

Machine learning

Classifieur SVM

Le dataset que nous allons utiliser ici est décrit dans : https://www.kaggle.com/raghupalem/bill_authentication (https://www.kaggle.com/raghupalem/bill_authentication).

Il s'agit pour faire simple d'une procédure d'authentification à partir d'images.

In [1]:

```
import pandas as pa
bankdata = pa.read_csv('https://www.labri.fr/perso/zemmari/datasets/bill_authentication.csv')
```

In [2]:

```
bankdata.head()
```

Out[2]:

	Variance	Skewness	Curtosis	Entropy	Class
0	3.62160	8.6661	-2.8073	-0.44699	0
1	4.54590	8.1674	-2.4586	-1.46210	0
2	3.86600	-2.6383	1.9242	0.10645	0
3	3.45660	9.5228	-4.0112	-3.59440	0
4	0.32924	-4.4552	4.5718	-0.98880	0

In [3]:

```
X = bankdata.drop('Class', axis=1)
y = bankdata['Class']
```

In [4]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.3, random_state=109)
```

In [5]:

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='linear')
svclassifier.fit(X_train, y_train)
```

Out[5]:

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='linear', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
```

In [6]:

```
y_pred = svcclassifier.predict(X_test)
```

In [7]:

```
from sklearn import metrics
scores = metrics.accuracy_score(y_test, y_pred)
print('Accuracy: ', '{:2.2%}'.format(scores))
```

Accuracy: 99.27%

In [8]:

```
cm = metrics.confusion_matrix(y_test, y_pred)
print(cm)
```

```
[[213  2]
 [ 1 196]]
```

Et si on comparait avec un classifieur bayésien :

In [9]:

```
from sklearn.naive_bayes import GaussianNB
clsb = GaussianNB()
clsb.fit(X_train, y_train)
y_pred = clsb.predict(X_test)
scores = metrics.accuracy_score(y_test, y_pred)
print('Accuracy: ', '{:2.2%}'.format(scores))
```

Accuracy: 83.25%

In []: