Τσιατούρας Βαγγέλης, 1115201200185

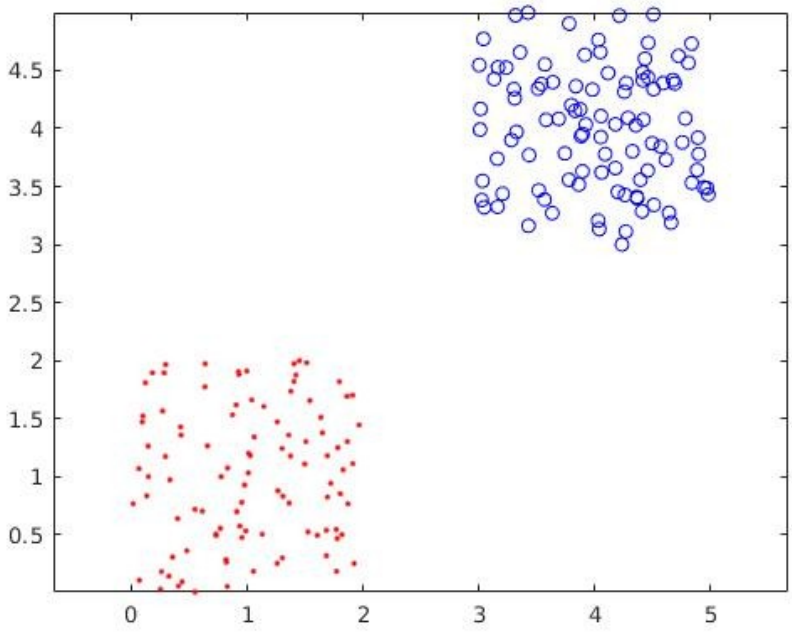
Ιούνιος 2, 2018

2η ΠΡΟΑΙΡΕΤΙΚΗ ΕΡΓΑΣΙΑ ΑΝΑΓΝΩΡΙΣΗΣ ΠΡΟΤΥΠΩΝ & ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

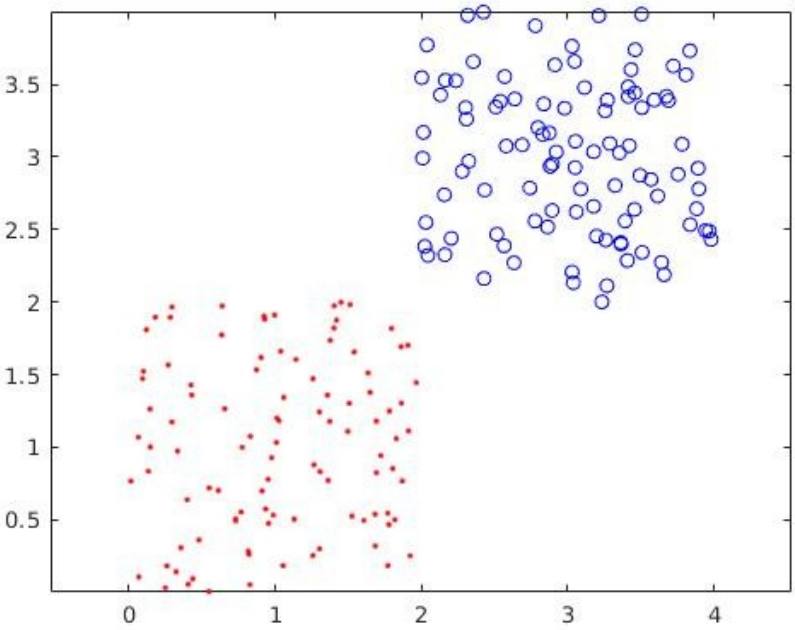
1η Άσκηση

(1)

