○[AI 自習帳 1]　機会学習ライブラリ「scikit-learn」の使い方（基本編）

amenbo the 3rd

2018.10.07

◆基本中の基本; AI プログラム（例）

本稿では、機会学習用ライブラリ「scikit-learn」の使い方を、基本的な例で解説します。

（Python コード（文法）の解説はしません、WEB 上に豊富にある情報を参照ください）

目次

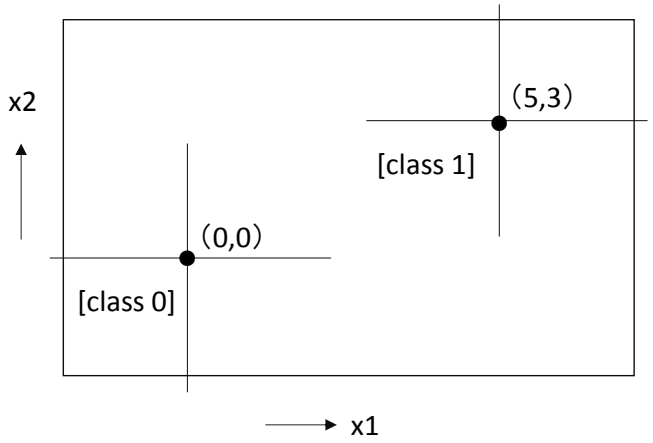
※本稿は、アメンボの学習用としてまとめたものです。（下記の記事を参考にして、若干変更）

（参照）「機械学習ライブラリ scikit-learn の便利機能の紹介」

https://qiita.com/ishizakiiii/items/0650723cc2b4eef2c1cf

1. 本稿で実現すること

先ず、[x1=0、x2=0]と[x1=5、x2=3]を中心とする乱数座標群を、100 個づつ生成します。



・[x1=0、x2=0]を中心とする乱数座標群は

⇒ [class 0] とし、

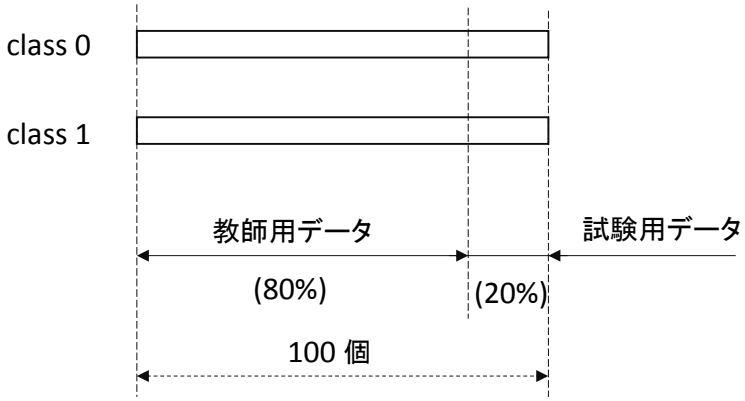
識別子[y=0]を付け足します。

・[x1=0、x2=0]を中心とする乱数座標群を

⇒ [class 1] とし、

識別子[y=1]を付け足します。

次に、[class 0]と[class 1]のデータを、「教師用データ」と「試験用データ」に分割します。



最後に、「教師用データ」を使って「機械学習（AI 手法）」を行い、その学習結果をベースとして「試験用データ」が、[class 0]と[class 1]のどちらのグループに属するのかを推定します。

＜参考＞

機械学習には、大別すると「回帰」と「分類」があります。

	予測する内容	予測内容
回帰	数値	どんな値になるのか？
分類	分類（クラス）	AかBの、どちらに属するのか？

本稿で使用する学習モデルは「分類」タイプです。

2. Python 用モジュール追加

※先ず、Python 側の準備が必要となります、下記に従ってライブラリをインストールしてください。

本稿で使うライブラリは、

```
「scikit-learn」　　；AI 用「機械学習ライブラリ」
「numpy」　　；数値計算ライブラリ
「pandas」　　；データ（1 次元、2 次元）処理ライブラリ
```

です。

※ただし、py コード上で呼出すときは、

```
scikit-learn　　⇒　from sklearn import **
numpy　　⇒　import numpy
pandas　　⇒　import pandas
```

と記述します。

下記の「pip（又は pip3)」コマンドを使いライブラリをインストールします。

```
pip install ライブラリ名
```

インストールした結果は、「pip list」で確認出来ます。

3. Python コード

※本 Python コードは「svm_03.py」として本稿に添付しています。
（コードの中には、アメンボが抱いた疑問やメモをそのまま残しています）

```
# -*- coding: utf-8 -*-
"""
Created on Fri May 11 02:11:55 2018
[svm_03.py]
@author: kenken
"""
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import numpy as np
#from numpy._distributor_init import NUMPY_MKL #
import pandas as pd
#
# class:0
df_a=pd.DataFrame({'x1':np.random.randn(100),
                   'x2':np.random.randn(100),
                   'y':0})
# class:1
df_b=pd.DataFrame({'x1':np.random.randn(100)+5,
                   'x2':np.random.randn(100)+3,
                   'y':1})
#
##OK#df=df_a.append(df_b)#これだと、行が連番にならない
df=pd.concat([df_a,df_b],ignore_index=True)
#
print("------ランダム•データ、生成順に表示-------")
print(df)
#print("==============")
#   生成されたデータを散布図で表示す
#
#   トレーニングデータとテストデータに分離する
X_train,X_test,y_train,y_test=train_test_split(df[['x1','x2']],df['y'],test_size=0.2)
#   モデルのインスタンスwp生成
clf=SVC()
#   fit   学習
```

・[x1=0, x2=0]と[x1=5, x2=3]を中心とする座標をランダムに生成し、それぞれ df_a [class:0]、df_b[class:1]としている。
・「y」はそれぞれが属するクラス、つまり「正解」を示す。

・df は、df_a と df_b をマージ（合成）したもの。

・df を、教師用データ（80%）と試験用データ(20%)に分ける

・scikit-learn で使える学習モデルの内、非線形な識別が可能と言われる SVC を指定してみた

```
clf.fit(X_train,y_train)
#   predict   予測
y_pred=clf.predict(X_test)
print("******予測結果******")
print(y_pred)
print("")
#   正解率の計算
#   方法1
print("正解率;計算方法1 ",accuracy_score(y_test, y_pred))
#   方法2
print("正解率;計算方法2 ",np.mean(y_pred==y_test))
print("")
#   結果を散布図で表示する
#1.先ず、X と y を結合する「列」方向
#(1)train データの結合
df_train=pd.concat([X_train,y_train],axis=1)
# axis=1 は「列」方向、axis=0 は「行」方向
print("------教師用ランダム・データ、生成順もランダム化-------")
print(df_train)
print("~~~~~~~~~~~~~~~~~~~~~~~~~")
#(2)test データの結合
df_test=pd.concat([X_test,y_test],axis=1)
print("------試験用ランダム・データ、生成順もランダム化-------")
print(df_test)
print("~~~~~~~~~~~~~~~~~~~~~~~~~")
#   動くけど、採用せず、だけど、これの方が良いみたい！！・・こっちでやってみる事
#2.y=0 と 1 のデータを分離「行」方向
#(1)train データの分離
#      A   y=0
df_train_0=df_train[df_train.y ==0]
print("------教師用データの内、[y=0]のもの-------")
print(df_train_0)
print("~~~~~~~~~~~~~~~~~~~~~~~~~")
#      B  y=1
df_train_1=df_train[df_train.y ==1]
print("------教師用データの内、[y=1]のもの-------")
print(df_train_1)
print("~~~~~~~~~~~~~~~~~~~~~~~~~")
#(2)test データの結合の分離
#      A   y=0
```

