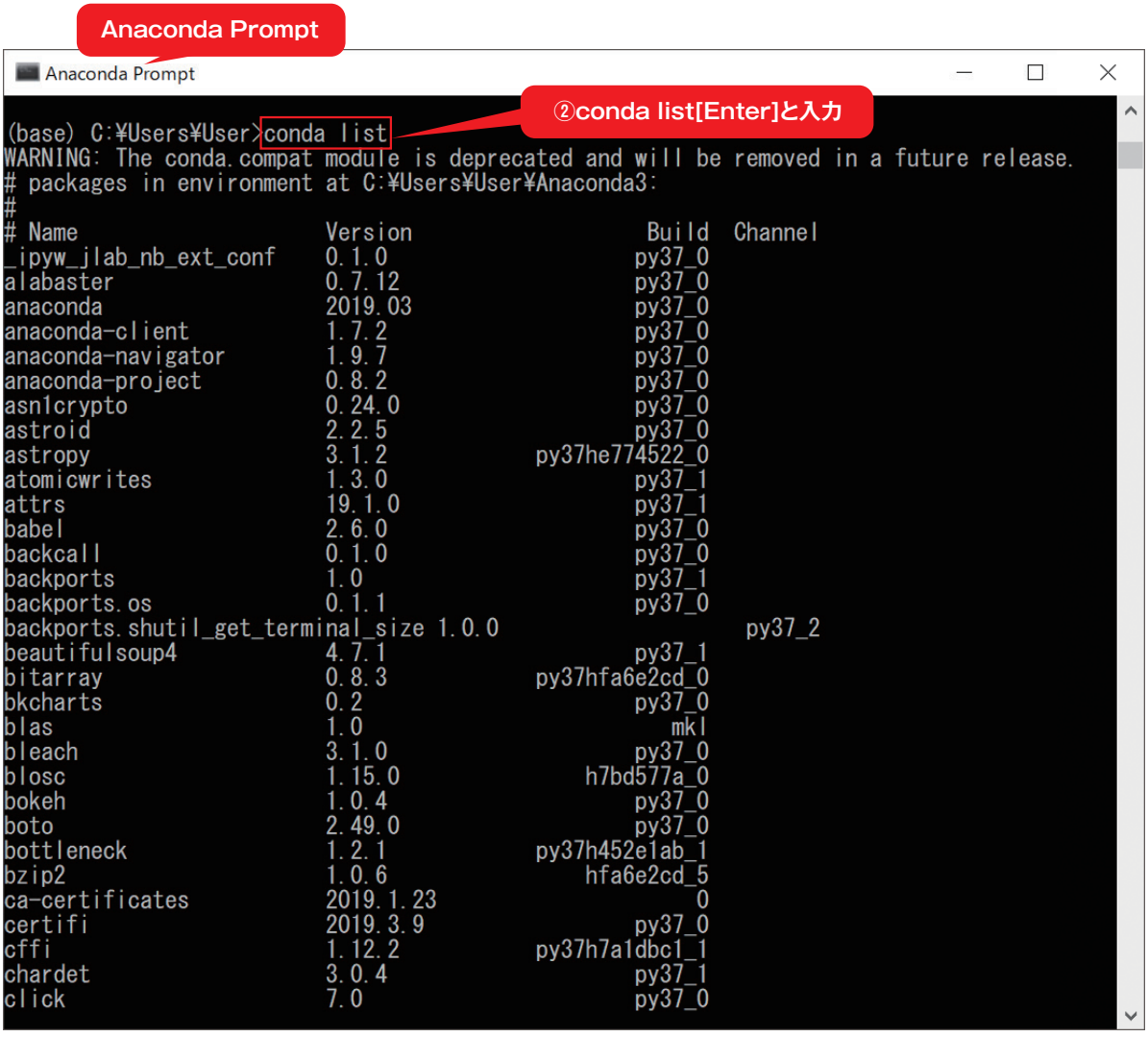
愛 pip…えらいね…。

講師 pipは、Pythonのパッケージをダウンロードしてインストールしたり、管理するためのプログラムです。listオプションを使えば、pipが管理している外部ライブラリを調べることができます。

進 pip自体も、パッケージなんじゃのう。

講師 この中に、condaというプログラムもあります。こちらはAnacondaが同梱している独立したPythonの環境管理プログラムです。Anacondaはcondaでライブラリを管理しています。conda listと入力すると管理しているライブラリの一覧が表示されます（**図2**）。ちなみに、もし、ほかの外部ライブラリを別途インストールしたいときは、pipやcondaコマンドにinstallオプションを付けてインストールします。

図2●condaが管理しているパッケージ一覧



愛 condaに、警告が出てる。

講師 そうですね。しばらくすると別のプログラムに置きかわるのかも知れません。それでは、この一覧の中にある「numpyパッケージ」と「python-dateutilモジュール」の使い方を確認しましょう。

Part 2 NumPy

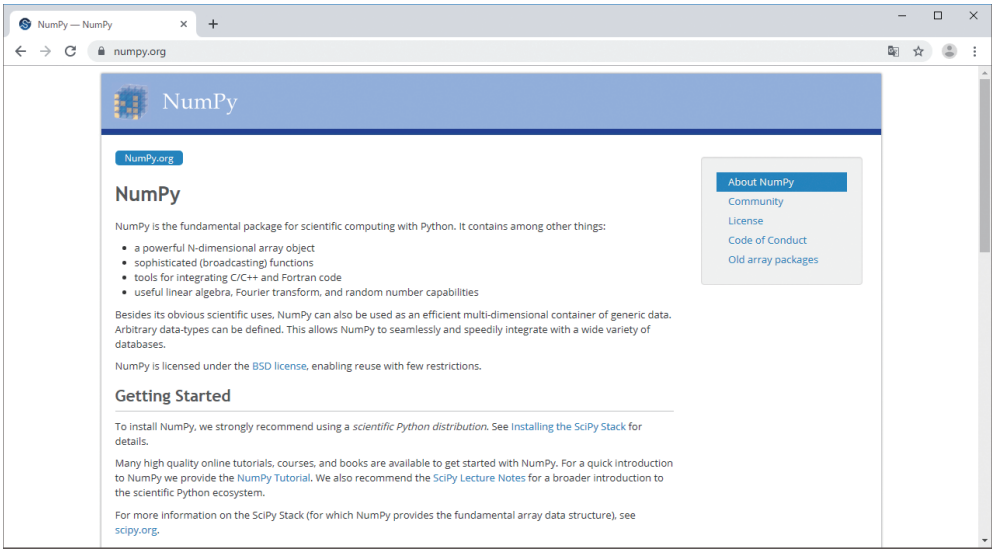
講師 最初の外部ライブラリとして、NumPyを紹介します。NumPyは、数値計算を効率的に行うためのモジュールです。AIプログラミングでは様々な関数を利用して計算を行いますが、NumPyには便利な関数が多数含まれています。NumPyは外部ライブラリではありますが、事実上の標準ライブラリのような存在になっています。

愛 パッケージ一覧の中に、NumPyもいる…。

講師 このAnacondaには、NumPyのバージョン1.16.2がインストールされているようですね。NumPyについて詳しく知りたいときは、**図1**の公式サイトを参照すると良いでしょう。

図1●NumPyの公式サイト

https://numpy.org/



講師 では、NumPyで配列を生成してみましょう。Anacondaに付属するSpyderを起動したら、エディターに次のコードを入力します。このコードは、NumPyのarray関数を使ってlistから配列を生成しています。

sample01.py

NumPyをインポートしてnpというオブジェクト名で利用できるようにする

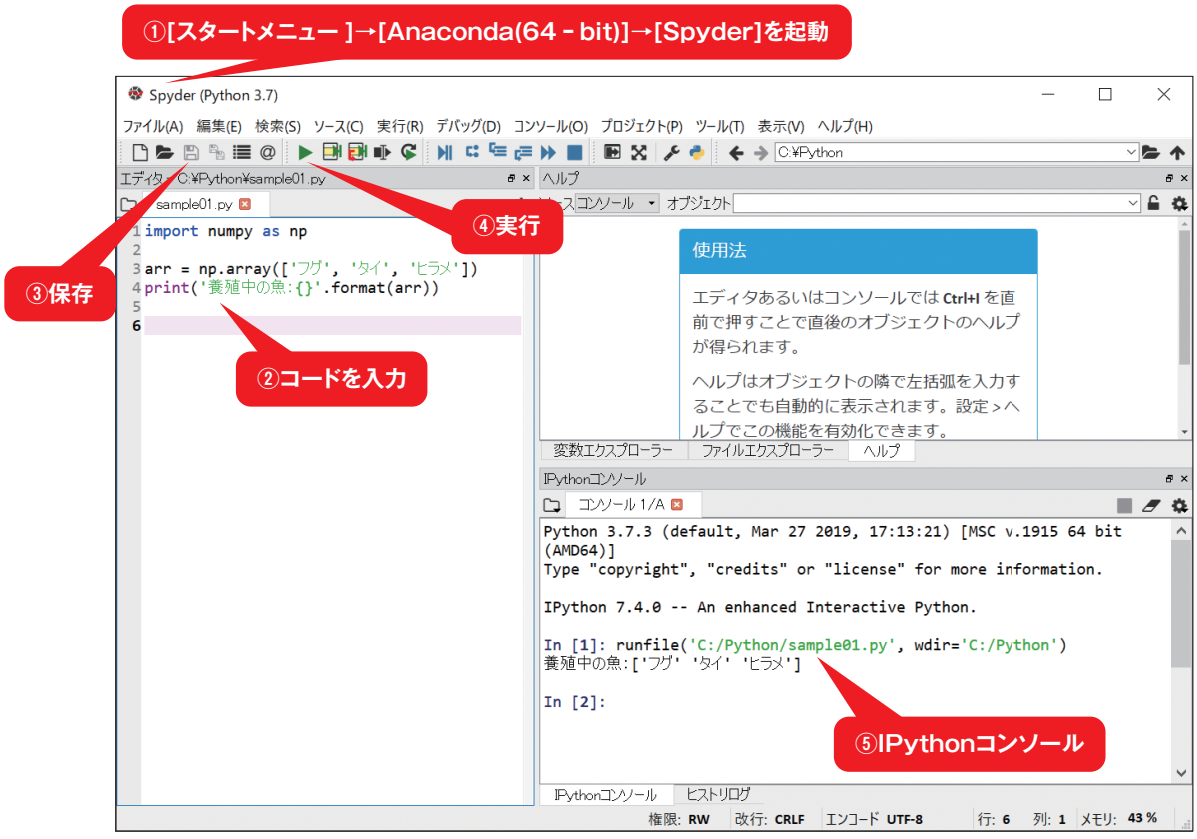
```
import numpy as np

arr = np.array(['フグ', 'タイ', 'ヒラメ'])
print('養殖中の魚:{}'.format(arr))
```

NumPyのarray関数でNumPyの配列を生成

NumPyの配列を表示

講師 入力できたら、実行します（図2）。NumPyの配列の中身を表示できると思います。

図2●Spyderの起動とコード入力、実行



進 なんや、listオブジェクトと変わらん気がするのう。

講師 確かにNumPyの配列は、listによく似たオブジェクトです。ただ、listよりも高速に動作するという利点があるので、AIプログラミングではNumPyが多用されます。次のコードは、多次元配列の生成とその次元数を表示するコードです。

sample02.py

```
import numpy as np

arr = np.array([['フグ', 'タイ', 'ヒラメ'], [380, 450, 70]])
print('養殖魚の個体数')
print(arr)
print('') #改行

print('各次元数の表示:{}'.format(arr.shape))
```

array関数でNumPyの2次元配列を生成

shape属性で配列の各次元ごとの要素数を取得

IPythonコンソール

```
runfile('C:/Python/sample02.py', wdir='C:/Python')
養殖魚の個体数
[['フグ' 'タイ' 'ヒラメ']
 ['380' '450' '70']]

各次元数の表示:(2, 3)
```

IPythonコンソール

各次元の要素数

愛 多次元って、いいよね。

進 愛ちゃん…変わってないのう。

講師 NumPyの本領が発揮されるのは、配列操作の機能です。転置も簡単に求めることができます。

sample03.py

```
import numpy as np

arr = np.array([1, 2, 3])
arr = arr * 3
print('[1, 2, 3] * 3:{}'.format(arr))

arr = np.array([[1, 2, 3], [2, 3, 4]])
print('') # 改行
print('転置前')
print(arr)
print('') # 改行
print('転置後')
print(arr.T)
```

NumPyの配列に3を掛ける

T属性に転置後の値が格納されている

IPythonコンソール

```
runfile('C:/Python/sample03.py', wdir='C:/Python')
[1, 2, 3] * 3:[3 6 9]

転置前
[[1 2 3]
 [2 3 4]]

転置後
[[1 2]
 [2 3]
 [3 4]]
```

要素がすべて3倍になっている

[[1, 2, 3], [2, 3, 4]]の転置後の値

進 ほう。こりゃ便利じゃ。

講師 さらに、同じ形の配列同士の計算は、同じ場所の要素同士が計算されて返ります。また、2つの配列を行列と見なして行列の積を求めることができます。この行列の積の計算にはdot関数を使います（**表1**）。dot関数で、ベクトルの内積も計算できます。

表1●dot関数の仕様

numpy.dot(a, b, out=None)	
引数	説明
a	左からかけるベクトルまたは行列
b	右からかけるベクトルまたは行列
out	結果を格納する代替配列

戻り値	ベクトルの内積の結果や、行列の積の結果

sample04.py

```
import numpy as np

arr_1 = np.array([1, 2, 3])
arr_2 = np.array([2, 3, 4])
print(np.dot(arr_1, arr_2))
print('') # 改行

arr_1 = np.array([[1, 2], [3, 4]])
arr_2 = np.array([[5, 6], [7, 8]])
print('[[1, 2], [3, 4]]と[[5, 6], [7, 8]]の行列の積')
print(np.dot(arr_1, arr_2))
```

ベクトルの内積

行列の積

IPythonコンソール

```
runfile('C:/Python/sample04.py', wdir='C:/Python')
20

[[1, 2], [3, 4]]と[[5, 6], [7, 8]]の行列の積
[[19 22]
 [43 50]]
```

[1, 2, 3]と[2, 3, 4]のベクトルの内積「1×2＋2×3＋3×4」

[[1, 2], [3, 4]]と[[5, 6], [7, 8]]の行列の積
「[1×5＋2×7, 1×6＋2×8],
[3×5＋4×7, 3×6＋4×8]」

講師 ベクトルの内積とは、各要素の積をすべて足し合わせた値です。行列の積では、横の並び、縦の並びの組の同じ順番の数同士を掛けたものを足します。

進 つまり、arr1とarr2のベクトルの内積は、1×2 + 2×3 + 3×4で20。arr1とarr2の行列の積は、[1×5 + 2×7, 1×6 + 2×8], [3×5 + 4×7, 3×6 + 4×8]じゃな。

講師 NumPyには、統計関数もあります。例えば、配列の平均を求めるには、mean関数を使うと簡単です（**表2**）。mean関数には配列の要素をすべて渡します。

表2●mean関数の仕様

numpy.mean(a, axis=None, dtype=None, out=None, keepdims=False)	
引数	**説明**
a	平均を求めたい配列
axis	どの軸(axis) に沿って平均を求めるか
dtype	計算結果を格納するための配列
out	結果を格納する代替配列
keepdims	返す配列の軸(axis) の数をそのままにする
戻り値	指定した配列の要素の平均、もしくは平均を要素とする配列

sample05.py

```
import numpy as np

r = np.array([9, 1, 1, 5, 8, 0, 4, 8, 7, 6])
m = np.mean(r)
print(m)
```

配列要素の平均値を計算

IPythonコンソール

```
runfile('C:/Python/sample05.py', wdir='C:/Python')
4.9
```

配列要素の平均値

講師 さらに、標準偏差を求めるstd関数などもあります（**表3**）。

表3●std関数の仕様

numpy.std(a, axis=None, dtype=None, out=None, ddof=0, keepdims=<class numpy._globals._NoValue>)	
引数	説明
a	標準偏差を計算したい配列
axis	どの軸(axis) に沿って平均を求めるか
dtype	計算結果を格納するための配列
out	結果を格納する配列
ddof	データ個数 N ではなく "N - ddof" で割る
keepdims	True にすると出力される配列の次元数が保存される

戻り値	指定された範囲での標準偏差を要素とする配列、または値

sample06.py

```
import numpy as np

r = np.array([9, 1, 1, 5, 8, 0, 4, 8, 7, 6])
s = np.std(r)
print(s)
```

標準偏差の計算

IPythonコンソール

```
runfile('C:/Python/sample06.py', wdir='C:/Python')
3.1128764832546763
```

配列要素の標準偏差値

進 NumPy、恐るべし。NumPyは、数学関数の宝石箱やぁ。

講師 まだまだ紹介したい関数やオブジェクトはあるのですが、続きは機械学習でやるとしましょう。それでは少し休憩を入れましょう。

1日目 Part 3 python-dateutil

講師 次に紹介する外部ライブラリは、python-dateutilです。Pythonは標準ライブラリで、timeオブジェクト、datetimeオブジェクト、calendarオブジェクトなどを用意しています。これらを使うと、日付や時刻を簡単に扱うことができます。

進 datetimeオブジェクトなら、知っとるで。例えば、今日の日付と時刻を表示するんなら、こうじゃ。

sample07.py

```
import datetime as dt

now = dt.datetime.today()
print('今日の日付:{}'.format(now.date()))
print('現在時刻:{}'.format(now.time()))
```

datetimeをインポートしてdtで利用できるようにする

現在の日時と時刻を保持するdateの取得

日付の取得

時刻の取得

IPythonコンソール

```
runfile('C:/Python/sample07.py', wdir='C:/Python')
今日の日付:2019-09-10
現在時刻:04:56:42.080493
```

講師 datetimeなら、今日の日付と時刻を同時に取得できて便利ですよね（**表1**）。datetimeはtimedeltaオブジェクトを生成できるので、日付の計算も簡単ですね。

愛 できた…。

datetimeをインポートしてdtで利用できるようにする

sample08.py

```
import datetime as dt

sample_date = dt.date(2019,10,24)
sample_timedelta = dt.timedelta(days = 10)
later = sample_date + sample_timedelta
print('2019年10月24日の10日後:{}'.format(later))
```

指定日のdatetime.dateオブジェクトを取得

10日間のtimedeltaを取得

IPythonコンソール

```
runfile('C:/Python/sample08.py', wdir='C:/Python')
2019年10月24日の10日後:2019-11-03
```

表1●使用したdatetimeのメソッドとオブジェクトの仕様

名前	種類	説明
today()	メソッド	現在のローカルな datetime オブジェクトを返す
date()	メソッド	datetime オブジェクト内から、年、月、日を date オブジェクトで返す
time()	メソッド	datetime オブジェクト内から、時、分、秒、マイクロ秒を持つ time オブジェクトを返す
timedelta(days=0, seconds=0, microseconds=0, milliseconds=0, minutes=0, hours=0, weeks=0)	オブジェクト（コンストラクタ）	指定した「経過時間」を表す timedelta オブジェクトを生成する。すべての引数が省略可能。デフォルト値は0。引数は整数、浮動小数点数共に可能、正、負どちらでも計算可能