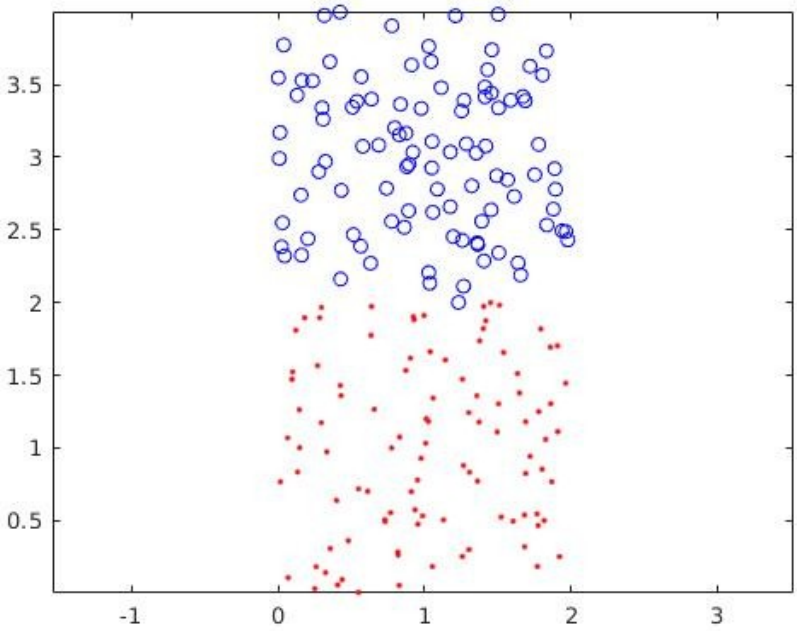
X1



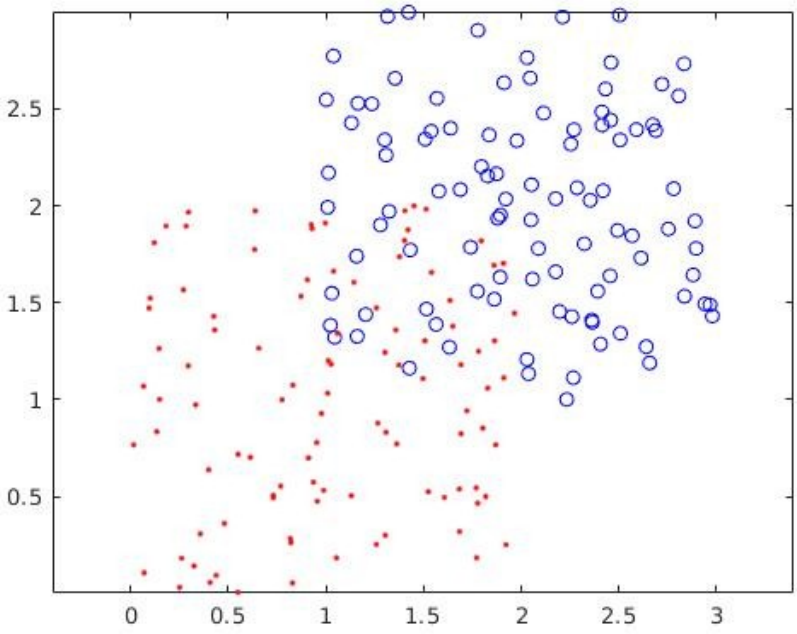
X2



X3



X4



(2)

Αποτελέσματα εκτέλεσης Perceptron για ρ=0.01:

```
######################
          X1
######################
First Iteration: # Misclassified points = 100
w = 0.0965
    0.0940
   -0.3451
iter = 109
mis_clas = 0

######################
          X2
######################
First Iteration: # Misclassified points = 100
w = 0.0965
    0.0940
   -0.3451
iter = 109
mis_clas = 0

######################
          X3
######################
First Iteration: # Misclassified points = 100
w = 0.0014
    0.1617
   -0.3252
```

```
iter = 1891
mis_clas = 0

######################
          X4
######################
First Iteration: # Misclassified points = 100
w = 0.0116
    0.0132
   -0.0440
iter = 20000
mis_clas = 30
```

