

Level3

採用したモデル: sklearn.svm.LinearSVC

クラス定義:

```
sklearn.svm.LinearSVC(penalty='l2', loss='squared_hinge', *, dual=True, tol=0.0001, C=1.0  
, multi_class='ovr', fit_intercept=True, intercept_scaling=1, class_weight=None, verbose=0  
, random_state=None, max_iter=1000)\[source\]
```

ハイパーパラメータの説明

C: float 型。Default = 1.0。正則化パラメータ。正則化の強さは C に反比例する。C は必ず正の値でなければならない。

max_iter: int 型。Default = 1000。実行する反復処理の最大数。

Level4

Code

```
import pandas as pd
from sklearn.svm import LinearSVC
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
df = pd.read_csv("beer.csv")
#print(df)
X = df[["OG", "ABV", "pH", "IBU"]]
Y = df["style"]
"""
print("X.shape = ", X.shape) shape -> (行数, 列数)をタプルで取得
print("Y.shape = ", Y.shape)
print(X.loc[0]) loc -> 行列と列名で位置を指定
print(Y[0])

X.shape = (150, 4)
Y.shape = (150,)
OG      11.5
ABV      4.2
pH       4.2
```


[illegible]

学習評価

C = 0.5: scores=[1. 0.96666667 1. 1. 1.],
average=0.993

C = 1.0 のとき

分類結果

[illegible]

[illegible]

学習評価

C = 1.5: scores=[0.93333333 1. 0.96666667 1. 1.],
average=0.980

Option

実行毎に average が異なる値を出すことより、さらに 10 回試行して C が 0.5, 1.0, 1.5 のときの average の平均をとって average の精度を上げ、C の値が 3 つのうちどの場合において一番精度が高いか調べてみた。

	1	2	3	4	5	6	7	8	9	10
C=0.5	0.993	0.987	0.993	0.987	0.993	0.987	0.993	0.993	0.987	0.987
C=1.0	0.987	0.987	0.987	0.993	0.987	0.980	0.993	0.980	0.993	0.987
C=1.5	0.980	0.987	0.987	0.967	0.987	0.993	0.993	0.987	0.993	1.000

従って、下の表より、 $C = 0.5$ のときが一番精度が高い結果となった。

	Average of aver
C = 0.5	0.990
C = 1.0	0.9874
C = 1.5	0.9874