進 なんや、愛ちゃんも使えるんかい。

講師 では、2020年2月の月末を知りたい時はどうすればよいでしょう。

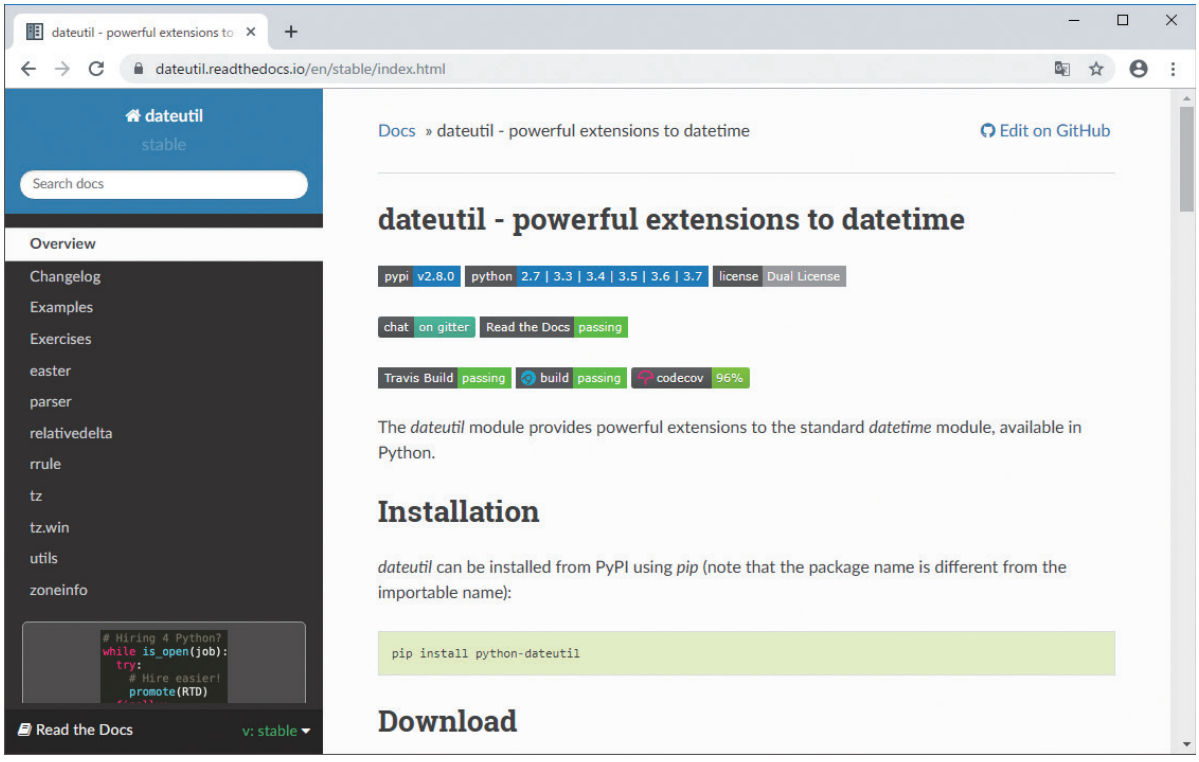
愛 2020年が閏（うるう）年かどうか調べる…。閏年なら29日。そうでなければ28日。

講師 なるほど。では、指定した年と月の月末を調べたいときはどうしますか。

進 閏年かどうかを調べるだけじゃ駄目じゃな。月末を調べるメソッドを作るかのう。

講師 そうなりますね。このようなときは外部ライブラリのdateutilモジュールを使うと便利です（**図1**）。Anacondaではpython-dateutilとしてインストールされています。

図1●dateutilのドキュメントサイト
https://dateutil.readthedocs.io/en/stable/index.html



進 自分で作る前に、外部ライブラリを探す方が楽ってことじゃ。

講師 python-dateutilには、relativedeltaというオブジェクトがあります（**表2**）。このオブジェクトは、datetimeのtimedeltaオブジェクトより強力です。

表2●dateutilモジュールのrelativedeltaの仕様

名前	種類	説明
relativedelta(dt1=None, dt2=None, years=0, months=0, days=0, leapdays=0, weeks=0, hours=0, minutes=0, seconds=0, microseconds=0, year=None, month=None, day=None, weekday=None, yearday=None, nlyearday=None, hour=None, minute=None, second=None, microsecond=None)	オブジェクト（コンストラクタ）	引数に特定の日時を表すdatetimeもしくは日時を指定して、そのrelativedeltaオブジェクトを生成する

講師 先ほどの「2020年2月の月末」を知りたい時は、次のコードで調べることができます。

sample09.py

```
import datetime as dt
from dateutil.relativedelta import relativedelta

day = dt.date(2020, 2, 1) + relativedelta(day=99)
print('2020年2月の月末は{}日です。'.format(day.day))
```


IPythonコンソール

```
runfile('C:/Python/sample09.py', wdir='C:/Python')
2020年2月の月末は29日です。
```

進 月末に99日なんてないから、自動的に本当の月末日が代入されるっちゅうわけや。

講師 今日は、ここまでにしましょう。明日はWebから情報を取り出すWebスクレイピングを紹介します。

2日目

Webから情報を取得する

Part 1 Webから取得できる情報

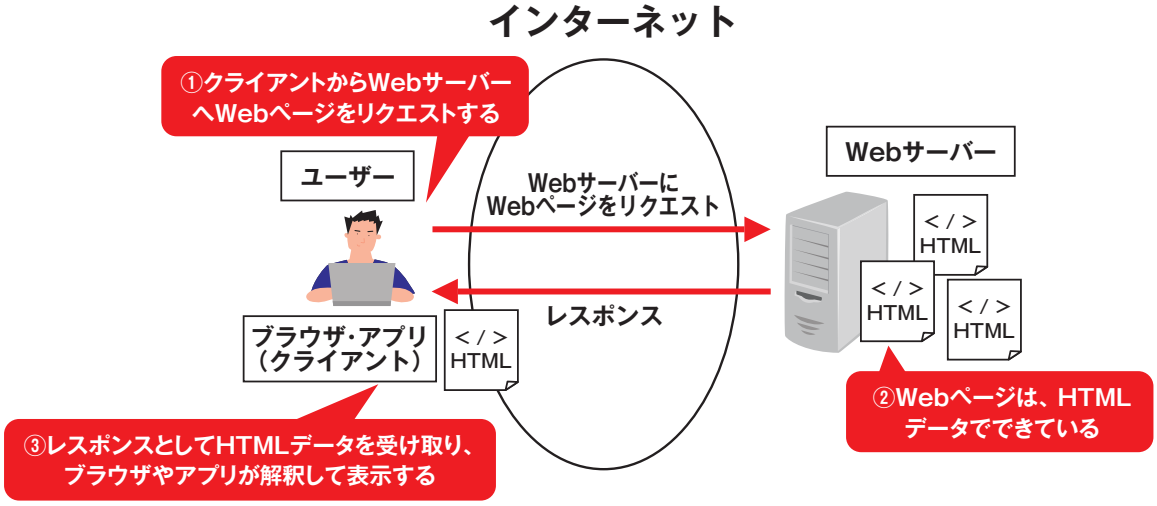
講師 今日は、Webサーバーが提供する情報を、Pythonで取得してみます。ところでお二人は、Webでどのようなサイトをよく見ますか。

進 そやな、常にチェックするのはお天気サイトじゃな。海での仕事が多いよって。

愛 波、風、海水温、満潮、干潮…。

講師 なるほど。ブラウザやスマホアプリで、お天気や海洋情報を調べてるんですね。こういったWebページは、インターネットに接続されたWebサーバーが提供しています。**図1**は、パソコンやスマホアプリからサイトを閲覧するときのデータの流れを簡単に解説した図です。

図1●Webサイト閲覧の仕組み



講師 Webサーバーは、ブラウザやアプリといったクライアントからのリクエストに応じて、HTMLデータをレスポンスとして送信します。クライアントとWebサーバーは、通信規約「HTTPプロトコル」で通信をしています。

愛 合い言葉は、HTTP…。

講師 例えば、ブラウザで天気予報のサイトにアクセスすると、Webサーバーは要求されたWebページをブラウザにレスポンスとして送信します。WebページはHTMLデータでできているので、このHTMLをブラウザが解釈して、お天気情報などのページを表示します。

進 この仕組みはよう知っとる。HTMLの研修は受けとるでな。

講師 この方法の問題は、人間が毎回サイトにアクセスしないと欲しい情報が得られない点ですが、Pythonなら、クライアントプログラムからWebサーバーに自動アクセスして、必要な情報だけを取得することができるんです。

愛 Python…ステキ…。

進 じゃが、送信されてくるHTMLデータは「お天気情報」だけじゃないじゃろう。どうやってお天気だけ取り出すんじゃ。

講師 HTMLデータから必要な情報を検索する方法はいろいろありますが、今回は文字列検索で取り出してみましょう。

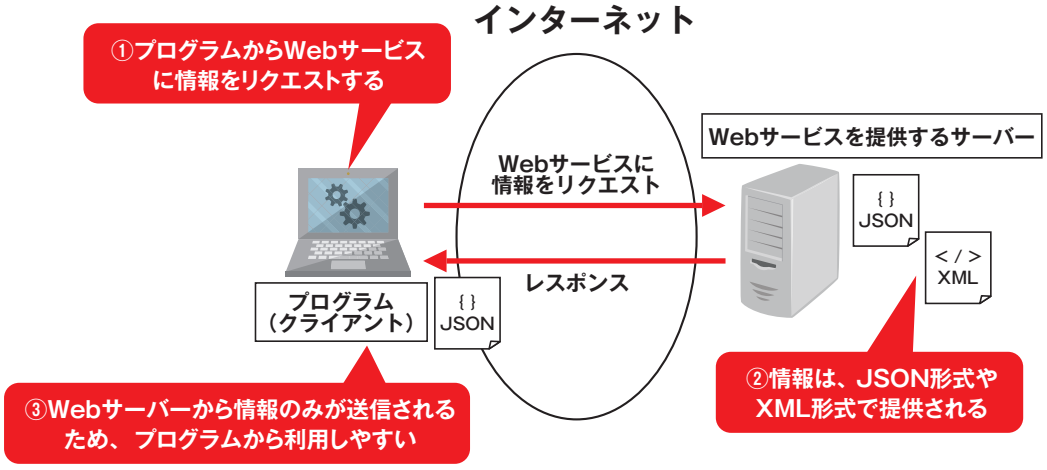
愛 正規表現…＜キラッ＞。

講師 そう、HTMLデータはテキストなので、正規表現で文字列パターンを検索することができます。

進 正規表現…聞いたことないのう。じゃったら、お天気データだけ送信してくれるサーバーがあったらええのに…。

講師 ありますよ。このようなサーバー機能を「Webサービス」と呼びます（**図2**）。Webサービスを提供してくれるサーバーなら、必要な情報だけをプログラムから取得できるので利用しやすいですね。

図2●Webサービス



進 なるほど…。このサービスは、どんなWebサイトでも利用できるんかのう。

講師 残念ながら、Webサービスを提供しているWebサーバーでなければ利用できません。それ以外のサイトは、送信されてきたHTMLデータの中から、必要な情報を検索しなければいけません。

進 う〜ん、やっぱり「正規なんとか」ってのを、勉強せなあかんのか…。

講師 そうですね。このようにHTMLデータから必要な情報だけを取り出すプログラムは「Webスクレイピングプログラム」とか、Webサイトを巡回して情報収集するので「Webクローラー」などと呼ばれます。では、正規表現を使ったWebスクレイピングから試してみましょう。

Part 2 Webスクレイピング

講師 Webスクレイピングを行うには、PythonプログラムからWebサーバーにリクエストを送信して、HTMLデータを取得できなければいけません。ここでは例として、**図1**の天気情報ページを、Pythonで取得してみましょう。

図1●日本経済新聞のお天気ページ

http://weather.nikkei.com/yoho/

進 おっ、ふるさとの岡山は、曇り時々晴れじゃ。

講師 小鯖さんは、岡山出身なんですね。

愛 私は、神奈川県の箱根…。

進 ほう、愛ちゃんは神奈川か。よう知らんかった。

講師 はい。それではPythonでWebページを取得してみましょう。利用するライブラリは、標準で搭載するurllibパッケージのrequestモジュールです。このモジュールのurlopen関数でWebサーバーにリクエストを送信します（**表1**）。

表1●urlopen関数の仕様（抜粋）

urllib.request.urlopen(url)	
引数	説明
url	オープンする URL 文字列または Request オブジェクト

講師 Spyderを起動して、次のコードを入力しましょう。

urllibパッケージのrequestモジュールをインポート

sample01.py

```
import urllib.request

res = urllib.request.urlopen('http://weather.nikkei.com/yoho/')
html = res.read().decode('utf-8')
res.close()

print(html)
```

urlopen関数でリクエストを送信する

日経新聞の天気情報ページ

取得したレスポンスをテキストに変換

表示

IPythonコンソール

```
Python 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915
64 bit (AMD64)]
Type "copyright", "credits" or "license" for more informat
ion.

IPython 7.4.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Python/sample01.py', wdir='C:/Python')
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional/
/EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.
dtd">
<html>
<head>
<meta http-equiv="Content-Language" content="ja">

...
```

実行

受信したHTML

講師 urlopen関数は、Webサイトにリクエストを送信し、アクセスに成功するとhttp.client.HTTPResponseオブジェクトを返します。このオブジェクトからHTMLデータを取り出すには、readメソッドを呼び出します。ただし、このデータはバイト単位なので、decode関数でUTF-8へ変換します。

進 うおっ、すごい量のHTMLデータが表示されたぞい。

講師 もし、HTMLデータをファイルに保存したいのなら、open関数でファイルを開き、write関数で書き込みましょう。open関数の仕様は**表2**や**表3**のようになります（**図2**）。

表2●open関数の仕様（抜粋）

open(file, mode='r', encoding=None)	
引数	説明
file	開くファイルの名前
mode	オープンモードの指定（デフォルトは読み取り専用）
encoding	ファイルのエンコードやデコードの方式
戻り値	開くことができたファイルオブジェクト

戻り値	開くことができたファイルオブジェクト

表3●open関数のモード

文字	意味
'r'	読み込み専用で開く。書き込みはできない（'rt' がデフォルト）。 ファイルがないときはエラー
'w'	書き込み専用で開く。読み込みはできない。ファイルがないときは、新規作成する
'x'	書き込み専用で開くが、既存ファイルがある場合はエラー
'a'	書き込み専用で開く。ファイルがないときは新規作成、既存ファイルがある場合は末尾に追加
'b'	バイナリモードで開く
't'	テキストモードで開く（'rt' がデフォルト）
'+'	更新可能にする。'r+' の場合、読み書き可能、ファイルがないときはエラー。 'w+' の場合、読み書き可能、ファイルがないときは新規に作成

sample02.py

```
import urllib.request

res = urllib.request.urlopen('http://weather.nikkei.com/yoho/')
html = res.read().decode('utf-8')
res.close()

f = open('weather.html', 'w', encoding='utf-8')
f.write(html)
f.close()
print('weather.htmlに保存しました。')
```

書き込みモードでファイルを開く

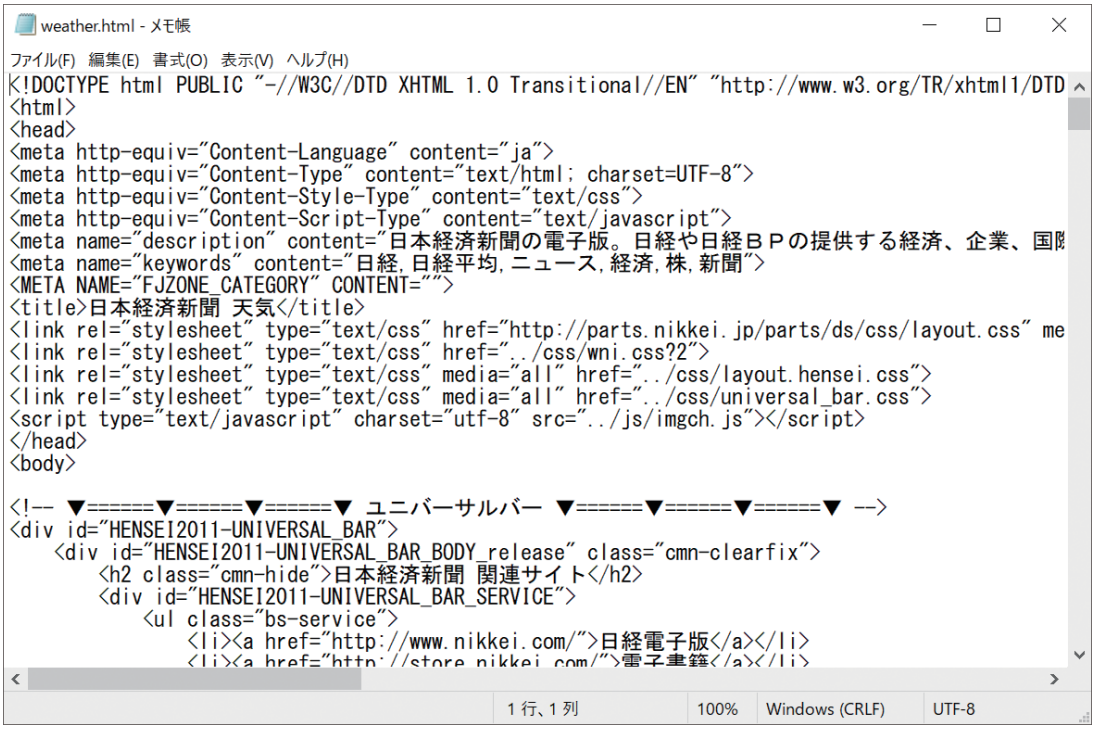
取得したHTMLデータをファイルに保存

IPythonコンソール

```
runfile('C:/Python/sample02.py', wdir='C:/Python')
weather.htmlに保存しました。
```

成功

図2●エディターで表示したHTML



進 このHTMLデータから、調べたい土地のお天気を調べるんじゃな。エディターを使えば「岡山」ですぐ検索して見つかるぞい。

「岡山」で検索

```
・・・
<td align="center" bgcolor="#dbf1ff" class="fs10constant">
岡山</td>
                <td bgcolor="#FFFFFF" class="fs10constant"
align="center"><img src="../img/telop/200.gif" alt="曇り時々
晴れ"></td>
・・・
```

岡山の今日の天気

講師 このような文字列検索は「正規表現」を利用すると簡単です。正規表現では文字とメタ文字と呼ばれる記号を組み合わせてパターンを作り、そのパターンと一致する文字列を検索することができます（**表4**）。

表4●正規表現の主なメタ文字

メタ文字	意味	例
.	（改行以外の）任意の1文字	a.c → abc a3c azc など
^	先頭	^ab → abc ab098 abbbb など
$	末尾	$ab → 123ab xyzab 8u7yab など
*	ないか、1個以上連続	ab*c → ac abc abbbbc など
+	1個以上連続	ab+c → abc abbc abbbbbc など
?	ないか、1つだけ	ab?c → ac abc のみ
{n}	nの繰り返し	ab{3} → abbb のみ
|	いずれかの文字列	abc | xyz | 012 → abc か xyz か 012 のいずれか
[]	指定した文字のどれか	a[xyz]b → axb ayb azb のいずれか [0-9] 0～9のいずれか [d-z] d～zのいずれか

（表4の続き）

メタ文字	意味	例
()	グループ化	a(bc)*d → ad abcd abcbcd abcbcbcd など a(b|c)d → abd acd のいずれか
¥d	アラビア数字	[0-9] と同じ
¥w	アルファベットまたはアンダーバー	A ～Z、a ～z、_ のいずれか [A-Za-z0-9_] と同じ

講師 正規表現のことを「regular expression」と言います。Pythonにはregular expressionの略である「re」モジュールが標準搭載されているので、取得したHTMLからマッチする文字列を見つけ出すことができます。reモジュールのfindallメソッドで、岡山のHTMLタグを取り出してみましょう

sample03.py

```
import urllib.request
import re

res = urllib.request.urlopen('http://weather.nikkei.com/yoho/')
html = res.read().decode('utf-8')
res.close()

match = re.findall('<td align="center" bgcolor="#dbf1ff" class="fs10constant">岡山</td>(.*?)</td>', html,re.DOTALL)
print(match[0])
```

岡山の天気を表示しているタグ

すべての文字列に一致

改行も含めて検索

findallの結果はリスト

IPythonコンソール

```
runfile('C:/Python/sample03.py', wdir='C:/Python')

                          <td bgcolor="#FFFFFF" class="fs10c
onstant" align="center"><img src="../img/telop/200.gif" al
t="曇り時々晴れ">
```

HTMLから抽出したタグ

愛 抽出完了!