Αποτελέσματα εκτέλεσης Perceptron για ρ=0.05:

```
######################
          X1
######################
First Iteration: # Misclassified points = 100
w = 0.2401
    0.2310
   -0.8900
iter = 10
mis_clas = 0

######################
          X2
######################
First Iteration: # Misclassified points = 100
w = 0.1765
    0.1662
```

```
   -0.7225
iter = 10
mis_clas = 0

######################
          X3
######################
First Iteration: # Misclassified points = 100
w =-0.0103
    0.3839
   -0.7525
iter = 57
mis_clas = 0

######################
          X4
######################
First Iteration: # Misclassified points = 100
w = 0.3030
    0.3370
   -1.1300
iter = 20000
mis_clas = 27
```

(3)

Αποτελέσματα εκτέλεσης Perceptron για ρ=0.05 για διανύσματα παραμέτρων $[1, 1, -0.5]^T$ και $[1, 1, 0.5]^T$:

```
######################
       [1 1 0.5]T
######################
 First Iteration: # Misclassified points = 100
w =-0.0103
    0.3839
   -0.7525
iter = 57
mis_clas = 0

######################
      [1 1 -0.5]T
######################
First Iteration: # Misclassified points = 95
w =-0.0002
    0.5364
   -1.0725
iter = 252
mis_clas = 0
```

(4)

Τα συμπεράσματα που προκύπτουν είναι τα εξής. Για μεγαλύτερο ρ ο αλγόριθμος ταξινομεί και πιο γρήγορα (λιγότερες επαναλήψεις) και με μεγαλύτερη ακρίβεια (λιγότερα misclassified points). Όσον αφορά τα διαφορετικά διανύσματα παραμέτρων ο αλγόριθμος για διάνυσμα $[1, 1, 0.5]^T$ πραγματοποίησε λιγότερες επαναλήψεις (57 iterations) σε σχέση με το διάνυσμα $[1, 1, -0.5]^T$ όπου πραγματοποίησε 252 επαναλήψεις.

2η Άσκηση

(1,2)

Χρησιμοποιώντας τον δοθέν κώδικα παράγουμε τα σύνολα και εφαρμόζουμε τους ταξινομητές για να παράγουμε τα αποτελέσματα.

(3)

Εκτελώντας το πρόγραμμα παίρνουμε τα εξής αποτελέσματα:

```
##################
      N=200
##################
err_Bayes_true = 0.1400
err_SSE = 0.1500

##################
     N=10000
##################
err_Bayes_true = 0.1468
err_SSE = 0.1498

##################
     N=100000
##################
err_Bayes_true = 0.1467
err_SSE = 0.1474
```

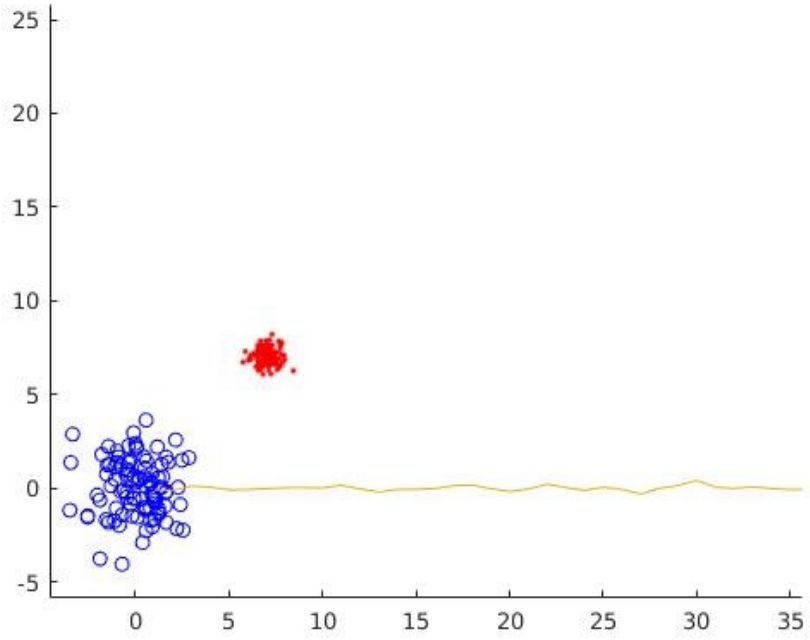
Παρατηρούμε ότι όσο το σύνολο δεδομένων X2 μεγαλώνει το σφάλμα του ταξινομητή ελάχιστου αθροίσματος τετραγωνικών σφαλμάτων (SSE) το ποσοστό λανθασμένης ταξινόμησης μειώνεται σε σημείο που τείνει να φτάσει τις επιδόσεις του Bayes.

