IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

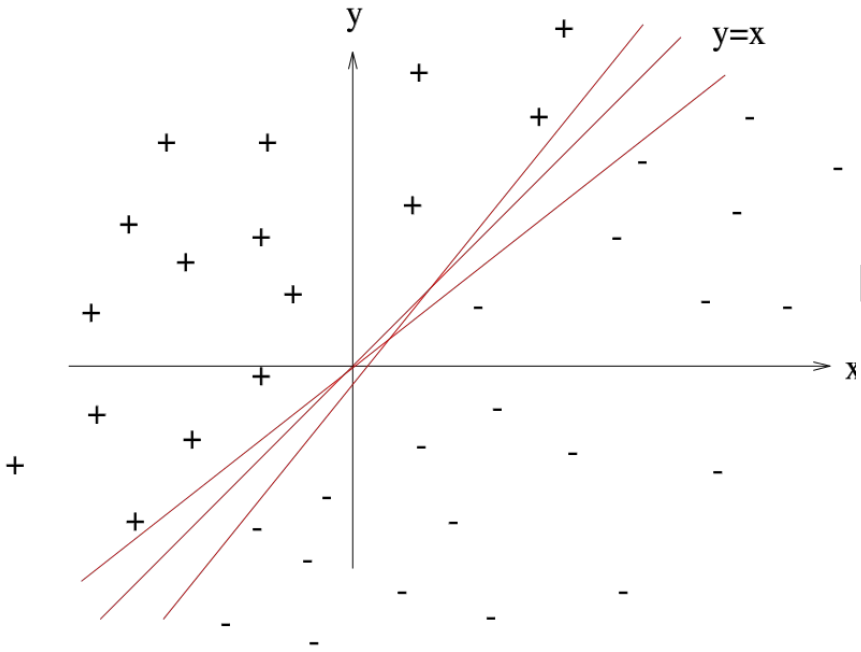
PROJECT P1
SVM CLASSIFIER

SUMMARY

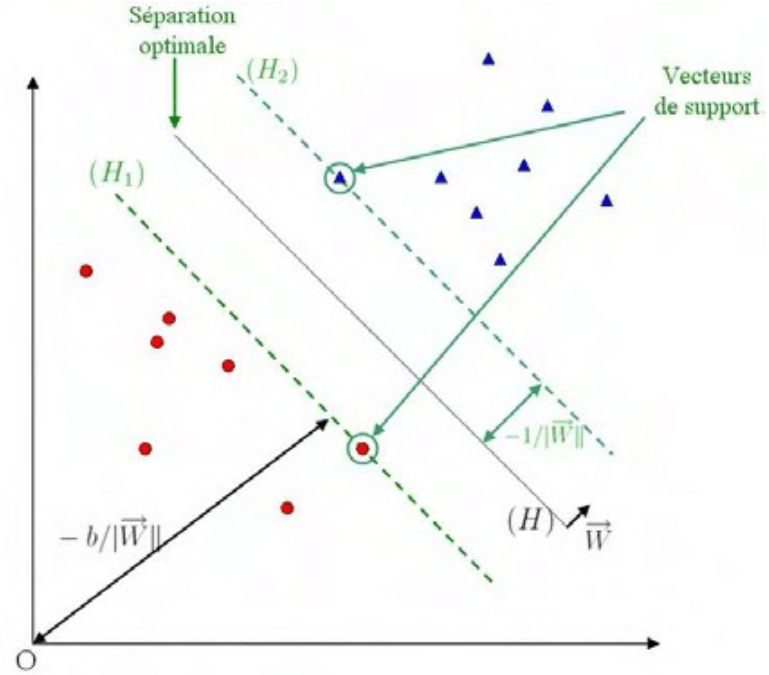
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

CHAPTER 1
HOW DOES IT WORK ?

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom



Multiple solutions for separating datas.



SVM = Support vector Machine

SVM Principe :

- Use Support Vector
- Find the optimal separation (hyperplan) of the datas by maximising the separation.

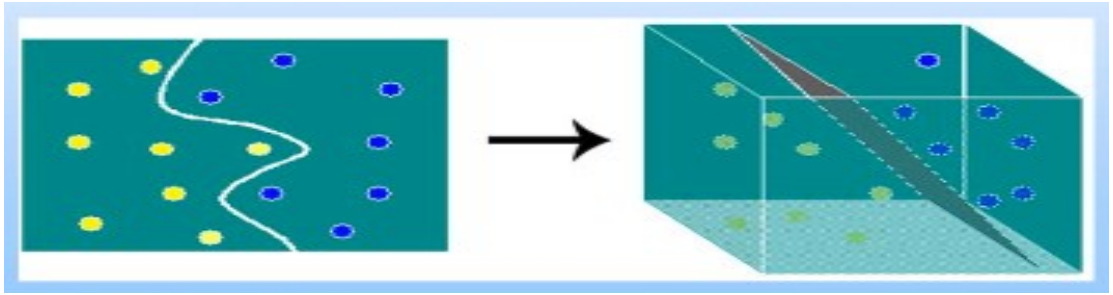
Problem : Data separation must be linear.

SVM : classifier = SVC(kernel='poly', degree=*int, C=C)*

First : linearisation of the datas

→ Choose the kernel (linear, polynomial, sigmoid, rbf, …)

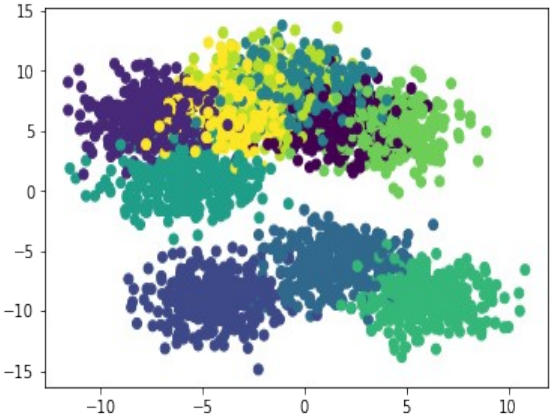
→ Choose the parameter (degree of the polynom, error range C, etc.)



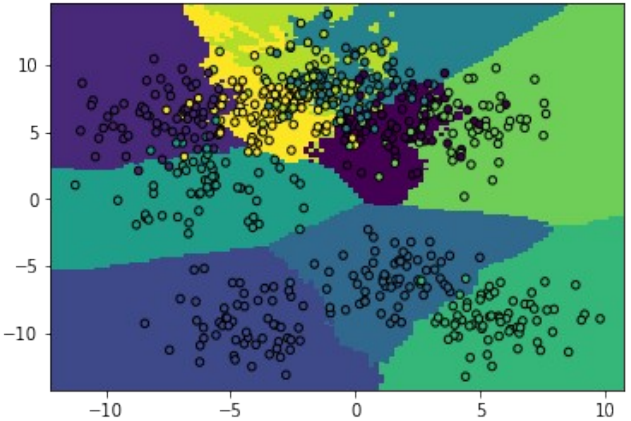
Then : train the classifier as usual

CHAPTER 2
BLOB EXPERIMENTATION

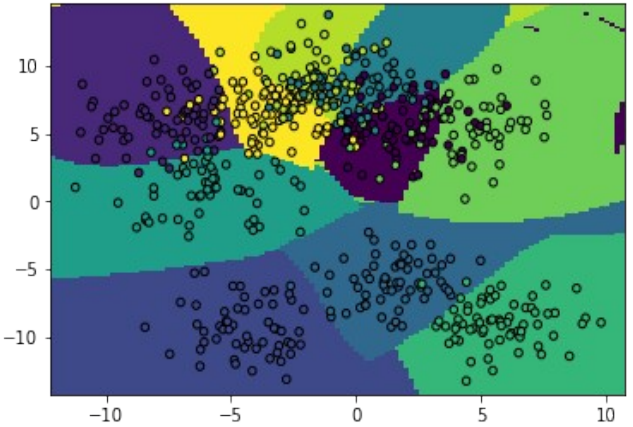
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom



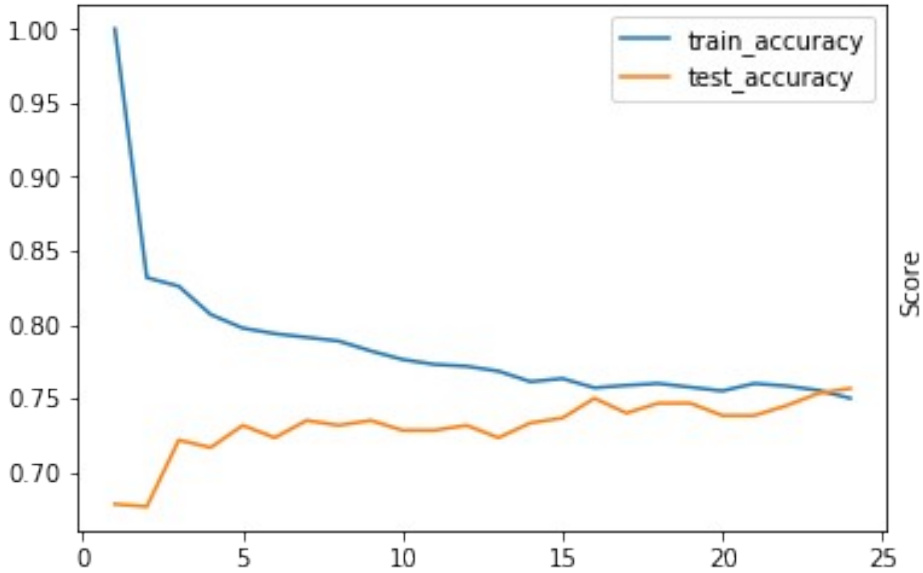
K-Kneighbours : Score = 0,73



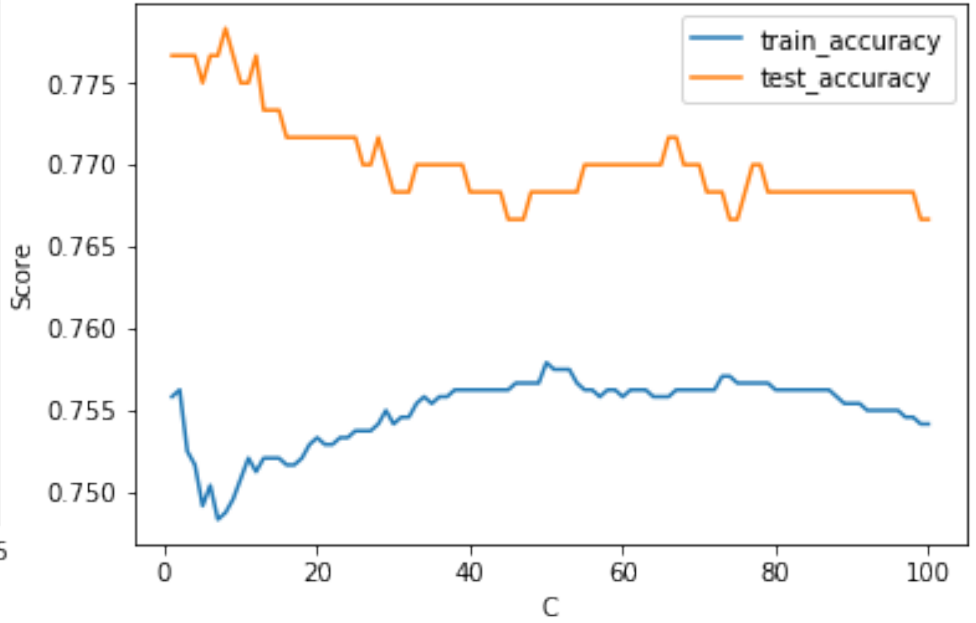
SVM_RBF : Score = 0,77



K-Kneighbours influence of k



SVM_RBF influence of C



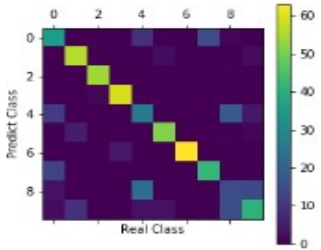
K-Kneighbours Matrix

Prediction for test

	precision	recall	f1-score	support
0	0.55	0.58	0.56	60
1	0.80	0.93	0.86	60
2	0.98	1.00	0.99	54
3	0.94	0.98	0.96	60
4	0.43	0.44	0.43	59
5	0.91	0.89	0.90	57
6	1.00	0.94	0.97	67
7	0.74	0.76	0.75	55
8	0.31	0.27	0.29	55
9	0.66	0.56	0.61	73
micro avg	0.74	0.74	0.74	600
macro avg	0.73	0.74	0.73	600
weighted avg	0.73	0.74	0.73	600

```
[[35  0  0  0  9  0  0 15  1  0]
 [ 0 56  0  0  0  2  0  0  0  2]
 [ 0  0 54  0  0  0  0  0  0  0]
 [ 0  0  1 59  0  0  0  0  0  0]
 [11  0  0  0 26  0  0  0 18  4]
 [ 0  5  0  0  0 51  0  0  0  1]
 [ 0  0  0  4  0  0 63  0  0  0]
 [13  0  0  0  0  0  0 42  0  0]
 [ 3  0  0  0 23  0  0  0 15 14]
 [ 2  9  0  0  3  3  0  0 15 41]]
```

Out[11]: Text(0, 0.5, 'Predict Class')



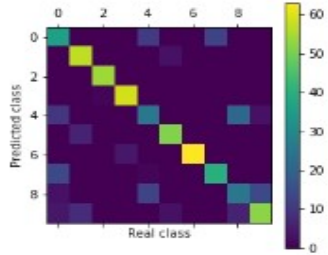
SVM_RBF Matrix

Test Set:

	precision	recall	f1-score	support
0	0.54	0.60	0.57	60
1	0.80	0.95	0.87	60
2	0.98	1.00	0.99	54
3	0.94	0.98	0.96	60
4	0.50	0.42	0.46	59
5	0.89	0.89	0.89	57
6	1.00	0.94	0.97	67
7	0.75	0.73	0.74	55
8	0.48	0.45	0.47	55
9	0.75	0.71	0.73	73
micro avg	0.77	0.77	0.77	600
macro avg	0.76	0.77	0.77	600
weighted avg	0.77	0.77	0.77	600

```
[[36  0  0  0 11  0  0 13  0  0]
 [ 0 57  0  0  0  3  0  0  0  0]
 [ 0  0 54  0  0  0  0  0  0  0]
 [ 0  0  1 59  0  0  0  0  0  0]
 [10  0  0  0 25  0  0  0 21  3]
 [ 0  6  0  0  0 51  0  0  0  0]
 [ 0  0  0  4  0  0 63  0  0  0]
 [14  0  0  0  1  0  0 40  0  0]
 [ 3  0  0  0 13  0  0  0 25 14]
 [ 4  8  0  0  0  3  0  0  6 52]]
```

Out[14]: Text(0, 0.5, 'Predicted class')



CHAPTER 3
PYRAT EXPERIMENTATION

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Context :

- 11000 games set ;
- Same number of draw, py win and rat win

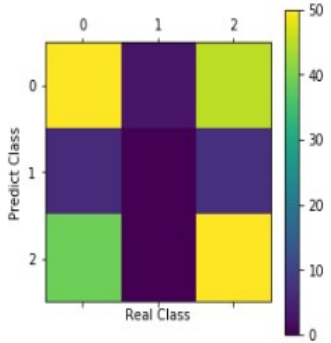
Prediction for test
Score Test: 0.5

	precision	recall	f1-score	support
-1.0	0.53	0.51	0.52	98
0.0	0.00	0.00	0.00	13
1.0	0.49	0.56	0.52	89
micro avg	0.50	0.50	0.50	200
macro avg	0.34	0.36	0.35	200
weighted avg	0.48	0.50	0.49	200

```
[[50  3 45]
 [ 6  0  7]
 [39  0 50]]
```

Without the same number of win, we had :

Out[48]: Text(0, 0.5, 'Predict Class')



K-Kneighbours result

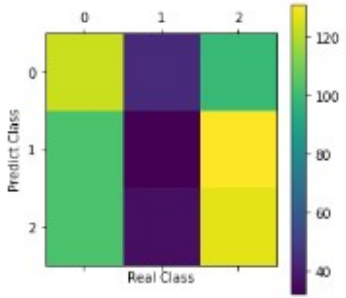
```
Prediction for test
Score Test: 0.3533834586466165
              precision    recall  f1-score   support

        -1.0       0.37      0.46      0.41       266
         0.0       0.29      0.12      0.17       266
         1.0       0.36      0.48      0.41       266

   micro avg       0.35      0.35      0.35       798
   macro avg       0.34      0.35      0.33       798
weighted avg       0.34      0.35      0.33       798

[[123  44  99]
 [103  32 131]
 [103  36 127]]
```

Out[78]: Text(0, 0.5, 'Predict Class')



SVM using RBF result

```
Score for rbf SVM : 0.39348370927318294
Prediction for test
              precision    recall  f1-score   support

        -1.0       0.41      0.42      0.42       266
         0.0       0.31      0.28      0.30       266
         1.0       0.44      0.48      0.46       266

   micro avg       0.39      0.39      0.39       798
   macro avg       0.39      0.39      0.39       798
weighted avg       0.39      0.39      0.39       798

[[111  92  63]
 [ 91  75 100]
 [ 66  72 128]]
```

Out[79]: Text(0, 0.5, 'Predict Class')