講師 このコードでは、「<td align="center" bgcolor="#dbf1ff" class="fs10constant">岡山</td>」で始まり「</td>」で終わる文字列を取り出しています。「(.*?)」の部分は、何らかの連続した文字列を表します。

進 なるほど、これなら岡山のタグだけを取り出せるのう。ここから、天気を切り取るんじゃな。

講師 文字列の切り出しも、findallメソッドが使えます。

sample04.py

```
import urllib.request
import re

res = urllib.request.urlopen('http://weather.nikkei.com/
yoho/')
html = res.read().decode('utf-8')
res.close()

match = re.findall('<td align="center" bgcolor="#dbf1ff" cla
ss="fs10constant">岡山</td>(.*?)</td>', html,re.DOTALL)
yoho = re.findall('alt="(.*)"', match[0])
print('今日の岡山は',yoho[0])
```

抽出した文字列の中からalt属性の文字列のみ取り出す

IPythonコンソール

```
runfile('C:/Python/sample04.py', wdir='C:/Python')
今日の岡山は  曇り時々晴れ
```

進 こりゃええぞい。ブラウザを使わんでも、岡山の天気をゲットできる…が、う〜ん…。

講師 どうしました。

進 じゃが…、HTMLのタグが変更されたらどうするんじゃ。また、調べ直さんといかん。

愛 …HTMLだもの。

講師 そうですね。こればかりは、Webスクレイピングの宿命ってやつです。でも、情報だけを提供してくれるサーバーがあったことを思い出してください。

進 **愛** Webサービス!

講師 そう。次は、Webサービスを利用してみましょう。

Part 3 Webサービス

講師 では、PythonでWebサービスを利用してみましょう。今回はlivedoorが提供する「Weather Hacks」を使い、天気情報を取得します（図1）。

図1●Weather Hacksのページ
http://weather.livedoor.com/weather_hacks/

進 いろんな方法で情報が取り出せそうじゃのう。

愛 プラグイン、XML、JSON…。

講師 Weather Hacksは、ブログなどに貼り付けるだけでお天気を表示するプラグインや、XML、JSON形式でお天気情報を送信してくれるサービスを提供しているので、Pythonプログラマでなくても利用することができるんですよ。

進 XMLってのは、タグでデータを表現するデータ形式じゃろう。これは知っとるが、ジェイソン…てのは、知らんのう。

講師 JSON、「JavaScript Object Notation」は、JavaScriptなどで利用されるデータ交換用のフォーマットです。Weather Hacksでは、HTTPプロトコルでデータを送受信するので、取得方法はHTMLと同じです。JSONで表現できるデータの種類を紹介しておきましょう（**表1**）。

表1●JSONのデータ表現

データ	記述例
文字列値(string)	“abc” や ”def” 等、ダブルクオーテーションで囲む
数値(number)	0や1.34等
真偽値(boolean)	true または false
ヌル値(null)	null と記述
配列(array)	[1,true] 等、カンマ区切りで要素を記述
オブジェクト(object)	{“name”:”Apple”, “age”:”24”} など { } の中に「キー：値」で記述

進 JSONは文字列や数値だけじゃなく、nullやオブジェクトもテキストで表現できるんじゃのう。

愛 この子たちは、この中でしか生きられないもの。私と同じ…。

進 愛ちゃん…。

講師 …え〜、このようなJSONのデータは、Pythonのjsonモジュールを使ってオブジェクトとして扱うことができます（**表2**）。

表2●JSONのデータとPythonオブジェクトの対応

JSON	Python
文字列値(string)	文字列
数値(number)	数値
真偽値(boolean)	true →True、false →False
ヌル値(null)	None
配列(array)	リスト
オブジェクト(object)	辞書

講師 それではWeather Hacksから、JSONデータフォーマットで神奈川県のお天気情報を取得しましょう。まずは、「お天気Webサービス(REST)」のページにある「お天気Webサービス仕様」リンクへ飛び、「全国の地点定義表(RSS)」リンクで「地域別に定義されたID番号」を調べます(**図2**)。

図2●地域別ID番号を調べる

http://weather.livedoor.com/weather_hacks/webservice

```
...
<pref title="神奈川県">
<warn title="警報・注意報" source="http://weather.livedoor.
com/forecast/rss/warn/14.xml"/>
<city title="横浜" id="140010" source="http://weather.live
door.com/forecast/rss/area/140010.xml"/>
<city title="小田原" id="140020" source="http://weather.liv
edoor.com/forecast/rss/area/140020.xml"/>
</pref>
...
```

神奈川県小田原地域のID番号

愛 小田原地域のID、140020…。

講師 地域IDがわかったら、「http://weather.livedoor.com/forecast/webservice/json/v1?city=ID番号」のようにURLを指定してリクエストを送信します。リクエストの送信とレスポンスの取得方法は、Webスクレイピングと同じです。取得できたら、jsonモジュールのloadsを使いオブジェクトに変換します。

sample01.py

```
import urllib.request
import json

url = 'http://weather.livedoor.com/forecast/webservice/
json/v1?city=140020'
res = urllib.request.urlopen(url )
json_data = res.read().decode('utf-8')
odawara = json.loads(json_data)
print(odawara)
```

jsonモジュールのインポート

小田原地域の情報をJSON形式で取得

Pythonのオブジェクトに変換

IPythonコンソール

```
In [2]: runfile('C:/Python/sample01.py', wdir='C:/Python')
{'pinpointLocations': [{'link': 'http://weather.livedoor.
com/area/forecast/1415000', 'name': '相模原市'}, {'link':
'http://weather.livedoor.com/area/forecast/1415011',
...
```

```
{'pinpointLocations':
[{'link': 'http://weather.livedoor.com/area/foreca
st/1415000', 'name': '相模原市'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1415011', 'name': '相模原市西部'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1415012', 'name': '相模原市東部'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1420600', 'name': '小田原市'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1421100', 'name': '秦野市'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1421200', 'name': '厚木市'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1421400', 'name': '伊勢原市'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1421700', 'name': '南足柄市'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1436100', 'name': '中井町'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1436200', 'name': '大井町'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1436300', 'name': '松田町'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1436400', 'name': '山北町'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1436600', 'name': '開成町'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1438200', 'name': '箱根町'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1438300', 'name': '真鶴町'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1438400', 'name': '湯河原町'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1440100', 'name': '愛川町'},
{'link': 'http://weather.livedoor.com/area/foreca
st/1440200', 'name': '清川村'}],
'link': 'http://weather.livedoor.com/area/foreca
st/140020', 'forecasts': [{'dateLabel': '今日', 'telop':
'晴のち曇', 'date': '2019-10-01', 'temperature': {'min':
None, 'max': None}, 'image': {'width': 50, 'url': 'http://
weather.livedoor.com/img/icon/5.gif', 'title': '晴のち曇',
'height': 31}}, {'dateLabel': '明日', 'telop': '晴のち曇',
```

取得したJSONデータは辞書になっている

'date': '2019-10-02', 'temperature': {'min': {'celsius': '18', 'fahrenheit': '64.4'}, 'max': {'celsius': '27', 'fahrenheit': '80.6'}}, 'image': {'width': 50, 'url': 'http://weather.livedoor.com/img/icon/5.gif', 'title': '晴のち曇', 'height': 31}}, {'dateLabel': '明後日', 'telop': '曇り', 'date': '2019-10-03', 'temperature': {'min': None, 'max': None}, 'image': {'width': 50, 'url': 'http://weather.livedoor.com/img/icon/8.gif', 'title': '曇り', 'height': 31}}], 'location': {'city': '小田原', 'area': '関東', 'prefecture': '神奈川県'}, 'publicTime': '2019-10-01T17:00:00+0900', 'copyright': {'provider': [{'link': 'http://tenki.jp/', 'name': '日本気象協会'}], 'link': 'http://weather.livedoor.com/', 'title': '(C) LINE Corporation', 'image': {'width': 118, 'link': 'http://weather.livedoor.com/', 'url': 'http://weather.livedoor.com/img/cmn/livedoor.gif', 'title': 'livedoor 天気情報', 'height': 26}}, 'title': '神奈川県 小田原 の天気', 'description': {'text': ' 北日本から東日本は高気圧に覆われています。一方、台風第18号が東シナ海にあって、北へ進んでいます。¥n\n 神奈川県は、おおむね晴れています。¥n¥n 1日は、高気圧に覆われますが、湿った空気の影響により、晴れ夜遅く曇りとなるでしょう。¥n¥n 2日は、午前中は高気圧に覆われておおむね晴れますが、台風が東シナ海を北東へ進み、湿った空気が流れ込むため、午後は曇りとなる見込みです。¥n¥n 神奈川県の海上では、1日から2日にかけて多少波があるでしょう。', 'publicTime': '2019-10-01T16:33:00+0900'}}

講師 ここから、正規表現を使って必要な情報だけを切り出すこともできますが、データはPythonの辞書やリストになっているので、キーを指定して取得できます。このキーの名前は、Weather Hacksの「お天気Webサービス仕様」ページで紹介されています（**図3**）。

図3●「お天気Webサービス仕様」ページ

http://weather.livedoor.com/weather_hacks/webservice

講師 例えば、「神奈川県 小田原 の天気」は、「title」キーで取り出せます。また「forecasts」キーの最初の要素の中にある「dateLabel」キーでは、「今日」という文字列、「telop」キーには「今日のお天気を表すテロップ文字列」があるので、次のコードを使いピンポイントで表示することが可能です。

sample02.py

```
import urllib.request
import json

url = 'http://weather.livedoor.com/forecast/webservice/
json/v1?city=140020'
res = urllib.request.urlopen(url )
html = res.read().decode('utf-8')
jdata = json.loads(html)

print(jdata['title'])
print(jdata['forecasts'][0]['dateLabel'])
print(jdata['forecasts'][0]['telop'])
```

タイトルと今日のお天気の表示

IPythonコンソール

```
In [2]: runfile('C:/Python/sample02.py', wdir='C:/Python')
神奈川県 小田原 の天気
今日
晴のち曇
```

実際の表示

愛 Pythonだもの…。

講師 今回は天気情報をHTTPプロトコルで取得しましたが、インターネットでは様々なWebサービスが提供されています。有料ですがビジネスですぐに利用できるものもあるので、どのようなWebサービスがあるのかを調べると業務に役立つと思いますよ。

進 愛 ハーイ。

講師 今日は、ここまでとしましょう。

3日目

OpenCVで画像認識

3日目 Part 1 OpenCVの使い方

講師 今日は、OpenCVを使って画像処理を行います。

愛 OpenCV？

進 おやっ、愛ちゃんでも知らんことあるんや。

講師 OpenCVは、Anacondaにインストールされていないので、opencv-pythonという非公式のパッケージをpipでインストールします。Anaconda Promptを起動して、次のコマンドでインストールしましょう。

Anaconda Prompt

```
(base) C:¥Users¥User>pip install opencv-python
Collecting opencv-python
   Downloading https://files.pythonhosted.
1f/51/e0b9cef23098bc31c77b0e06221dd8d05119b
2e22b5f5e/opencv_python-4.1.1.26-cp37-cp37m-win_amd64.whl
(39.0MB)
     100% |████████████████████████████████
████| 39.0MB 89kB/s
Requirement already satisfied: numpy>=1.14.5 in c:¥users¥use
r¥anaconda3¥lib¥site-packages (from opencv-python)
(1.16.2)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.1.1.26

(base) C:¥Users¥User>
```

pip install opencv-python[Enter]と入力

opencv-pythonのインストール完了

講師 インストールが完了したら、OpenCVを使って画像を表示してみます。表示する画像は、Spyderの作業フォルダーにコピーしてあるので確認してください（**図1**）。

図1●サンプル画像のコピー

進 うまそうな本ズワイガニとツブ貝じゃ。本ズワイガニは山陰産、ツブ貝は…北海道産と見た!

講師 お、さすがは水産学部出身ですね。

進 まあのう。

講師 今回は一般的なPNG画像にしましたが、OpenCVは様々な画像フォーマットに対応しています(**表1**)。

表1●OpenCVが対応する画像フォーマットと拡張子

フォーマット	拡張子
Windows bitmaps	*.bmp, *.dib
JPEG files	*.jpeg, *.jpg, *.jpe
JPEG 2000 files	*.jp2

表1の続き

フォーマット	拡張子
Portable Network Graphics	*.png
WebP	*.webp
Portable image format	*.pbm, *.pgm, *.ppm
Sun rasters	*.sr, *.ras
TIFF files	*.tiff, *.tif

愛 PNGじゃなくても代わりはいるもの…。

講師 そう…ですね。JPEGなどの画像もPNGと同じように表示できます。これらの画像ファイルをOpenCVで表示するには、最初にimread関数を使って画像を読み込んでから（表2）、imshow関数で表示します（表3）。

表2●cv2.imread関数の仕様

cv2.imread(filename)	
ファイルから画像データを読み込む	
引数	説明
filename	読み込む対象のファイル名

戻り値	retval（numpy.ndarray）

表3●cv2.imshow関数の仕様

cv2.imshow(winname, image)	
画像オブジェクトをウインドウに表示する	
引数	説明
winname	ウィンドウの名前
image	表示する画像

戻り値	None

講師 imread関数の戻り値である画像のオブジェクトは、NumPyの多次元配列numpy.ndarray型です。また、imshow関数を呼び出した後で、waitKey関数でキーイベント待ちをしないと表示できません（表4）。そして、ウインドウのクローズボタンでウインドウを破棄するために、destroyWindow関数で終了します（表5）。

表4●cv2.waitKey関数の仕様

cv2.waitKey(delay)	
キーイベントを待つ	
引数	説明
delay	ミリ秒単位で表される遅延時間（0の場合は、キーが押されるまで待つ）

戻り値	retval（キーコードまたは－1）

表5●destroyWindow関数の仕様

cv2.destroyWindow()
オープンされているウインドウを破棄する

戻り値	None

講師 それでは、次のコードでカニの画像を表示してみましょう（図2）。

OpenCVのインポート

sample01.py

```
import cv2

try:
  img = cv2.imread('sample01.png')
  if img is None:
    raise FileNotFoundError('ファイルが見つかりません。')

  cv2.imshow('sample01', img)
  cv2.waitKey(0)
  cv2.destroyAllWindows()

except FileNotFoundError as e:
  print(e)
```

画像ファイルの読み込み

画像の表示

IPythonコンソール

```
In [1]: runfile('C:/Python/sample01.py', wdir='C:/Python')
```

図2●OpenCVで表示した画像

進 できたぞい。try:～exceptは、例外処理じゃな。

講師 そうです。imread関数はファイルが見つからない場合でもエラーにならないので、画像オブジェクトが生成されなかった場合は、raise文でFileNotFoundErrorを発生するようにしています。

愛 画像表示ならOpenCVじゃなくてもいいわ。

進 愛ちゃん…、それを言っちゃあ…。

講師 確かに、OpenCV以外にも画像を表示できるライブラリはありますね。それでは、「透視投影」を試しましょう。透視投影とは、3次元の物体を見た通りに2次元平面に描画するための図法です。OpenCVには、透視投影用の変換行列を生成するgetPerspectiveTransform関数があります（**表6**）。この関数を使ってみましょう。

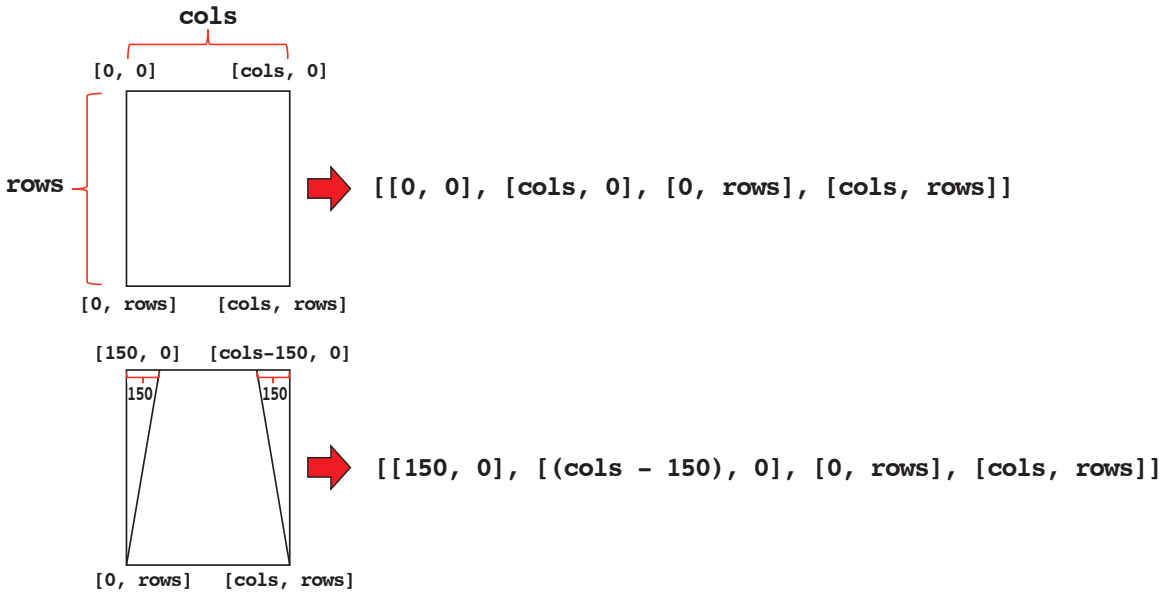
表6●cv2.getPerspectiveTransform関数の仕様

cv2.getPerspectiveTransform(src, dst)		
透視変換行列を生成する		
引数	データ型	説明
src	numpy.ndarray	入力画像、四角形の頂点座標
dst	numpy.ndarray	出力画像、四角形の頂点座標