・機会学習を実行（コードはたった、これだけ！）

・試験用データの所属クラスを予測する、「0 か 1」？

・２通りの方法で、
正解率を計算してみる

・・ココから先は、
「データの表示」と「グラフ化」を
行う処理のみ

```
df_test_0=df_test[df_test.y ==0]
print("------試験用データの内、[y=0]のもの-------")
print(df_test_0)
print("~~~~~~~~~~~~~~~~~~~~~~~~~")
#     B y=1
df_test_1=df_test[df_test.y ==1]
print("------試験用データの内、[y=1]のもの-------")
print(df_test_1)
print("~~~~~~~~~~~~~~~~~~~~~~~~~")
#3.「4つ」の FrameData を散布図上に表示する
ax_1=df_train_0.plot(title='Random Data:[x1=0,x2=0] & [x1=5,x2=3]',x='x1',
y='x2',kind='scatter',color="red",label="train_0")
plt_1=df_train_1.plot(ax=ax_1,x='x1', y='x2', kind='scatter',color="blue",label='train_1')
plt_2=df_test_0.plot(ax=ax_1,x='x1', y='x2',
kind='scatter',marker='x',color="green",label='test_0')
plt_3=df_test_1.plot(ax=ax_1,x='x1', y='x2',
kind='scatter',marker='x',color="magenta",label='test_1')
plt_3.get_figure()
#  一時的にコメント化、動作 OK 品
##df_new.plot(x='x1', y='x2',kind='scatter',c=df_new.set_color)#OK なんと動いた
```

4.実行結果

※以下は、Spyder の IPython コンソール上への出力（表示）例です。

```
19    0.268022  -0.937487   0
20   -0.000783  -0.250401   0
21   -0.846048   0.645944   0
22   -0.683927   0.429744   0
23    0.309173  -0.400589   0
24   -0.796997  -2.755034   0
25    0.231647   0.856993   0
26   -0.402528  -0.547527   0
27   -0.614643  -1.007258   0
28    2.981603  -0.618210   0
29    0.894258   0.790368   0
..         ...        ... ..
170   5.541425   3.918820   1
171   7.106822   2.818566   1
172   5.818338   2.959447   1
173   4.684647   2.238710   1
174   5.490063   2.790717   1
175   6.164889   2.792870   1
176   5.976441   4.830100   1
177   4.684058   3.222981   1
178   4.443595   4.098760   1
179   4.791729   2.425365   1
180   5.557853   3.985055   1
181   5.448064   2.982960   1
182   3.914661   2.533990   1
183   4.828423   1.336010   1
184   3.976313  -0.351165   1
185   4.401184   2.356143   1
186   3.318089   3.811983   1
187   5.967959   3.897221   1
188   5.142879   1.551251   1
189   3.625524   3.653180   1
190   5.957404   4.186075   1
191   5.196530   2.837632   1
192   4.459085   2.764152   1
193   5.599929   2.757757   1
194   4.368720   3.687068   1
195   5.133278   1.978505   1
196   6.257291   2.160522   1
197   5.170111   2.954445   1
198   6.235098   1.742132   1
199   4.554081   5.629734   1

[200 rows x 3 columns]
******予測結果******
[0 0 1 ... 0 0 1]

正解率;計算方法1   1.0
正解率;計算方法2   1.0

------教師用ランダム・データ、生成順もランダム化-------
           x1        x2  y
84    1.072508  1.342741  0
37   -0.913427  0.052332  0
```