3η Άσκηση

(1)

Για την υλοποίηση του ζητούμενου έγινε η χρήση του παρακάτω κώδικα:

```
close('all');
clear;

m(:,1)=[0 0]';
m(:,2)=[7 7]';

S1=2*eye(2);
S2=0.2*eye(2);

P=[1/2 1/2];

% Generate X1 and the required class labels
N1=200;
randn('seed',0)
X1=[mvnrnd(m(:,1),S1,fix(N1/2)); mvnrnd(m(:,2),S2,N1-fix(N1/2))]';
z1=[ones(1,fix(N1/2)) -1*ones(1,N1-fix(N1/2))];

% Generate X2 and the required class labels
N2=200;
%N2=10000 % for X3
%N2=100000 % for X4
randn('seed',100)
X2=[mvnrnd(m(:,1),S1,fix(N2/2)); mvnrnd(m(:,2),S2,N2-fix(N2/2))]';
z2=[ones(1,fix(N2/2)) -1*ones(1,N2-fix(N2/2))];


% Compute the Bayesian classification error based on X2
S_true(:,:,1)=S1;
S_true(:,:,2)=S2;
[z]=bayes_classifier(m,S_true,P,X2);
err_Bayes_true=sum(z~=z2)/sum(N2)

% Compute the classification error of the LS classifier based on X2
[w]=SSErr(X1,z1,0);
SSE_out=2*(w'*X2>0)-1;
err_SSE=sum(SSE_out.*z2<0)/sum(N2)

figure(1)
hold on
scatter(X1(1,z1==1),X1(2,z1==1),'bo')
scatter(X1(1,z1==-1),X1(2,z1==-1),'r.')
plot(w'*X1)
hold off
figure(1), axis equal
```



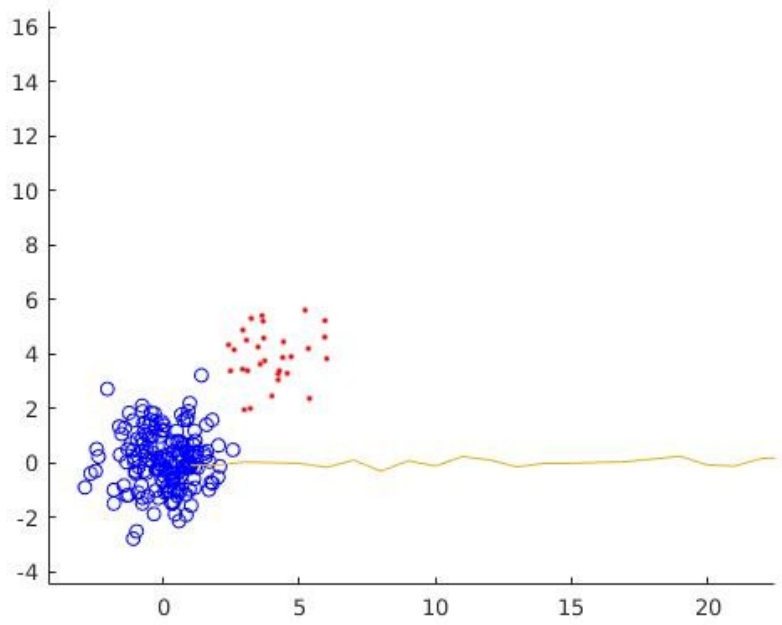

```
err_Bayes_true = 0.5000
err_SSE = 0.2750
```

(2)