戻り値	retaval（透視変換用、3×3の行列）

講師 引数に渡す変換前の座標と変換後の座標は、次のように指定します（**図3**）。

図3●透視投影の座標



講師 この座標をgetPerspectiveTransform関数に渡し、透視変換行列を生成します。続いて、getPerspectiveTransform関数で生成した透視変換行列を、warpPerspective関数の引数Mに渡して、画像を変換します（**表7**）。

表7●cv2.warpPerspective関数の仕様

cv2.warpPerspective(src, M, dsize)	
透視変換行列を使い入力画像を変換する	
引数	**説明**
src	入力画像
M	3×3の透視変換行列
dsize	出力画像のサイズ

戻り値	出力画像

講師 では、ツブ貝の画像を上に行くほど小さくリサイズされ、奥に倒れたような画像に透視変換してみましょう。ソースコードは、次のようになります。

sample02.py

```
import cv2
import numpy as np

try:
  img = cv2.imread('sample02.png')
  if img is None:
    raise FileNotFoundError('ファイルが見つかりません。')

  rows,cols,ch = img.shape

  pts1 = np.float32([[0,0],[cols,0],[0,rows],[cols,rows]])
  pts2 = np.float32([[(0 + 100),0],[(cols - 100),0],[0,rows
],[cols,rows]])

  M = cv2.getPerspectiveTransform(pts1,pts2)
  dst = cv2.warpPerspective(img,M,(cols,rows))

  cv2.imshow('sample02', dst)

  cv2.waitKey(0)
  cv2.destroyAllWindows()

except FileNotFoundError as e:
  print(e)
```

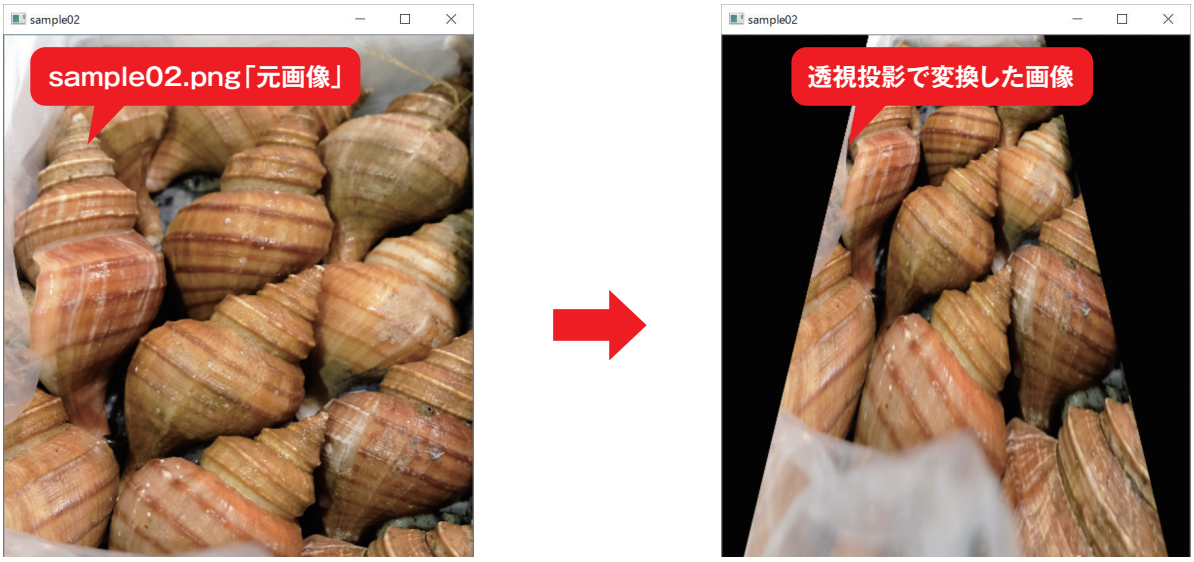
OpenCV、NumPyのインポート
画像ファイルの読み込み
元画像の座標
変換後の座標
透視変換行列を求める
透視変換
画像の表示

IPythonコンソール

```
In [2]: runfile('C:/Python/sample02.py', wdir='C:/Python')
```

講師 図4のような結果になりましたか?

図4●透視投影



愛 キレイ…。

進 愛ちゃんも、喜んでおります。

講師 それはよかった。では、休憩としましょう。

3日目 Part 2 QRコードの利用

講師 OpenCVの機能と言えば、イメージ処理、動画処理、カメラ制御が有名ですが、最新のOpenCV 4にはQRコード用の関数が追加されています。

愛 QRコード決済!

講師 決済機能はありませんが、ORコードを読むことはできます。QRコードは文字や数字を固有のパターンで表したもので、バーコードよりも多くの情報を埋め込むことができます。URL入力や商品情報タグ、キャッシュレス決済などに利用されていますね。

進 そういえば、最近は何でもQRコードがついてるのう。

講師 例えば次のように、QRコードの写真や画像があれば、OpenCVでQRコードを読むことができます（**図1**）。

図1●QRコードの写真

講師 OpenCVでQRコードを読むには、QRCodeDetector関数でQRCodeDetectorオブジェクトを取得し、detectAndDecode関数を使い、情報を取り出します（**表1**、**表2**）。

表1●cv2.QRCodeDetector関数の仕様

cv2.QRCodeDetector()	
QRCodeDetector オブジェクトを生成する	
戻り値	QRCodeDetector object

表2●QRCodeDetector.detectAndDecode関数の仕様

QRCodeDetector.detectAndDecode(InputArray img)	
QR コードを検出およびデコードする	
引数	説明
img	QR コードを検出する画像

表2の続き

戻り値	String, OutputArray, OutputArray

講師 それでは、次のコードで確認しましょう。

sample01.py

```
import cv2

try:
  img = cv2.imread('sample03.png')
  if img is None:
    raise FileNotFoundError('ファイルが見つかりません。')

  detector = cv2.QRCodeDetector()
  retval, points, straight_qrcode = detector.detectAndDeco
de(img)
  print(retval)

except FileNotFoundError as e:
  print(e)
```

画像の読み込み

Detectorの生成

QRコードの読み取り

IPythonコンソール

```
In [3]: runfile('C:/Python/sample01.py', wdir='C:/Python')
ズワイガニ,2019年12月1日,鳥取県
```

進 表示された「ズワイガニ,2019年12月1日,鳥取県」が、QRコードに埋め込まれていた文字列じゃな。

愛 自分のQRコードが欲しい…。

進 おお、そうじゃ。オリジナルのQRコードが作りたいんじゃな。

講師 QRコードの生成は、qrcodeモジュールをインストールします。qrcodeをpipでインス

トールしましょう。

Anaconda Prompt

```
(base) C:¥Users¥User>pip install qrcode
Collecting qrcode
  Downloading https://files.pythonhosted.org/packages/42/87
/4a3a77e59ab7493d64da1f69bf1c2e899a4cf81e51b2baa855e8cc811
5be/qrcode-6.1-py2.py3-none-any.whl
Requirement already satisfied: colorama; platform_system
== "Windows" in c:¥users¥user¥anaconda3¥lib¥site-packages
(from qrcode) (0.4.1)
Requirement already satisfied: six in c:¥users¥user¥anaconda
3¥lib¥site-packages (from qrcode) (1.12.0)
Installing collected packages: qrcode
Successfully installed qrcode-6.1

(base) C:¥Users¥User>
```

pip install qrcode [Enter]と入力

講師 qrcodeは、pillowという画像処理モジュールを使っていますが、こちらはAnacondaにインストール済みなので、インストールしなくても大丈夫です。それでは、qrcodeモジュールを使ってQRコードを生成してみましょう。QRコードを生成するには、qrcode.make関数で文字列からQRコードのImegeオブジェクトを生成するだけです（**表3**）。生成されたImegeオブジェクトのsaveメソッドでファイルに保存することもできます（**表4**）。

表3●qrcode.make関数の仕様

qrcode.make(qr_string)	
QR コードを生成する	
引数	**説明**
qr_string	QR コードを検出する画像

戻り値	PIL（pillow 形式）の Image オブジェクト

表4●Image.save関数の仕様

Image.save(fp)	
画像データをファイルに保存する	
引数	説明
fp	保存するファイル

戻り値	None

講師 次のプログラムを実行して、生成されたQRコードを確認しましょう（図2）。

sample02.py

qrcodeのインポート

```
import qrcode

file_name = 'qr20191201.png'
qr_data = 'ツブ貝,2019年11月30日,北海道'
image = qrcode.make(qr_data)

image.save(file_name)
print('ファイル名 : ', file_name)

print('データ : ',  qr_data)
print('QRコードを生成しました。')
```

QRコード画像を生成

画像をファイルに保存

IPythonコンソール

```
In [4]:runfile('C:/Python/sample02.py', wdir='C:/Python')
ファイル名 :  qr20191201.png
データ :  ツブ貝,2019年11月30日,北海道
QRコードを生成しました。
```

図2●生成されたQRコード

進 わしらの会社は、商品を出荷するとき、荷札にバーコードを使っとるが、そろそろQRコードに切り替えんといかんじゃろうなぁ。

愛 ワタシ…、作る…。

講師 QRコードは強力な誤り訂正の機能を持っているそうなので、水産業の現場にはピッタリでしょう。期待してますよ。次は、OpenCVで画像から「オブジェクト」を検出します。

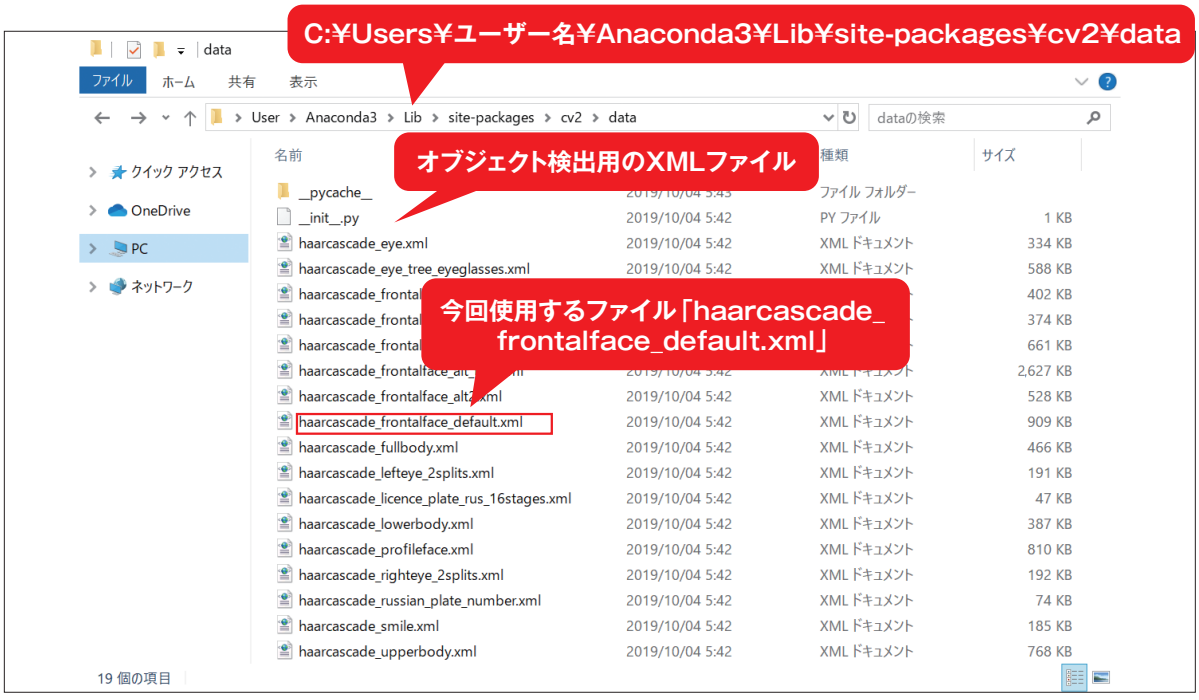
Part 3 3日目 オブジェクト検出

講師 OpenCVは、機械学習を利用した画像認識もできるんです。

愛 オブジェクトを認識できるの?

講師 はい。画像からオブジェクトを検出するには、機械学習によりオブジェクトの特徴を収集したデータが必要ですが、OpenCVにはオブジェクトの特徴を学習したデータがXMLファイルで用意されています(**図1**)。

図1●オブジェクト検出用のXMLファイル



講師 このXMLファイルを使って、お二人の写真から「顔」を検出してみましょう。

進 いよいよ、AIプログラミングの開始じゃのう。

講師 それでは、顔検出用のXMLファイル「haarcascade_frontalface_default.xml」と、お二人の写真をSpyderの作業フォルダーにコピーしましょう(**図2**)。

図2●XMLファイルと検出用のファイルのコピー

講師 OpenCVで顔検出を行うには、まずCascadeClassifier関数で特徴量を学習したXMLファイルを読み込みます（**表1**）。

表1●cv2.CascadeClassifier関数の仕様

cv2.CascadeClassifier(filename)	
オブジェクト検出用 XML ファイルを読み込む	
引数	**説明**
filename	検出用ファイル名

戻り値	retval（CascadeClassifier object）

講師 次に、取得したCascadeClassifierオブジェクトのdetectMultiScale関数の引数に、画像データを渡します（**表2**）。

表2●detectMultiScale関数の仕様

CascadeClassifier.detectMultiScale(image)	
オブジェクト検出用 XML ファイルを読み込む	
引数	説明
image	入力画像（8ビット）

戻り値	objects（検出したオブジェクトの矩形座標配列）

講師 次のコードで、栄井さんの写真から「顔」を検出してみましょう。

```
import cv2

try:
  img_name = 'sakai.png'
  xml_data = 'haarcascade_frontalface_default.xml'

  img = cv2.imread(img_name)
  if img is None:
    raise FileNotFoundError('imgファイルが見つかりません。')

  cascade = cv2.CascadeClassifier(xml_data)
  if cascade is None:
    raise FileNotFoundError('xmlファイルが見つかりません。')

  face = cascade.detectMultiScale(img)

  if len(face) > 0:
    for r in face:
      x, y = r[0: 2]
      width, height = r[0:2] + r[2:4]
      cv2.rectangle(img, (x, y), (width, height), (0, 255,
0), thickness=2)
  else:
    print('顔が見つかりません')

  cv2.imwrite(img_name, img)
  cv2.imshow(img_name, img)
```

顔の検出に使う画像

検出用のXMLファイル

顔の検出

顔が検出できたら、その部分に四角を描く

画像データをファイルに保存する

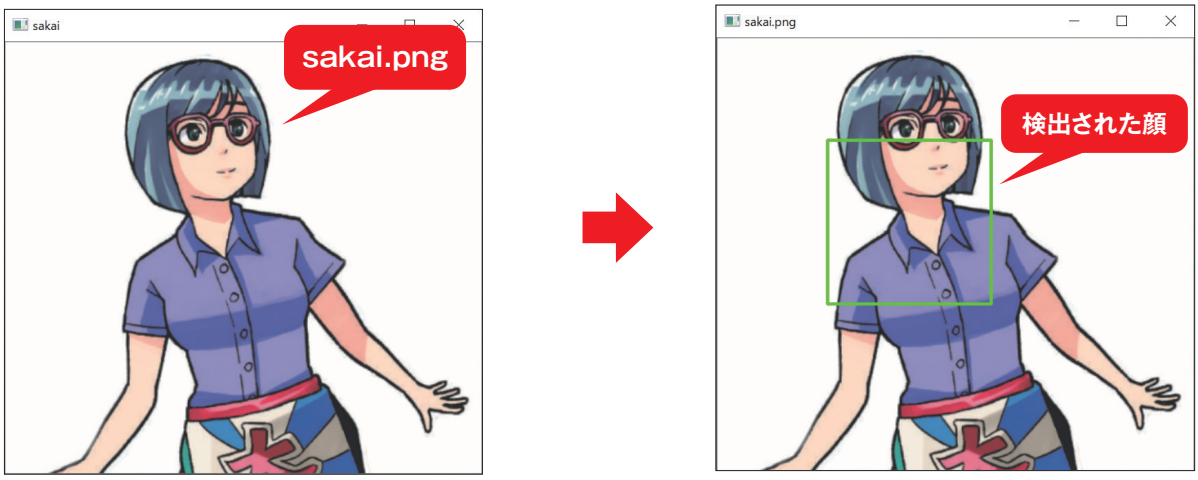
（次ページに続く）

（前ページからの続き）

```
    cv2.waitKey(0)
    cv2.destroyAllWindows()

except FileNotFoundError as e:
    print(e)
```

愛 違う…私の顔…（図3）。

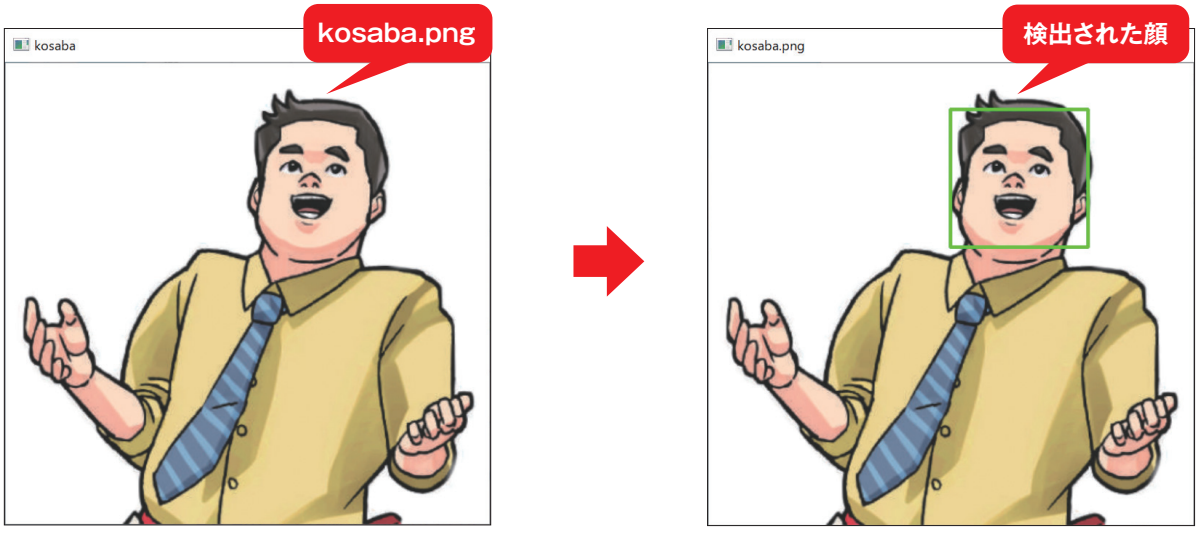
図3●愛ちゃんの顔の検出結果



講師 そうですねぇ。もしかすると、検出しにくい写真なのかもしれません。

進 じゃ、わしの顔ではどうじゃ（図4）。

図4●小鯖君の顔の検出結果



進 やったぞい。わしの顔は検出したぞい。

講師 検出精度は、検出用データや写真によって変わります。検出用のXMLファイルは、インターネット上に多数アップロードされています。様々なデータで認識率の違いを試してみてください。それでは今日はここまで。お疲れ様でした。

4日目

scikit-learnで機械学習その1

4日目 Part 1 機械学習とは

講師 いよいよ、機械学習プログラミングに入ります。

進 ついに!