80	-1.014036	0.868621	0
112	6.276435	1.711985	1
42	1.676471	2.353844	0
56	0.607247	-0.996072	0
137	5.515826	3.036372	1
59	-0.102886	0.049354	0
154	5.232101	1.080281	1
101	2.494313	2.211040	1
194	4.368720	3.687068	1
158	6.013198	3.644314	1
50	-0.801237	-0.136858	0
182	3.914661	2.533990	1
138	4.336848	3.226577	1
65	-0.139687	-0.262880	0
3	-2.958598	0.444200	0
120	5.507787	2.603292	1
197	5.170111	2.954445	1
156	5.419380	3.729616	1
75	1.270227	0.566368	0
169	5.621123	4.207681	1
4	1.012786	-0.670444	0
97	0.747107	-0.882275	0
170	5.541425	3.918820	1
147	6.106141	2.394551	1
193	5.599929	2.757757	1
91	1.436203	-1.219628	0
94	0.025537	0.427293	0
20	-0.000783	-0.250401	0
..	...	...	..
191	5.196530	2.837632	1
143	5.622159	4.498221	1
82	-0.015135	-0.447052	0
25	0.231647	0.856993	0
119	4.229454	0.955831	1
186	3.318089	3.811983	1
184	3.976313	-0.351165	1
135	4.959128	3.505117	1
174	5.490063	2.790717	1
133	4.772080	2.918334	1
47	1.227121	-0.667077	0
86	-0.047023	0.613892	0
113	4.701724	3.289398	1
102	5.851574	2.659365	1
117	4.528491	3.943103	1
60	0.492340	0.455161	0
116	7.044828	3.093308	1
118	4.504786	3.199206	1
23	0.309173	-0.400589	0
96	-1.223740	-0.788729	0
63	-0.015751	1.236358	0
140	5.852368	2.435190	1
141	5.356038	1.651119	1
173	4.684647	2.238710	1
164	5.205348	3.698445	1
159	4.637253	3.905365	1
83	0.233430	1.304486	0
22	-0.683927	0.429744	0
162	3.838483	1.424911	1
87	0.312785	0.409590	0

[160 rows x 3 columns]

------試験用ランダム•データ、生成順もランダム化-------

	x1	x2	y
18	-0.499020	1.081150	0
43	1.279869	-0.407040	0
106	3.794254	0.439823	1
67	0.614923	-1.244792	0
181	5.448064	2.982960	1
51	-1.081999	0.882653	0
161	6.189347	5.412740	1
27	-0.614643	-1.007258	0
128	4.350675	4.462923	1
17	0.934493	0.295218	0
90	-0.345608	1.548628	0
107	4.066837	2.656768	1
150	3.085401	2.617152	1
24	-0.796997	-2.755034	0
58	1.404999	-2.243534	0
157	4.767034	3.288503	1
35	0.196299	0.814694	0
61	-0.231108	-0.008514	0
195	5.133278	1.978505	1
30	-1.531875	0.407854	0
124	4.939134	2.440870	1
166	5.823674	2.495720	1
5	0.183448	-1.020758	0
136	6.455605	2.752282	1
110	5.800494	4.499049	1
16	-1.631942	-0.449343	0
13	0.595398	-1.254619	0
160	5.651274	1.726304	1
142	4.186803	3.756483	1
198	6.235098	1.742132	1
40	-0.884352	0.268839	0
175	6.164889	2.792870	1
121	4.905535	4.095046	1
49	0.677822	-1.239316	0
21	-0.846048	0.645944	0
126	4.583618	1.638263	1
151	4.912178	3.713219	1
95	-1.007515	0.918856	0
19	0.268022	-0.937487	0
179	4.791729	2.425365	1