愛 フッ…。

講師 一言で「機械学習」と言っていますが、機械学習の手法には「教師あり学習」と「教師なし学習」の2種類があります。

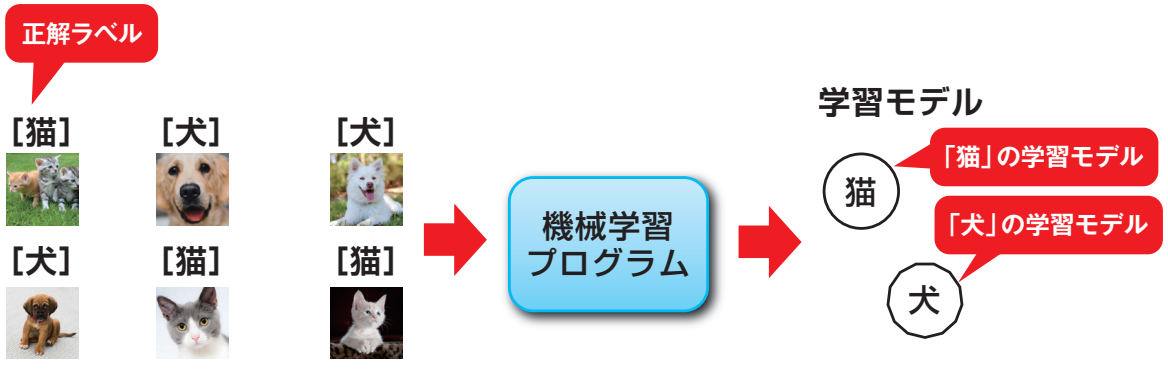
進 つまり、学校に通うか、独学するかの違いじゃのう。

講師 え〜と…違います。

進 違うんか〜い。

講師 教師あり学習は、事前に与えられた「正解ラベル」をガイドにして機械学習を行います。例えば、「猫」の画像を検出できるように学習するため、あらかじめ「猫」とラベリングされた画像を用意して、「猫の特徴」を学習させます。こうして、オブジェクトの特徴を学習した機械学習プログラムを「学習モデル」と呼びます(**図1**)。

図1●教師あり学習



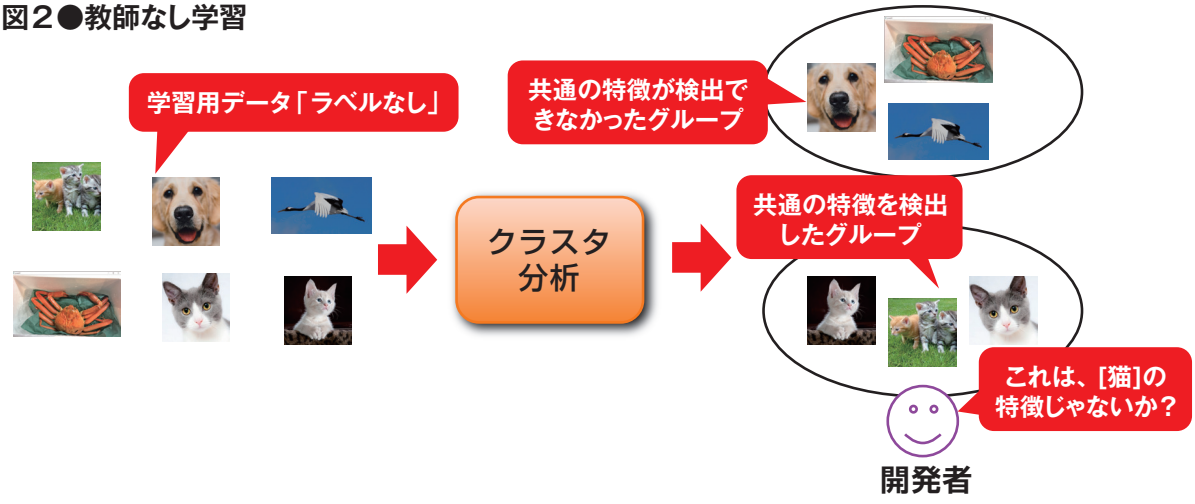
愛「猫の学習モデル」は、猫を検出できる…。

講師 そうです。こうしてオブジェクトの特徴を学習した「学習モデル」は、ラベルの付いていないデータが入力されると、学習した特徴と一致するかどうかを調べて、同じオブジェクトかどうかを判断します。

進 教師とは、正解ラベルのことかい。なら「教師なし」は、データに正解が書いてないっちゅうことか。

講師 その通り。「教師なし学習」とは、何もラベリングされていない画像の中から、共通の特徴を見つけ出すことです。もし、共通の特徴を持つ画像に「猫」が多かった場合、「この特徴は、猫の特徴ではないか」と判断できます（図2）。

図2●教師なし学習



愛「教師なし学習」は、共通の特徴を見つけ出す…。

講師 そして、共通の特徴によりデータをグループ分けします。この分類手法には「クラスタリング」と呼ばれる手法があります。

進 なんや…急に難しゅうなってきたのう…。

講師 機械学習のアルゴリズムは、教師ありと教師なしを合わせると、非常に多くの種類があり、ここですべてを解説することはできません。今回は、一番基本となる機械学習の

アルゴリズムを利用して、「アヤメの分類」と「手書き文字の認識」を行ってみましょう。

進 お手柔らかにたのんます…。

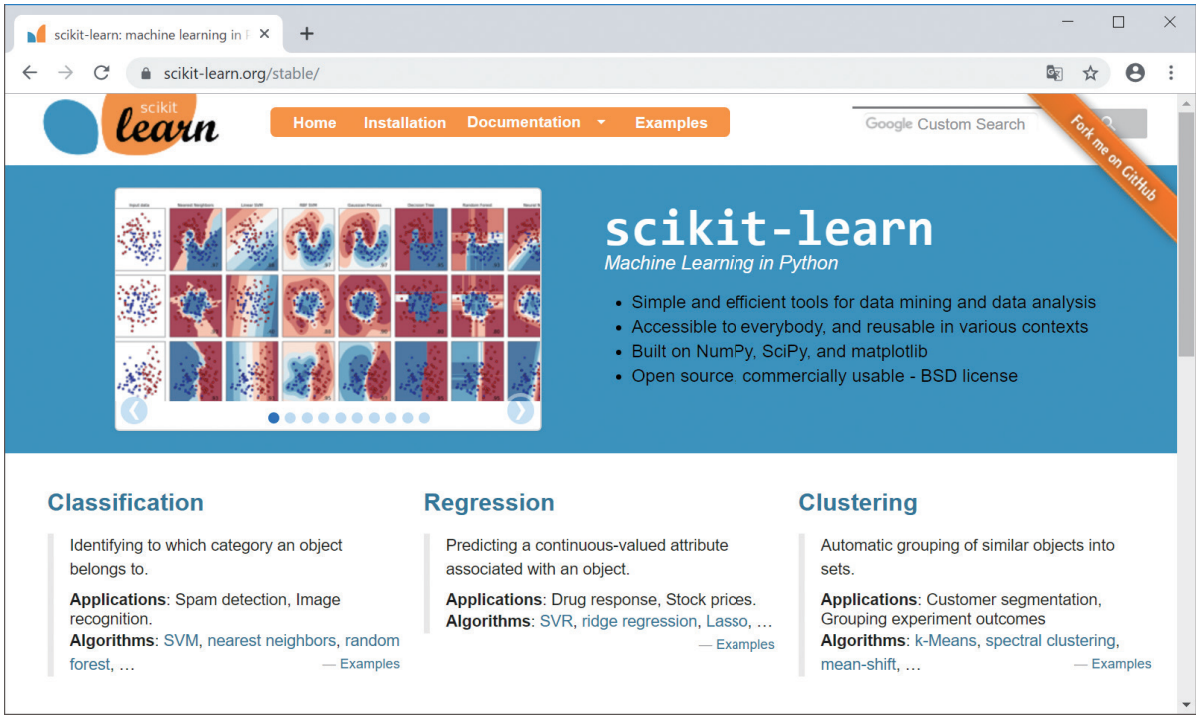
講師 それではいったん休憩して、Pythonの機械学習ライブラリの話へ進みます。

4日目 Part 2 scikit-learnの使い方

講師 Pythonで機械学習を行うのであれば、はじめのうちは「scikit-learn」（サイキット・ラーン）という外部ライブラリを使うのがオススメです（**図1**）。

図1●scikit-learnの公式サイト

http://scikit-learn.org



講師 scikit-learnは外部ライブラリですが、Anacondaには含まれているので、すぐに機械学習のプログラムを試すことができます。

進 Anacondaは、神じゃのう。神過ぎるのう。

愛 神…。

講師 scikit-learnには、機械学習に利用できるアルゴリズムが数多く実装されています。表1で代表的なものをいくつか紹介しておきます。

表1●機械学習に利用される主なアルゴリズム

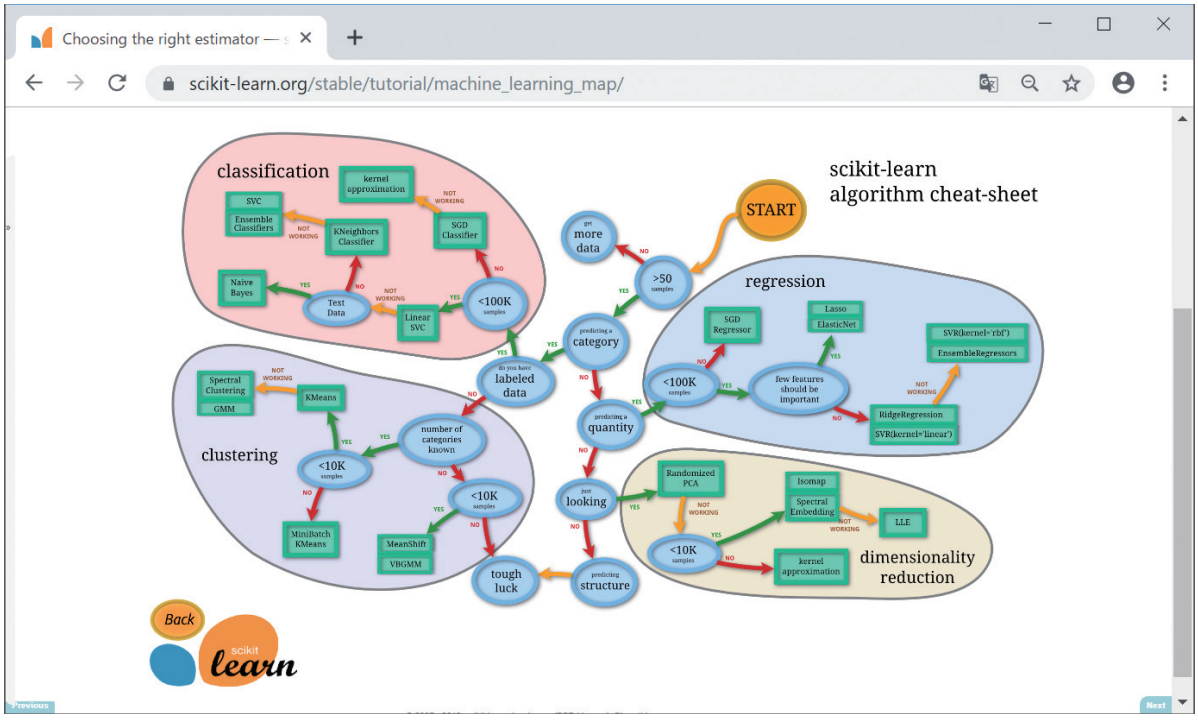
アルゴリズムの種類	説明
回帰 (regression)	実数値をデータで学習し、実数値を予測する。scikit-learn が搭載するアルゴリズムには、SGD 回帰、LASSO 回帰などがある
分類 (classification)	正解ラベルとそのデータを学習し、データに対してのラベルを予測する。scikit-learn が搭載するアルゴリズムには、カーネル近似、k 近傍法などがある
クラスタリング (clustering)	データの似ている部分をグループにして、データの特徴やパターンを発見する。scikit-learn が搭載するアルゴリズムには、k 平均法、スペクトラルクラスタリングなどがある

愛 どのアルゴリズムがいいかな…。

講師 アルゴリズムに迷ったら、scikit-learnのサイトを参照しましょう。質問にYes、Noで答えていけば、最適なアルゴリズムにたどり着くチャートが用意されています（図2）。

図2●アルゴリズムマップ

http://scikit-learn.org/stable/tutorial/machine_learning_map/



進 親切じゃ。

講師 それではさっそく機械学習を試しましょう…と言いたいのですが、まずは「機械学習で何を学習するのか」を決める必要があります。

進 やっぱり、オブジェクト検出じゃろう。カメラからの映像で魚介類の種別を判断できると仕事が楽になる。

講師 そうですね。ただ、OpenCVでも確認しましたが画像内のオブジェクト検出はかなり難しく、精度を上げるには大量の画像データが必要です。

愛 私の顔…認識できなかった…。

進 あ〜、愛ちゃんが思い出して悲しんどる…。じゃが、魚の画像なら会社にたくさんありそうじゃ。

愛 でも、整理されてない…。

講師 そうなのです。学習用のデータとして整理してまとめられていないと、ちょっと使いにくいですね。実は機械学習ではプログラムの開発以上に、学習用データの収集と整理が大変なのです。皆さんもこの研修が終わったら、社内に散らばっている学習用のデータを集める作業がアサインされるかもしれませんね。

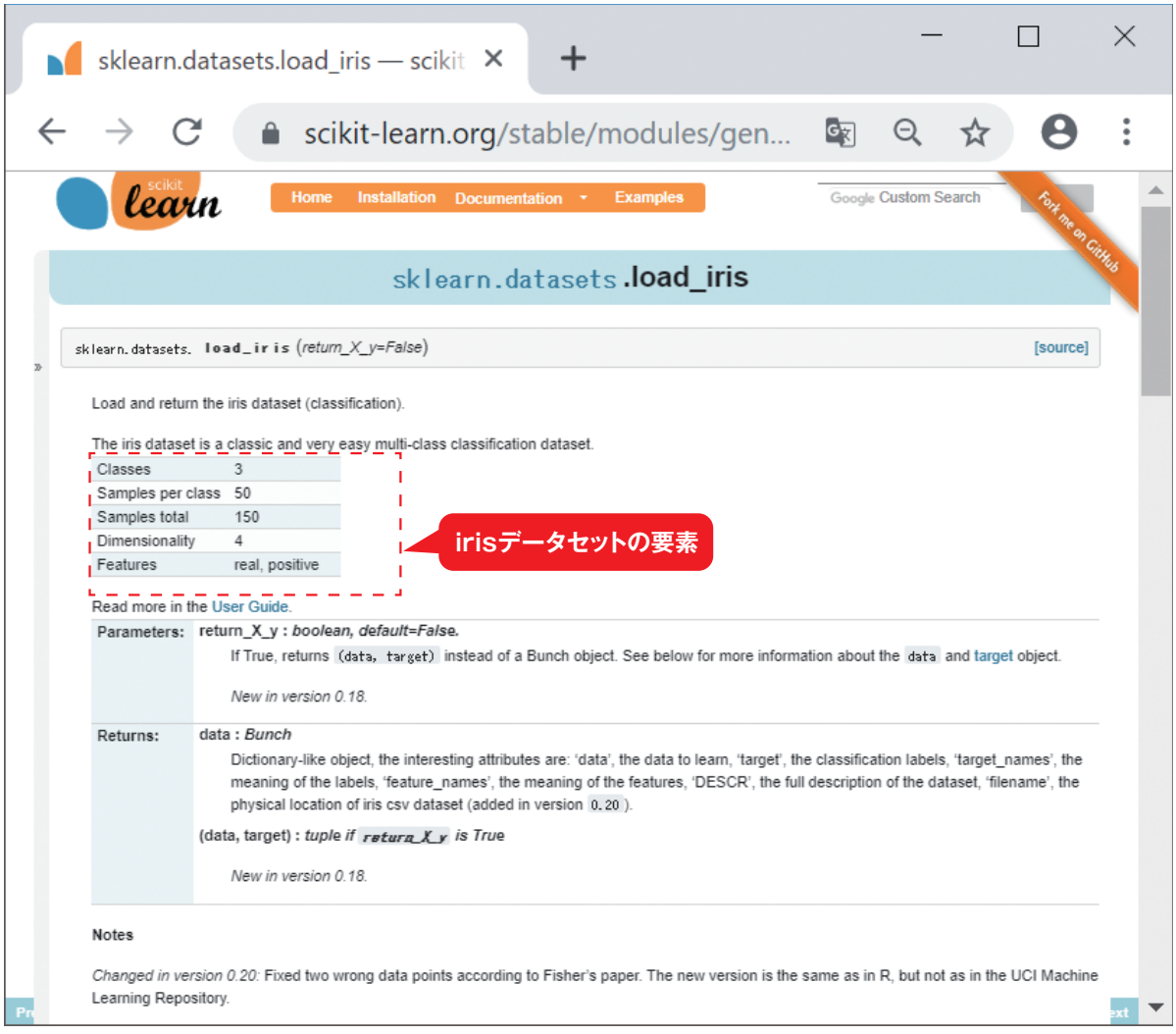
進 なんと!

講師 今回は残念ながら魚介類のデータではありませんが、機械学習を手軽に試すためにscikit-learnがあらかじめ用意している学習用のデータを使います。いくつかあるのですが、「iris」(アイリス)という「アヤメ」(花)のデータセットを使ってみましょう。

進 魚じゃなくて、花を識別するんじゃな。

講師 「irisデータセット」は、統計分析、機械学習のサンプルとして有名で、機械学習の入門の際、必ずと言ってよいほど登場するデータセットです。irisデータセットの読み込みは、専用のload_iris関数があるのでこれを利用します。詳しくは、scikit-learnのサイトに解説があるので、参照してください(**図3**)。

図3●load_irisの説明
http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html



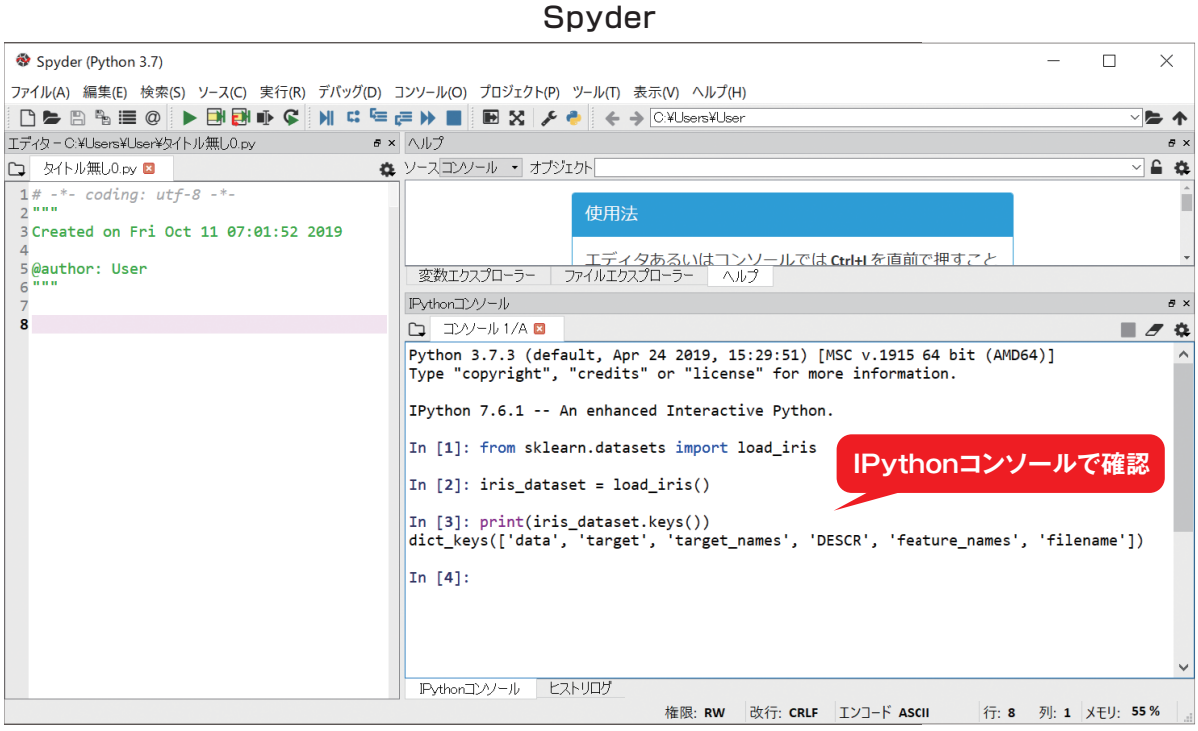
講師 この解説を読むと、irisデータセットの各データは'data'、'target'などのキーで取得できることがわかります。さらに、アヤメのデータは「150個」。単位は「センチメートル」。アヤメには種類があって、「Iris-Setosa」(アイリス-セトサ)、「Iris-Versicolor」(アイリス-バーシクル)、「Iris-Virginica」(アイリス-バージニカ)の3種類があります。それぞれの各アヤメのデータには、「Sepal」(がく片)と、「Petal」(花弁)のlength(長さ)およびwidth(幅)を計測した値が特徴として格納されています。

愛 データがいっぱい…。

進 確かにこれだけのデータを集めるのは大変じゃ。

講師 それでは、irisデータセットをimportして、読み込んでみましょう。機械学習プログラムは、データの確認をしながら進めるので、IPythonコンソールで入力と実行を行ってください（**図4**）。

図4●IPythonコンソールで確認



講師 まずは、iris_dataset.keys関数で、キーを確認します。

IPythonコンソール

```
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915
64 bit (AMD64)]
Type "copyright", "credits" or "license" for more informat
ion.

IPython 7.6.1 -- An enhanced Interactive Python.

In [1]: from sklearn.datasets import load_iris

In [2]: iris_dataset = load_iris()

In [3]: print(iris_dataset.keys())
dict_keys(['data', 'target', 'target_names', 'DESCR', 'fea
ture_names', 'filename'])

In [4]:
```

iris_datasetのキー

講師 この中の'data'キーを使って、アヤメの特徴データを表示できます。ガクの長さ、ガクの幅、花弁の長さ、花弁の幅の順に、4つの値を表示します。

IPythonコンソール

```
In [4]: print(iris_dataset['data'])
[[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]
...
```

ガクの長さ、ガクの幅、花弁の長さ、花弁の幅

講師 'target'キーを表示すると、0、1、2の3種類のアヤメのデータが50個ずつ並んでいることがわかります。また、それぞれの名前を'target_names'キーで確認すると、0がsetosaのデータ、1がversicolorのデータ、2がvirginicaのデータであることがわかります。

IPythonコンソール

```
In [5]: print(iris_dataset['target'])
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2
2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2
 2 2]

In [6]:print(iris_dataset['target_names'])
      ['setosa' 'versicolor' 'virginica']
```

３種類のアヤメのデータが50個単位で並んでいることが分かる

３種類のアヤメの名前

講師 irisデータセットが、どのような構成かわかりました。次に、このデータセットを訓練用（trainimg data）と評価用（test data）に分割します。

愛 クロスバリデーション…。

講師 栄井さん、よく知ってますね。このように、学習データを訓練用と評価用に分割して、学習と評価を繰り返し、限られたデータを効率良く利用する方法を「クロスバリデーション」と呼びます。データの分割には、scikit-learnの「train_test_split」という関数を使います。書式は次のようになります（**表2**）。

表２●model_selection.train_test_split関数の書式。引数は抜粋

model_selection.train_test_split(*arrays, **options)	
引数	説明
*arrays	トレーニング用の特徴行列、評価用の特徴行列、トレーニング用の目的変数、評価用の目的変数
test_size	テストデータのサイズ（1で100%）

表2の続き

model_selection.train_test_split(*arrays, **options)	
引数	説明
random_state	乱数ジェネレータによって使用されるシード値
shuffle	データをシャッフルするか否か（デフォルトTrue）

戻り値	分割されたリスト

講師 train_test_split関数は、渡した引数でデータセットをランダムにシャッフルして並び替え、好きな割合に分割することができます。分割後のリストは、次のように命名しました。scikit-learnでは、データを大文字、正解ラベルを小文字で表現するようなのでそれにならっています。

```
X_train :訓練用の特徴
X_test  :評価用の特徴
x_train :訓練用の正解ラベル
x_test  :評価用の正解ラベル
```

講師 では、irisデータセットを、70%の訓練データと30%の評価用データに分割してみます。

IPythonコンソール

```
In [7]: from sklearn.model_selection import train_test_spl
it

In [8]: X_train, X_test, y_train, y_test = train_test_sp
lit(iris_dataset['data'], iris_dataset['target'], test_
size=0.3,  random_state=0)
```

訓練用データ7割、評価用のデータ3割に分割する

講師 これで訓練用のデータと、学習モデルの評価用データが用意できました。それでは、少し長くなってしまったので休憩としましょう。

Part 3 k近傍法による機械学習

講師 いよいよアヤメのデータを使った機械学習を行います。利用するアルゴリズムは、「k近傍法」(k-Nearest Neighbor algorithm)です。

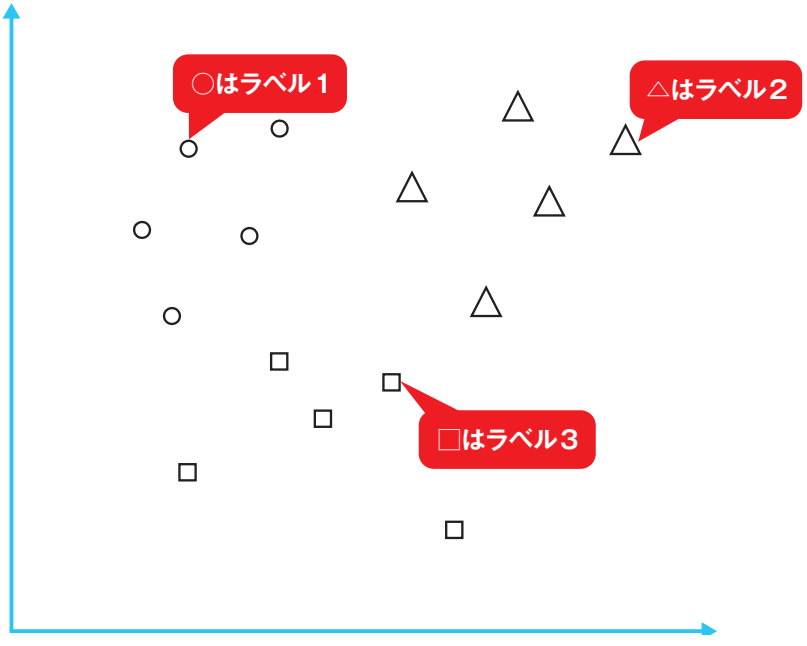
進 ケイ…キン、なんじゃそれは。

講師 k近傍法は「教師あり学習」で分類問題を解くためのアルゴリズムです。分類問題とは、学習データを「クラス」と呼ばれるグループに分類しておき、評価データがどのクラスに分類されるのかを予測する手法です。

愛 アヤメを分類する学習モデル…。

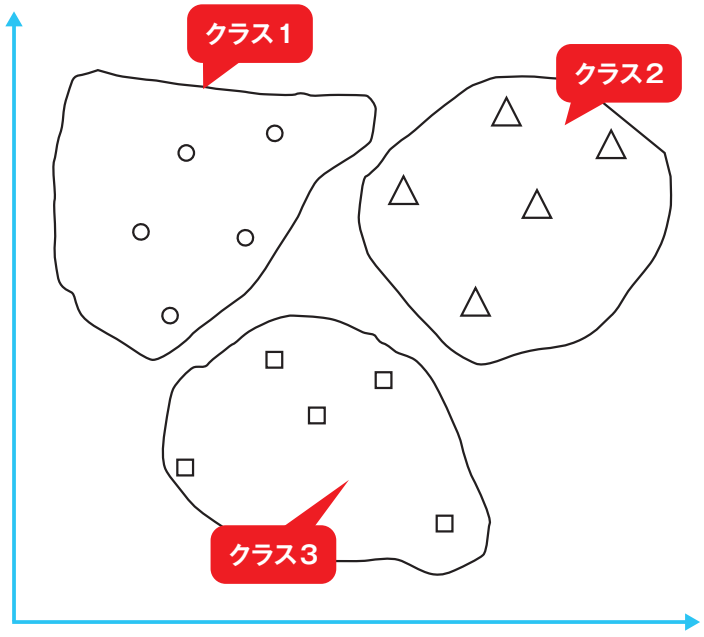
講師 そうです。はじめに、k近傍法の考え方を解説しておきましょう。最初に学習データをベクトル空間上に正解ラベルとともに配置します。教師あり学習なので、正解ラベルが必要です(図1)。

図1●近傍法の考え方(その1)



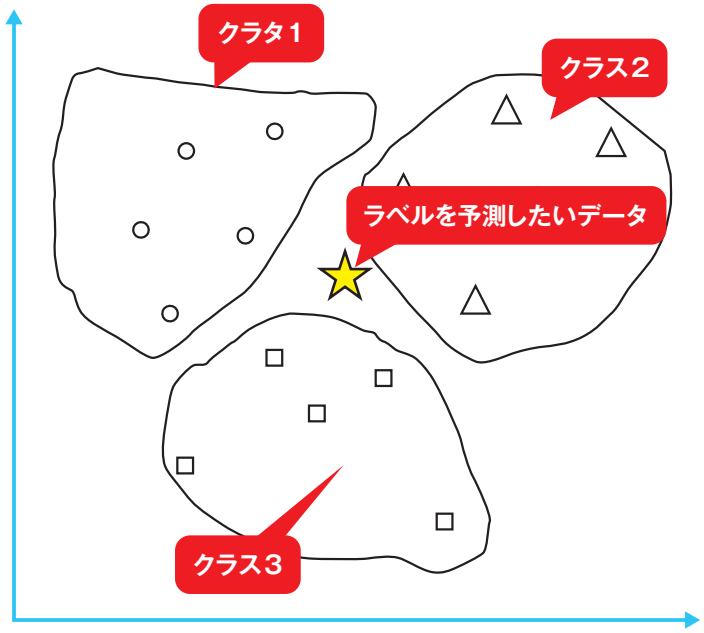
講師 次に、与えられたデータを「特徴ベクトル」として、それぞれ近い場所にある正解ラベルのデータを使ってクラスに分類します。これが、k近傍法の学習モデルです（図2）。

図2●k近傍法の考え方（その2）



講師 続いて、正解ラベルのないデータを学習モデルに与えます（図3）。

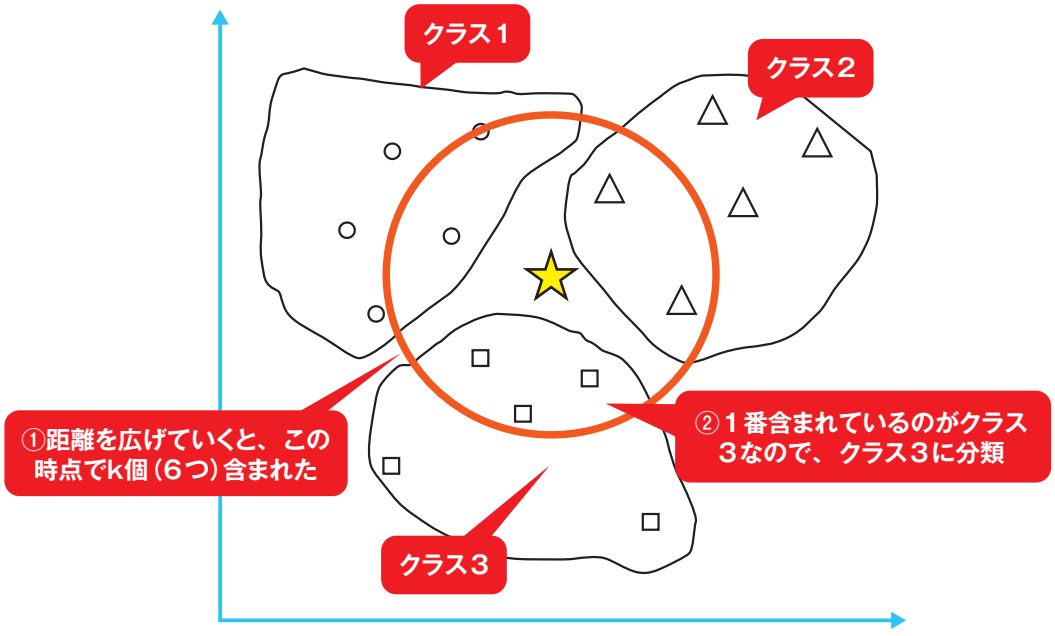
図3●k近傍法の考え方（その3）



進 このデータが、どのクラスに属するのか判定するんじゃな。

講師 はい。学習モデルは、与えられたデータと存在するクラスとの近さ（距離）をもとに、そのデータがどのクラスに属するのかを予測します。k近傍法では「最も近いk個の点がどのクラスに一番多く存在するか」をもとに、属するクラスを決定します。例えば、kが6だったとすると、予測したいデータはクラス3に属することになります（**図4**）。

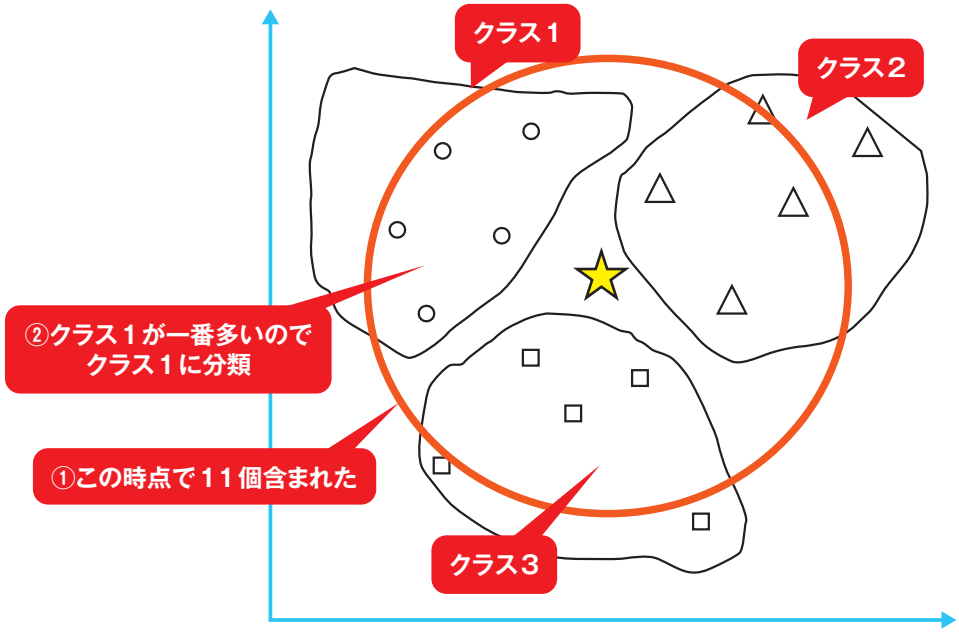
図4●k近傍法の考え方（その4）



愛 でも…、kを変更すると予測結果が変わってしまう…。

講師 そうですね。kを11にすると評価データは、クラス1に分類されることになります（**図5**）。

図5●k近傍法の考え方（その5）



講師 つまり、予測した結果の正解率が高くなるようにkの値を決めることが重要です。最適なkの値は簡単にはわかりませんから、試行錯誤しながら決めることになりますね。

進 なるほど。考え方は理解したぞい。

講師 それでは、アヤメのデータを使って判定してみましょう。まず、訓練用のデータと評価用のデータが、正しく分割されているか、正解ラベルの数とランダム度合いで確認します。

IPythonコンソール

```
In [9]: y_train
Out[9]:
array([1, 2, 2, 2, 2, 1, 2, 1, 1, 2, 2, 2, 2, 1, 2, 1, 0,
2, 1, 1, 1, 1,
       2, 0, 0, 2, 1, 0, 0, 1, 0, 2, 1, 0, 1, 2, 1, 0, 2,
2, 2, 2, 0, 0,
       2, 2, 0, 2, 0, 2, 2, 0, 0, 2, 0, 0, 0, 1, 2, 2, 0,
0, 0, 1, 1, 0,
```

訓練用の正解ラベル

（次ページに続く）

（前ページからの続き）

```
        0, 1, 0, 2, 1, 2, 1, 0, 2, 0, 2, 0, 0, 2, 0, 2, 1,
1, 1, 2, 2, 1,
       1, 0, 1, 2, 2, 0, 1, 1, 1, 1, 0, 0, 0, 2, 1, 2, 0])

In [10]: y_test
Out[10]:
array([2, 1, 0, 2, 0, 2, 0, 1, 1, 1, 2, 1, 1, 1, 1, 0, 1,
1, 0, 0, 2, 1,
       0, 0, 2, 0, 0, 1, 1, 0, 2, 1, 0, 2, 2, 1, 0, 1, 1,
1, 2, 0, 2, 0,
       0])
```

評価用の正解ラベル

講師 k近傍法でクラス分類を行う学習モデルを構築するには、scikit-learnのKNeighborsClassifierクラスを使います。書式は**表1**のようになります。

表1●KNeighborsClassifierクラスのコンストラクタの書式。引数は抜粋

sklearn.neighbors. KNeighborsClassifier(n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=1, **kwargs)	
引数	**説明**
n_neighbors	k の値を指定（デフォルトは k=5）
weights	近傍までの距離を考慮するか否か（デフォルトは距離を考慮しない）
algorithm	最も近い近傍を計算するために使用されるアルゴリズム
n_jobs	並列ジョブの数。 -1 の場合、利用可能な CPU の数に設定される
戻り値	k 近傍法の学習モデルのオブジェクト