------教師用データの内、[y=0]のもの-------

	x1	x2	y
84	1.072508	1.342741	0
37	-0.913427	0.052332	0
80	-1.014036	0.868621	0
42	1.676471	2.353844	0
56	0.607247	-0.996072	0
59	-0.102886	0.049354	0
50	-0.801237	-0.136858	0
65	-0.139687	-0.262880	0
3	-2.958598	0.444200	0
75	1.270227	0.566368	0
4	1.012786	-0.670444	0
97	0.747107	-0.882275	0
91	1.436203	-1.219628	0

```
94   0.025537   0.427293   0
20  -0.000783  -0.250401   0
55   0.006145  -0.411445   0
28   2.981603  -0.618210   0
26  -0.402528  -0.547527   0
10  -0.970191   0.597251   0
69  -0.088949  -0.133209   0
12   1.073293   0.319995   0
74   0.071882  -0.324397   0
92  -0.105234  -0.502069   0
98  -0.430623  -0.918473   0
64   0.451747  -0.389080   0
53  -1.791435  -0.181722   0
79  -0.183757   0.629306   0
15   0.274348   1.312486   0
99  -0.544092  -0.940213   0
48  -1.367297   1.977855   0
..        ...        ...  ..
29   0.894258   0.790368   0
31  -1.040491   0.205569   0
33   0.282759  -1.241133   0
41  -0.138337  -0.842587   0
54  -0.980242  -0.081265   0
72   0.220571  -0.756728   0
52  -0.925358   1.221285   0
89   1.387968   1.140766   0
81  -0.360207   0.066044   0
8    1.626328   0.669497   0
34   1.295755   0.256180   0
2   -0.378520  -0.810432   0
78   0.137403   0.045911   0
68   2.410700  -1.142694   0
44  -1.292708  -0.764216   0
7    0.872353   0.050191   0
76   0.669582   0.058827   0
57  -0.627787   0.740314   0
46   0.677712   0.408627   0
82  -0.015135  -0.447052   0
25   0.231647   0.856993   0
47   1.227121  -0.667077   0
86  -0.047023   0.613892   0
60   0.492340   0.455161   0
23   0.309173  -0.400589   0
96  -1.223740  -0.788729   0
63  -0.015751   1.236358   0
83   0.233430   1.304486   0
22  -0.683927   0.429744   0
87   0.312785   0.409590   0

[80 rows x 3 columns]
```

------教師用データの内、[y=1]のもの-------

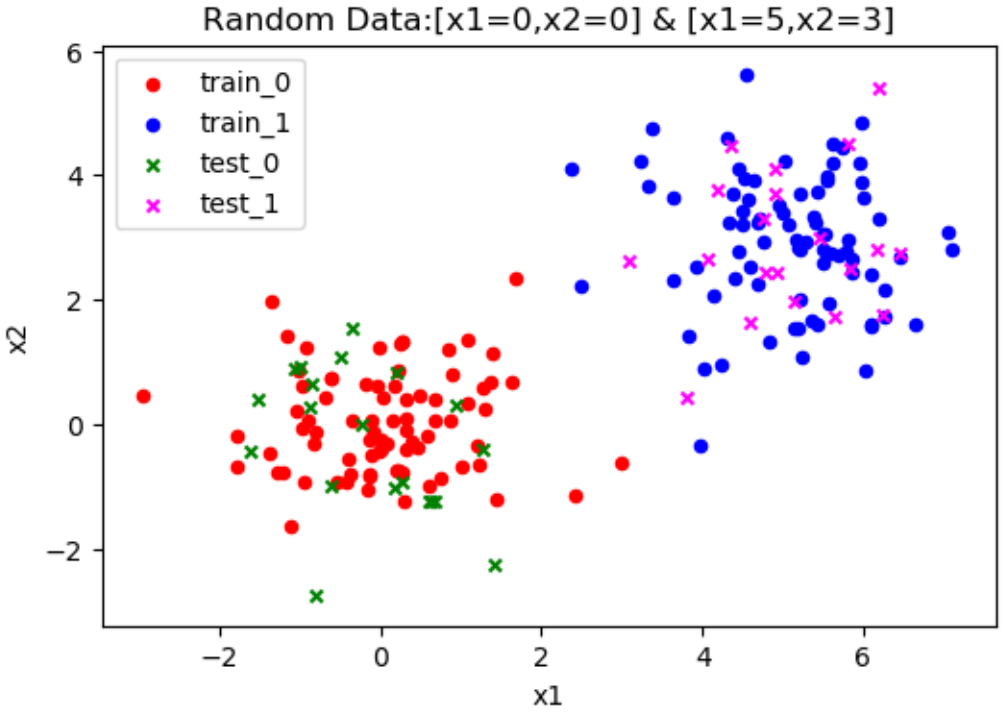
```
           x1         x2  y
112   6.276435   1.711985   1
137   5.515826   3.036372   1
154   5.232101   1.080281   1
101   2.494313   2.211040   1
194   4.368720   3.687068   1
158   6.013198   3.644314   1
182   3.914661   2.533990   1
```

```
138   4.336848   3.226577   1
120   5.507787   2.603292   1
197   5.170111   2.954445   1
156   5.419380   3.729616   1
169   5.621123   4.207681   1
170   5.541425   3.918820   1
147   6.106141   2.394551   1
193   5.599929   2.757757   1
172   5.818338   2.959447   1
115   5.783108   2.761422   1
152   3.634210   2.303131   1
199   4.554081   5.629734   1
122   5.179830   1.524560   1
196   6.257291   2.160522   1
188   5.142879   1.551251   1
125   6.096131   1.589118   1
146   5.747279   4.447672   1
105   6.027251   0.872729   1
144   5.282418   2.934797   1
103   5.007068   3.376559   1
148   5.429483   1.594972   1
167   5.203876   2.817756   1
187   5.967959   3.897221   1
..         ...        ... ..
134   4.021299   0.901900   1
123   6.094303   1.571091   1
185   4.401184   2.356143   1
168   6.185435   3.309066   1
178   4.443595   4.098760   1
114   5.011246   4.222152   1
171   7.106822   2.818566   1
145   5.070473   3.220634   1
129   4.591863   2.530268   1
180   5.557853   3.985055   1
192   4.459085   2.764152   1
191   5.196530   2.837632   1
143   5.622159   4.498221   1
119   4.229454   0.955831   1
186   3.318089   3.811983   1
184   3.976313  -0.351165   1
135   4.959128   3.505117   1
174   5.490063   2.790717   1
133   4.772080   2.918334   1
113   4.701724   3.289398   1
102   5.851574   2.659365   1
117   4.528491   3.943103   1
116   7.044828   3.093308   1
118   4.504786   3.199206   1
140   5.852368   2.435190   1
141   5.356038   1.651119   1
173   4.684647   2.238710   1
164   5.205348   3.698445   1
159   4.637253   3.905365   1
162   3.838483   1.424911   1

[80 rows x 3 columns]
~~~~~~~~~~~~~~~~~~~~~~~~~~~

------試験用データの内、[y=0]のもの-------
           x1         x2  y
18 -0.499020   1.081150   0
```

```
43   1.279869 -0.407040   0
67   0.614923 -1.244792   0
51 -1.081999   0.882653   0
27 -0.614643 -1.007258   0
17   0.934493   0.295218   0
90 -0.345608   1.548628   0
24 -0.796997 -2.755034   0
58   1.404999 -2.243534   0
35   0.196299   0.814694   0
61 -0.231108 -0.008514   0
30 -1.531875   0.407854   0
5    0.183448 -1.020758   0
16 -1.631942 -0.449343   0
13   0.595398 -1.254619   0
40 -0.884352   0.268839   0
49   0.677822 -1.239316   0
21 -0.846048   0.645944   0
95 -1.007515   0.918856   0
19   0.268022 -0.937487   0
~~~~~~~~~~~~~~~~~~~~~~~~~~
```

------試験用データの内、[y=1]のもの-------

```
           x1          x2   y
106  3.794254   0.439823   1
181  5.448064   2.982960   1
161  6.189347   5.412740   1
128  4.350675   4.462923   1
107  4.066837   2.656768   1
150  3.085401   2.617152   1
157  4.767034   3.288503   1
195  5.133278   1.978505   1
124  4.939134   2.440870   1
166  5.823674   2.495720   1
136  6.455605   2.752282   1
110  5.800494   4.499049   1
160  5.651274   1.726304   1
142  4.186803   3.756483   1
198  6.235098   1.742132   1
175  6.164889   2.792870   1
121  4.905535   4.095046   1
126  4.583618   1.638263   1
151  4.912178   3.713219   1
179  4.791729   2.425365   1
~~~~~~~~~~~~~~~~~~~~~~~~~~
```



※上記の散布図は、「class 0」と「class 1」として生成される
［教師用データ］と「試験用データ」の配置状況（例）を示します。

※［○］で示される教師用データで機会学習し、
［×］で示される試験用データが、「class 0」か「class 1」の何方に属するのかを
推測（判別）しています。

以　上