講師 訓練データから学習モデルを構築しましょう。KNeighborsClassifierクラスをインポートして、KNeighborsClassifierオブジェクトを生成します。kの値のn_neighborsは、1にしています。これで、学習モデルが生成されました。次に、訓練データをfit関数を使って読み込んで、学習します。

IPythonコンソール

```
In [11]: from sklearn.neighbors import KNeighborsClassifier

In [12]: knn = KNeighborsClassifier(n_neighbors=1)

In [13]: knn.fit(X_train, y_train)
Out[13]:
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric
='minkowski',
                     metric_params=None, n_jobs=None, n_ne
ighbors=1, p=2,
                     weights='uniform')

In [14]:
```

学習モデルの生成

訓練用データの読み込み

講師 KNeighborsClassifierオブジェクトを生成したときのパラメータが表示されます。n_neighborsが1であること以外は、すべてデフォルトなのがわかります。試しに学習データの中からデータを1つpredict関数に与えて、正しく学習しているか確認してみましょう。評価データは、setosaのガクの長さ、ガクの幅、花弁の長さ、花弁の幅です。判別が済むとprediction1に種別を返すので、学習モデルが予想したラベルを表示してみましょう。

IPythonコンソール

```
In [14]: import numpy as np

In [15]: X_new = np.array([[5.0, 2.9, 1.0, 0.2]])

In [16]: prediction1 = knn.predict(X_new)

In [17]: print(iris_dataset['target_names'][prediction1])
['setosa']
```

setosaの特徴データ

どのアヤメのデータか判断させる

正解した

講師 setosaのデータを与えたので、正解です。正しく学習しているようなので、先ほど分割した評価用データを使って、学習モデルがどれくらいの精度を持っているのかを評価しましょう。評価用データを学習モデルにセットします。結果が出たら、正解ラベルと比較します。

IPythonコンソール

評価用の特徴データ

```
In [18]: y_pred = knn.predict(X_test)

In [19]: y_pred
Out[19]:
```

判別した結果

```
array([2, 1, 0, 2, 0, 2, 0, 1, 1, 1, 2, 1, 1, 1, 1, 0, 1,
1, 0, 0, 2, 1,
       0, 0, 2, 0, 0, 1, 1, 0, 2, 1, 0, 2, 2, 1, 0, 2, 1,
1, 2, 0, 2, 0,
       0])

In [20]: y_test
Out[20]:
```

正解ラベル

```
array([2, 1, 0, 2, 0, 2, 0, 1, 1, 1, 2, 1, 1, 1, 1, 0, 1,
1, 0, 0, 2, 1,
       0, 0, 2, 0, 0, 1, 1, 0, 2, 1, 0, 2, 2, 1, 0, 1, 1,
1, 2, 0, 2, 0,
       0])

In [21]: np.mean(y_pred == y_test)
Out[21]: 0.9777777777777777
```

間違えた！

正解率

講師 1カ所、間違えていますね。この学習モデルでは、評価用データに対する精度は約97%の正解率ということです。

進 kの値を3（n_neighbors=3）にしてみても変わらんのう。

愛 これ以上は、無理ってことね…。

講師 そうですね。学習データがもっと多ければ、多少は正解率が上がるかもしれませんが、十分な正解率だと思います。明日は、ニューラルネットワークというアルゴリズムに挑戦します。

5日目

scikit-learnで機械学習その2

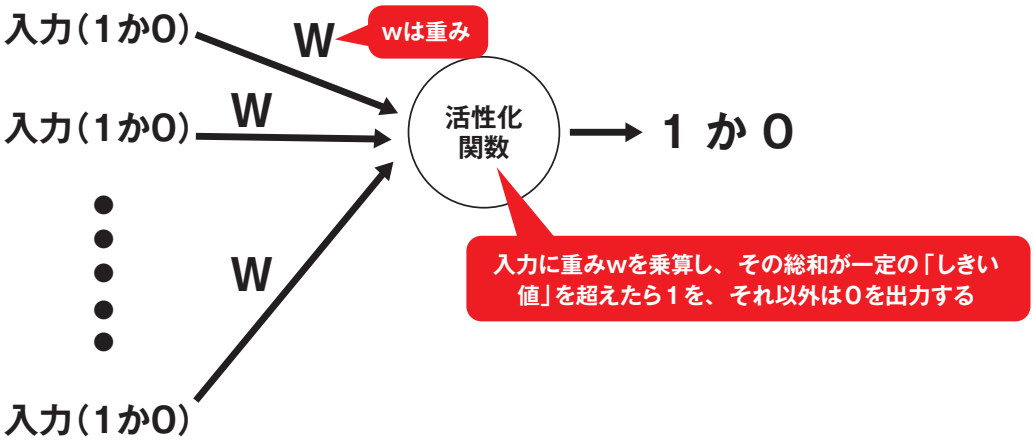
5日目 Part 1 ニューラルネットワーク

講師 最終日の今日は、ニューラルネットワークとクラスタリングを試してみます。ニューラルネットワークとは、人間の脳にある神経細胞をコンピュータのプログラムで再現したものです。

愛 ニューロン…。

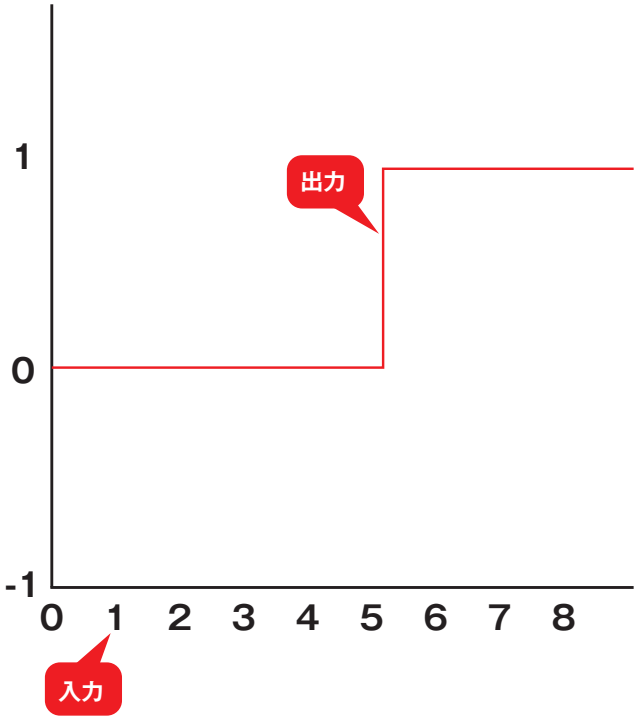
講師 そうです。脳の神経細胞、すなわちニューロンは、外部から様々な電気刺激を受けて興奮し、やがて別のニューロンに電気信号を出力することで興奮が収束するという性質があります。このニューロンを模して作られたのが「形式ニューロン」(人工ニューロン)というプログラムです(図1)。

図1●形式ニューロン



講師 形式ニューロンは、1か0に重みを掛けた値を「活性化関数」に入力し、その総和がしきい値を超えると1を出力し、超えなければ0を出力するという単純なプログラムです。判断基準となる「活性化関数」は、「ステップ関数」が有名ですね(図2)。

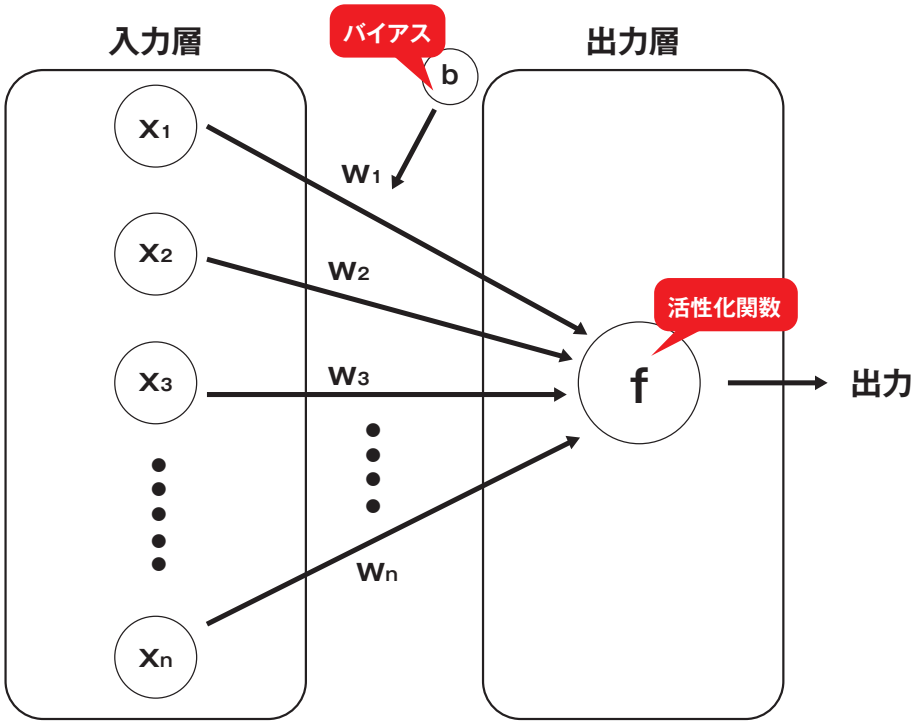
図2●ステップ関数



進 ステップ関数は知ってるぞい。しきい値5とした場合は、入力が5以下なら出力は0、5より大きくなると1になるような関数のことじゃろ。

講師 はい。形式ニューロンは、入力が1か0のどちらかしかありません。そこで、入力を実数に対応させ、かつ、学習により重みを更新できるように改良したのが「単純パーセプトロン」です。単純パーセプトロンでは、入力部分を「入力層」、出力部分を「出力層」、それぞれを構成するパーツを「ニューロン」と呼びます（**図3**）。

図3●単純パーセプトロン



講師 単純パーセプトロンは、学習することで入力したデータを2つのクラスに分類します。このように、クラスを分離するための境界線を「決定境界」と呼びます。

愛 ここが、わたしと小鯖君の決定境界…。

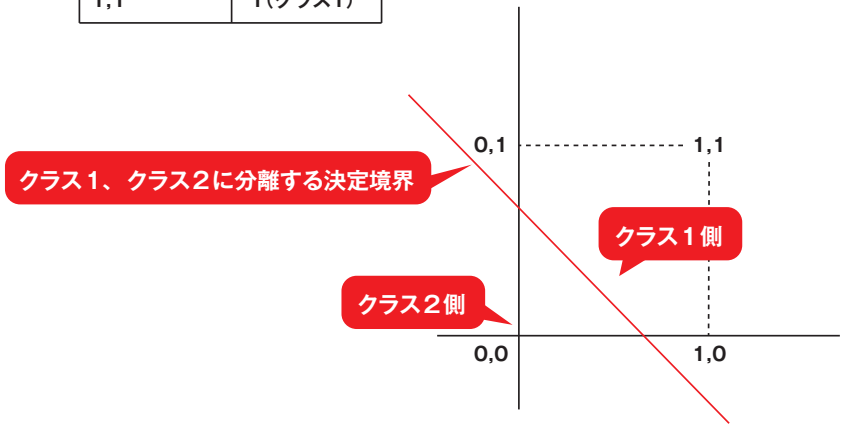
進 愛ちゃん、寂しいこと言わんといて。

講師 実は、この単純パーセプトロンには「XOR問題」という欠点があります。例えば、図4のような、学習データ(入力)と正解ラベルにORの関係があるデータがあったとします。このデータでは、次のように、「クラス1」と「クラス2」を分離する決定境界の直線を引くことができます。

図4●XOR問題（その1）

入力(x_1,x_2)	正解ラベル
0,0	0（クラス2）
1,0	1（クラス1）
0,1	1（クラス1）
1,1	1（クラス1）

入力と正解ラベルの関係はOR

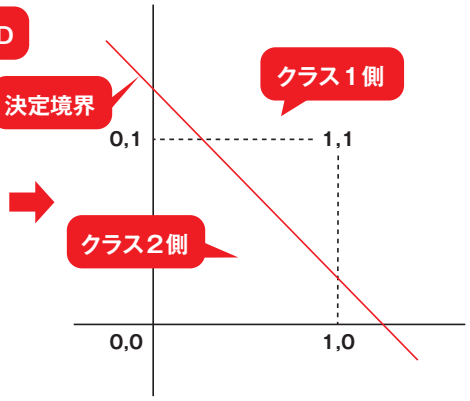


講師 では、図5のような学習データではどうでしょうか。

図5●XOR問題（その2）

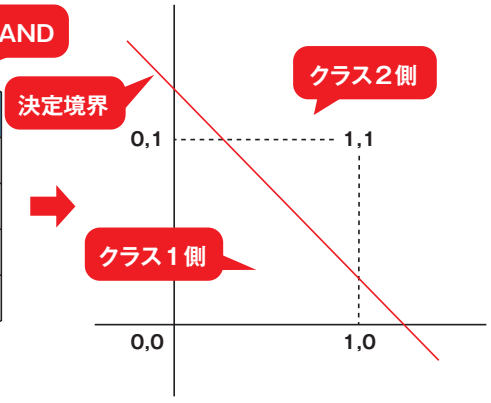
入力と正解ラベルの関係はAND

入力(x_1,x_2)	正解ラベル
0,0	0（クラス2）
1,0	0（クラス2）
0,1	0（クラス2）
1,1	1（クラス1）



入力と正解ラベルの関係はNAND

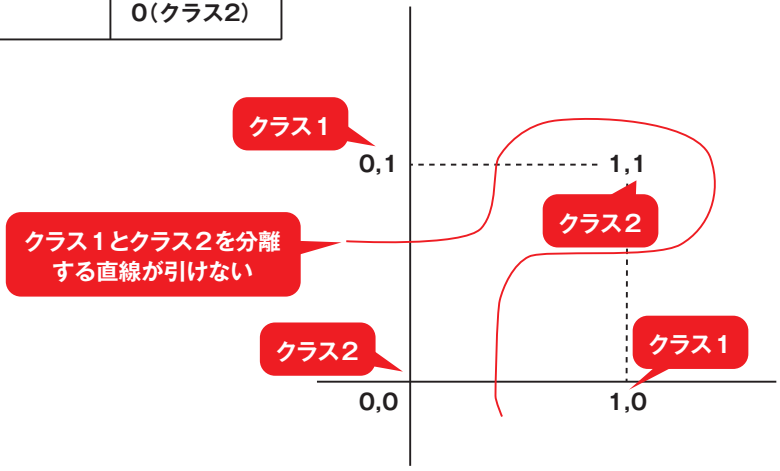
入力(x_1,x_2)	正解ラベル
0,0	1（クラス1）
1,0	1（クラス1）
0,1	1（クラス1）
1,1	0（クラス2）



講師 ANDとNANDは、分離するクラスが逆になるだけで、決定境界の直線は同じです。OR、AND、NANDときたら、最後はXORですね（**図6**）。

図6●XOR問題（その3）

入力（x_1,x_2）	正解ラベル
0,0	0（クラス2）
1,0	1（クラス1）
0,1	1（クラス1）
1,1	0（クラス2）



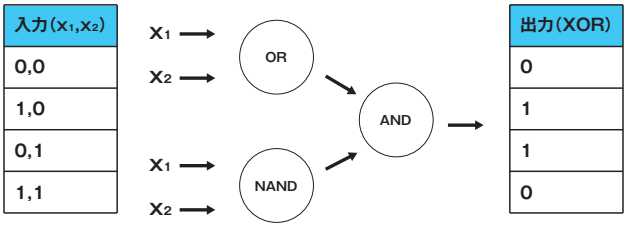
講師 このように、正解ラベルがXORになる場合、単純パーセプトロンのようなプログラムでは決定境界の直線を引くことができません。これが「XOR問題」です。ところが、OR、AND、NANDの判別をする単純パーセプトロンを多層組み合わせると、XORの識別が可能になります（**図7**）。

図7●XOR問題（その4）

x_1,x_2	OR
0,0	0
1,0	1
0,1	1
1,1	1

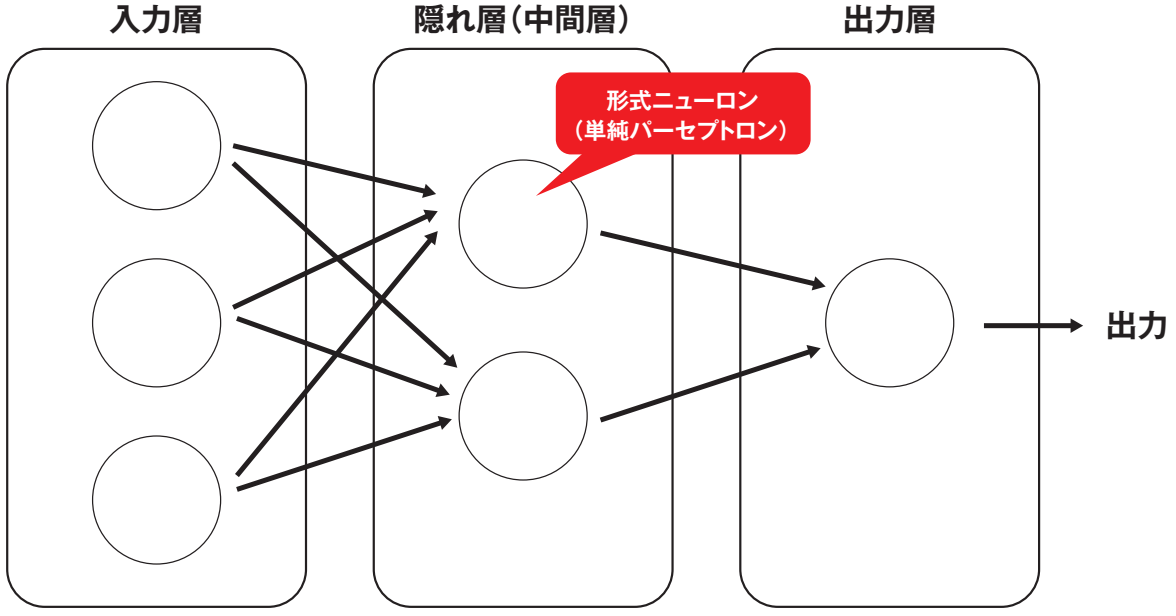
x_1,x_2	NAND
0,0	1
1,0	1
0,1	1
1,1	0

x_1,x_2	AND
0,0	0
1,0	0
0,1	0
1,1	1



講師 単純パーセプトロンを何層にも組み合わせることで、複雑な決定境界を引けるようにしたものを「多層パーセプトロン」と呼んでいます（図8）。

図8●多層パーセプトロン



愛 単純パーセプトロンの絆だから…。

進 単体でダメでも、複数ならできるっちゅうことじゃ。

講師 多層パーセプトロンでは、入力部分を「入力層」、出力部分を「出力層」、その間にあるパーセプトロンを「隠れ層」（中間層）と呼び、この層を2重、3重と増やすことで複雑な決定境界を引くことができるようになります。そして、このような多層パーセプトロンを利用した機械学習システムを「ニューラルネットワーク」と呼びます。

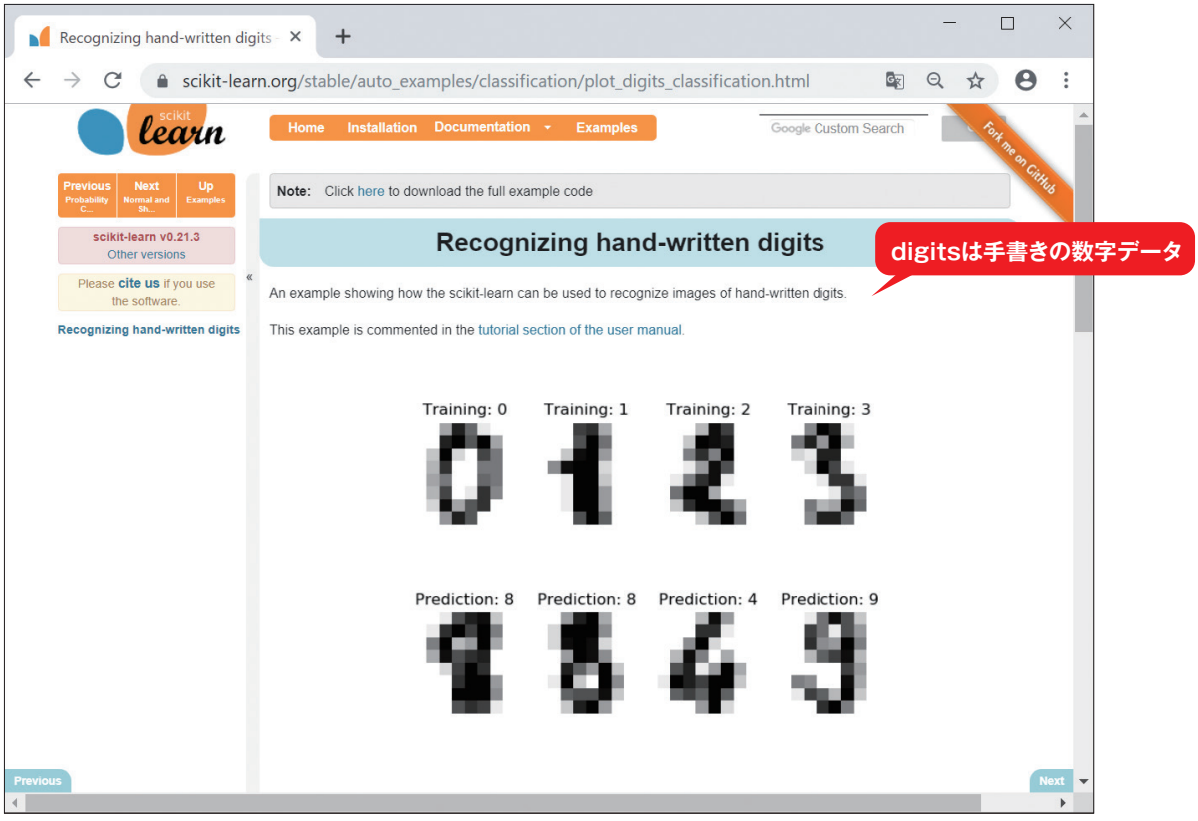
進 出た! ニューラルネットワーク。

講師 それでは、休憩後にscikit-learnでニューラルネットワークを試しましょう。

Part 2 多層パーセプトロン

講師 それでは、多層パーセプトロン(MLP)方式で機械学習を行ってみましょう。利用するデータは「digits」という手書き数字の画像データです(**図1**)。

図1●digitsデータセット
http://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html



進 この画像データを学習させれば、手書きした数字(0～9)を認識するんじゃな。

講師 このdigitsデータセットは、オリジナル版が「MNIST」という名前で「http://yann.lecun.com/exdb/mnist/」から入手できます。scikit-learnに付属しているデータセットは、MNISTの簡易版です。まずは、Spyderを起動して、digitsデータセットをロードしてみましょう。irisデータセットと同様に、load_digits関数が用意されています。

IPythonコンソール

```
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915
64 bit (AMD64)]
Type "copyright", "credits" or "license" for more informat
ion.

IPython 7.6.1 -- An enhanced Interactive Python.

In [1]: from sklearn.datasets import load_digits

In [2]: digits = load_digits()
```

digitsデータセットの読み込み

講師 エラーにならなければ、digitsデータセットが読み込めています。特徴データを確認しましょう。

IPythonコンソール

```
In [3]: print(digits.data)
[[ 0.  0.  5. ...  0.  0.  0.]
 [ 0.  0.  0. ... 10.  0.  0.]
 [ 0.  0.  0. ... 16.  9.  0.]
 ...
 [ 0.  0.  1. ...  6.  0.  0.]
 [ 0.  0.  2. ... 12.  0.  0.]
 [ 0.  0. 10. ... 12.  1.  0.]]

In [4]: print(digits.data.shape)
(1797, 64)
```

digitsデータセットの特徴データ

8×8=64のNumPyの配列が1797件分ある

進 画像データだけに、データの量が多いのじゃ!

講師 特徴データは1件が、8×8=64のNumPyの配列になっています。データ件数は、1797件あることがわかります。この1797件分の正解ラベル(0 ～ 9)は、irisデータセットと同様に、targetに入っています。

IPythonコンソール

```
In [5]: print(digits.target)
[0 1 2 ... 8 9 8]
```

digitsデータセットの正解ラベル

一番最初の画像は0の画像

講師 0 1 2 … の表示は、一番最初の画像は0の画像、次の画像は1の画像…と続くことを表しています。また、digits.imagesには8×8のピクセルデータが画像データとして格納されています。matplotlibをimportして、imshow関数でデータ内の一番最初の画像を表示してみましょう。

IPythonコンソール

グラフ描画ライブラリ

```
In [6]: import matplotlib.pyplot as plt

In [7]: plt.figure(figsize=(3, 3))
Out[7]: <Figure size 216x216 with 0 Axes><Figure size
216x216 with 0 Axes>

In [8]: plt.imshow(digits.images[0], cmap=plt.cm.gray_r)
Out[8]: <matplotlib.image.AxesImage at 0x2bb54488fd0>
```

3×3インチの新規ウインドウを作成

imshow関数で最初の画像を表示。
cmapにはカラーマップのグレイを指定

最初の画像データを表示

講師 これで、digitsがどのようなデータか大体わかりました。digitsデータセットも、irisデータセットと同様に、訓練用と評価用に分けてから機械学習を行います。

IPythonコンソール

```
In [9]: from sklearn.model_selection import train_test_spl
it

In [10]: X_train, X_test, y_train, y_test = train_test_spl
it(digits['data'], digits['target'], test_size=0.3,  rand
om_state=0)
```

訓練用7割、評価用3割に分割

講師 これで学習用データは用意できたので、多層パーセプトロン（MLP）方式で機械学習を行います。利用するのはMLPClassifierクラスです。コンストラクタの書式は**表1**のようになります。

表1●Perceptronの書式。引数は抜粋

MLPClassifier(hidden_layer_sizes=(100,), activation='relu', solver='adam', alpha=0.0001, batch_size='auto', learning_rate='constant', learning_rate_init=0.001, power_t=0.5, max_iter=200, shuffle=True, random_state=None, tol=0.0001, verbose=False, warm_start=False, momentum=0.9, nesterovs_momentum=True, early_stopping=False, validation_fraction=0.1, beta_1=0.9, beta_2=0.999, epsilon=1e-08)

引数	説明
hidden_layer_sizes	隠れ層の層の数とニューロンの数
activation	活性化関数 'identity', 'logistic', 'tanh', 'relu'
solver	最適化手法 'lbfgs', 'sgd', 'adam'
alpha	L2正則化のパラメータ
learning_rate_init	重みの学習率の初期値
learning_rate	重みの学習率の更新方法 'constant', 'invscaling', 'adaptive'

表1の続き

MLPClassifier(hidden_layer_sizes=(100,), activation='relu', solver='adam', alpha=0.0001, batch_size='auto', learning_rate='constant', learning_rate_init=0.001, power_t=0.5, max_iter=200, shuffle=True, random_state=None, tol=0.0001, verbose=False, warm_start=False, momentum=0.9, nesterovs_momentum=True, early_stopping=False, validation_fraction=0.1, beta_1=0.9, beta_2=0.999, epsilon=1e-08)	
引数	説明
max_iter	試行回数の最大値
shuffle	学習を反復するごとに学習データをシャッフルするかどうか
random_state	乱数のシード値
warm_start	2回目のfit関数を呼ぶ際、学習済みの重みを引き継ぐか否か

講師 まず、MLPClassifierの学習モデルを生成します。パラメータのmax_iter（試行回数の最大値）は、デフォルトの200では少なすぎるので1000に変更します。続いて、fit関数で訓練用のデータを読み込み、学習します。

進 機械学習の方法は、アヤメと同じじゃな。

IPythonコンソール

```
In [11]: from sklearn.neural_network import MLPClassifier

In [12]: mlpc = MLPClassifier(max_iter=1000)

In [13]: mlpc.fit(X_train, y_train)
Out[13]:
MLPClassifier(activation='relu', alpha=0.0001, batch_size='
auto', beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=
1e-08,
              hidden_layer_sizes=(100,), learning_rate='co
nstant',
              learning_rate_init=0.001, max_iter=1000, mom
```

試行回数の最大値を1000にする

訓練用データで学習開始

（次ページに続く）

（前ページからの続き）

```
entum=0.9,
                  n_iter_no_change=10, nesterovs_momentum=Tr
ue, power_t=0.5,
                  random_state=None, shuffle=True, solver='ad
am', tol=0.0001,
                    validation_fraction=0.1, verbose=False,
warm_start=False)
```

講師 これで学習が完了したので、精度を評価してみましょう。

IPythonコンソール

```
In [14]: pred = mlpc.predict(X_test)

In [15]: import numpy as np

In [16]: np.mean(pred == y_test)
Out[16]: 0.9722222222222222
```

正解ラベルと比較

正解率約97%

講師 正解率が悪い場合や、何回か実行して常に間違える画像がある場合は、パラメータを色々変更しながら試行錯誤する必要があるでしょう。今回の正解率は約97%でした。それから、scikit-learnには、confusion_matrix関数という便利な関数があります。このconfusion_matrixを使うと、縦軸に0 ～ 9までの正解を、横軸に学習モデルがその数字を予測した数を表示できます。これを見れば、どの数字で誤認識が発生しやすいのかや、どの数字と間違えやすいのかが一目でわかります。

IPythonコンソール

```
In [17]: from sklearn.metrics import confusion_matrix

In [18]: confusion_matrix(y_test, pred, labels=digits['tar
get_names'])
Out[18]:
array([[44,  0,  0,  0,  0,  0,  1,  0,  0,  0],
       [ 0, 48,  0,  0,  0,  0,  0,  0,  3,  1],
       [ 0,  1, 51,  0,  0,  0,  0,  0,  1,  0],
       [ 0,  0,  0, 52,  0,  0,  0,  0,  2,  0],
       [ 0,  0,  0,  0, 47,  0,  0,  1,  0,  0],
       [ 0,  0,  0,  0,  0, 52,  2,  0,  0,  3],
       [ 0,  1,  0,  0,  0,  0, 59,  0,  0,  0],
       [ 0,  0,  0,  0,  2,  0,  0, 50,  0,  1],
       [ 0,  2,  0,  0,  0,  0,  1,  0, 58,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0, 57]], dtype=i
nt64)
```

縦軸は0～9までの正解。横軸は学習モデルが予測したその数字の数

8を6と間違った

進 scikit-learnを使えば、いろんな機械学習アルゴリズムを簡単に試せるのう。

愛 ありがとう、scikit-learn。あの人にも言ったことなかったのに…。

講師 …栄井さんが、遠い目をしているので、休憩にしましょう。

進 …それがいいのじゃ。

5日目 Part 3 クラスタリング

講師 最後に「教師なし学習」を試してみましょう。教師なし学習と言えば、正解ラベルのないデータから共通の特徴を見つけてクラスタに分ける「クラスタリング」が有名です。scikit-learnには、cluster.KMeansクラスが用意されています。このクラスのアルゴリズムは、k平均法によるクラスタリングを実行します。書式は、**表1**のようになります。

表1●KMeansクラスのコンストラクタの書式。引数は抜粋

KMeans(n_clusters=8, init='k-means++', n_init=10, max_iter=300, tol=0.0001, precompute_distances='auto', verbose=0, random_state=None, copy_x=True, n_jobs=1)	
引数	説明
n_clusters	クラスタ数
max_iter	k-means アルゴリズムの最大反復回数
n_init	初期の重心を選ぶ処理の数
init	初期化方法 'k-means++' 'random' 'ndarray'
tol	収束判定に用いる許容可能誤差
precompute_distances	距離を事前に計算するか否か

愛 なんの平均?

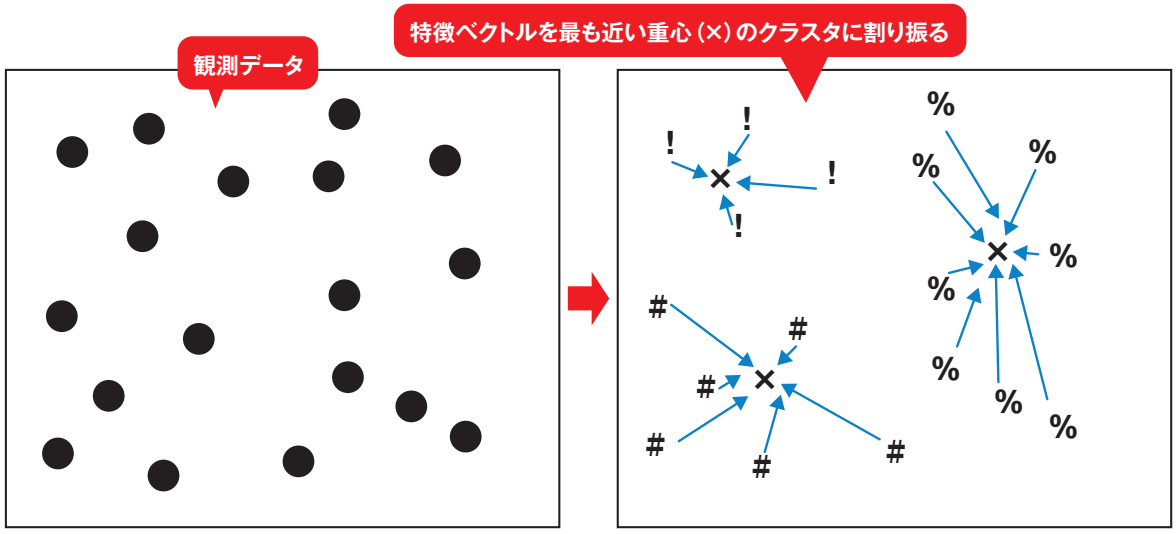
講師 k平均法は、非階層型クラスタリングと呼ばれています。非階層型クラスタリングでは、観測データをいくつに分割するのか、その数(クラスタ数)をあらかじめ指定します。

進 特徴データの中から共通点を見つけて、決められたグループに分けるっちゅうことじゃ。

講師 例えば、クラスタ数を「3」でクラスタリングを行うと、観測データを、似た特徴を持つ3つのグループ(クラスタ)に分割します。k平均法のアルゴリズムでは、特徴ベクトルを最

も近い重心のクラスタに割り振ることで分割を行います（図2）。

図2●k平均法



進 特徴をベクトルにすることで、計算がしやすくなるからのう。

講師 k平均法では、まず観測データをランダムなクラスタに割り当てます。そして、各クラスタごとに重心（座標の平均）を求めます。

愛 最初はランダムね…。

講師 そして、重心に一番近い観測データを、そのクラスタに変更し、再び重心を計算します。重心に変更があった場合は、また一番近い観測データを重心のクラスタに変更し、重心を求めるという作業を繰り返します。重心に変化がなくなるとクラスタリングが完了します。これが、k平均法のアルゴリズムです。

愛 同じ重心を持つ仲間…。

講師 それでは、irisデータセットを3つのクラスタに分類してみましょう。

進 irisデータは、そもそも3種類じゃから、正解データと一致すればOKじゃ。

講師 まず、irisデータセットをロードして、正解を確認しておきます。

IPythonコンソール

```
In [19]: from sklearn.datasets import load_iris

In [20]: iris_dataset = load_iris()

In [21]: iris_dataset['target']
Out[21]:
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2])
```

irisデータセットの読み込み

3種類のアヤメのデータが50個ずつ並んでいる

講師 irisデータセットには、3種類のアヤメのデータ（ラベル0×50個、ラベル1×50個、ラベル2×50個）が入っています。それでは、KMeansの学習モデルにirisの計測データだけを渡して、うまく3つのクラスタに分離できるか試しましょう。

IPythonコンソール

```
In [22]: from sklearn.cluster import KMeans

In [23]: kme = KMeans(n_clusters=3)

In [24]: kme.fit(iris_dataset['data'])
Out[24]:
KMeans(algorithm='auto', copy_x=True, init='k-means++',
max_iter=300,
       n_clusters=3, n_init=10, n_jobs=None, precompute_di
stances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

クラスタ数には3を指定

すべてのデータをそのままクラスタリング

講師 完了したらクラスタリングした結果のラベルを表示してみます。

IPythonコンソール

```
In [25]: kme.labels_
Out[25]:
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2,
       2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 1, 2, 1, 2, 1, 2, 2, 1, 1, 2, 2,
       2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 1])
```

クラスに分類した結果

間違い

1種類目（ラベル0）はうまくグループに分けることができたが、2種類目と3種類目（ラベル1と2）は間違いが多い

講師 ラベル番号は、最初に割り当てたランダムな番号なので正解とは多少異なりますが、取りあえず3分割はできています。

進 じゃが、あまりキレイに分けることができてないのう。教師なし学習は、ハードルが高そうじゃ。

愛 もっと強力なアルゴリズムはないの…。

講師 お二人とも、かなり機械学習の本質を理解できてきたようですね。研修は、これでおしまいですが、配属先でも今回の研修で得た経験をぜひ役立ててくださいね。

進 わかったぞい。わしは、魚の病気を画像で判断するシステムを考えるとするかのう。

愛 ワタシは、魚にQRコードを印刷する技術を開発する。

講師 夢が広がりますね。それでは、5日間お疲れ様でした。

■筆者紹介

中島 省吾

有限会社メディアプラネット代表取締役。テクニカルライターとして、ネットワークやプログラミング関連の記事を執筆するほか、IT企業向けのセミナーや新人研修の講師なども手掛ける。最近は、ボランティアで子どもたちにもプログラミングを教えている。近著に「ビジネスPython超入門」（日経BP）がある。

■表紙イラスト　ぷちめい

Pythonが5日でわかる本 AI基礎編

発行人●村上 広樹
編集長●久保田 浩
発行●日経BP
Nikkei Business Publications, Inc.
発売●日経BPマーケティング
〒105-8308 東京都港区虎ノ門4-3-12
URL●https://nkbp.jp/bpshopqa

©中島 省吾　日経BP　2019

■本書掲載記事の無断転載を禁じます。また無断複写・複製（コピー等）は著作権法上の例外を除き、禁じられています。購入者以外の第三者による電子データ化は、私的使用を含め一切認められておりません。詳しくは、ウェブサイト（https://nkbp.jp/copyright）をご参照ください。
日経BP