Για την υλοποίηση του ζητούμενου έγινε η χρήση του παρακάτω κώδικα:

```
close('all');
clear;

m(:,1)=[0 0]';
m(:,2)=[4 4]';

S=eye(2);

P=[1/2 1/2];

% Generate X1 and the required class labels
N1=200;
randn('seed',0)
X1=[mvnrnd(m(:,1),S,170); mvnrnd(m(:,2),S,30)]';
z1=[ones(1,170) -1*ones(1,30)];

% Generate X2 and the required class labels
N2=200;
%N2=10000 % for X3
%N2=100000 % for X4
randn('seed',100)
X2=[mvnrnd(m(:,1),S,170); mvnrnd(m(:,2),S,30)]';
```

```
z2=[ones(1,170) -1*ones(1,30)];


% Compute the Bayesian classification error based on X2
S_true(:,:,1)=S;
S_true(:,:,2)=S;
[z]=bayes_classifier(m,S_true,P,X2);
err_Bayes_true=sum(z~=z2)/sum(N2)

% Compute the classification error of the LS classifier based on X2
[w]=SSErr(X1,z1,0);
SSE_out=2*(w'*X2>0)-1;
err_SSE=sum(SSE_out.*z2<0)/sum(N2)

figure(2)
hold on
scatter(X1(1,z1==1),X1(2,z1==1),'bo')
scatter(X1(1,z1==-1),X1(2,z1==-1),'r.')
plot(w'*X1)
hold off
figure(2), axis equal
```

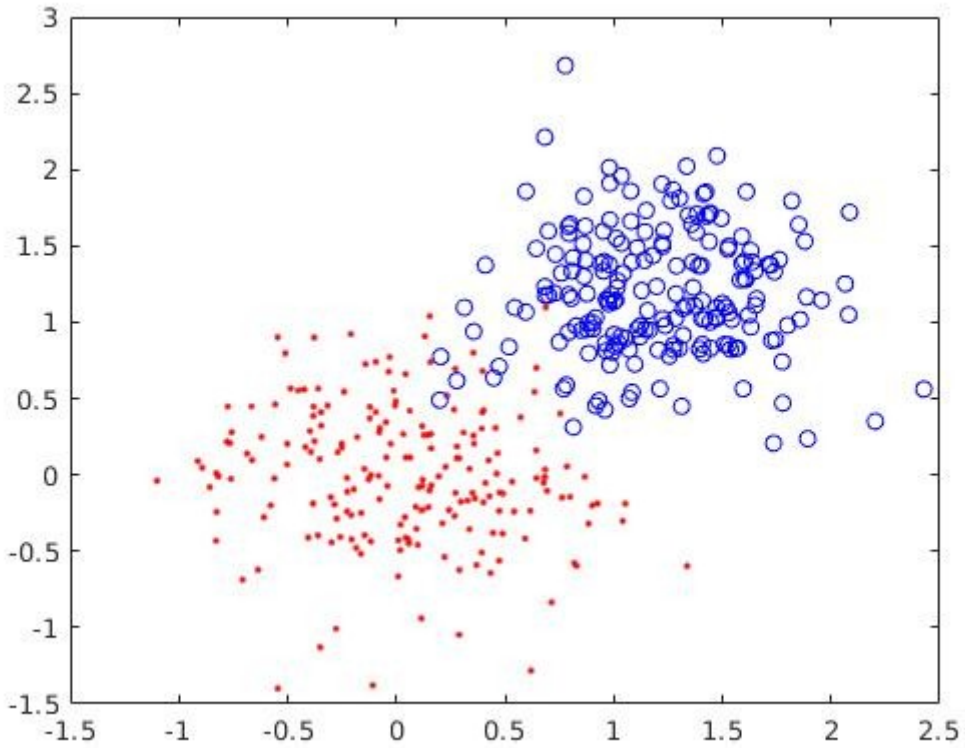



```
err_Bayes_true = 0.1550
err_SSE = 0.4350
```

4η Άσκηση

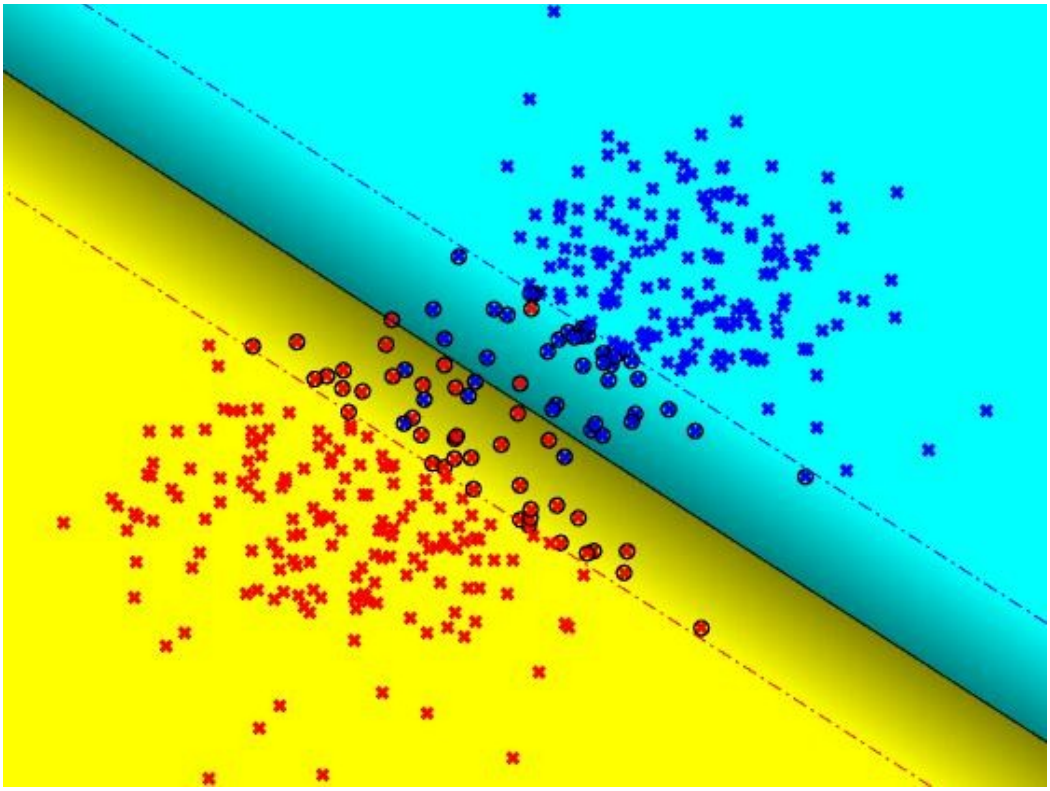
(1)

Χρησιμοποιώντας τον δοθέν κώδικα παράγουμε τα εξής σύνολα:

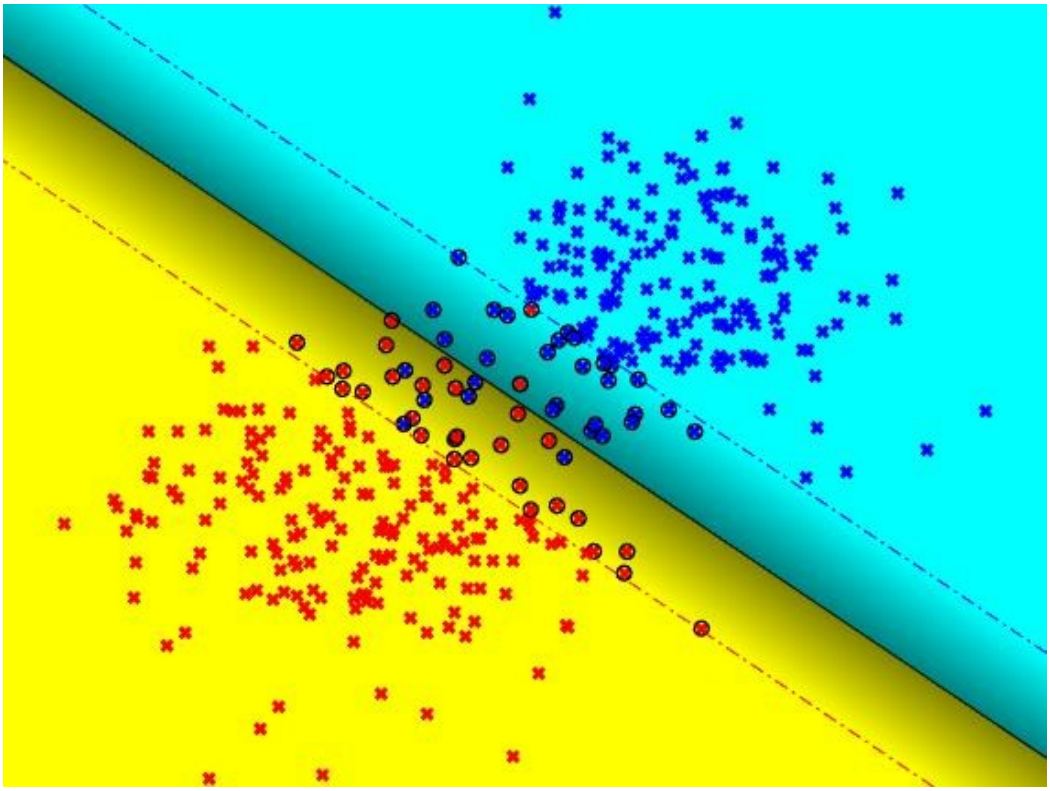


(2)

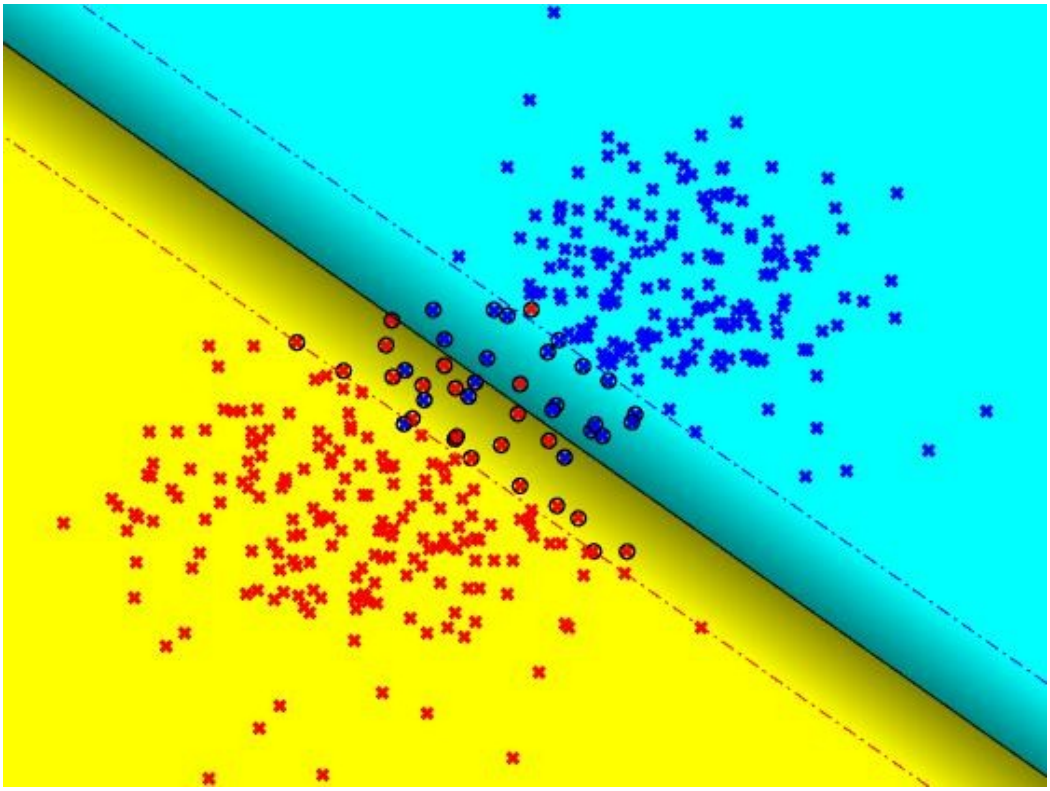
```
########################
      C = 0.1
########################
Pe_tr = 0.0225
Pe_te = 0.0325
sup_vec = 82
marg = 0.9414
```



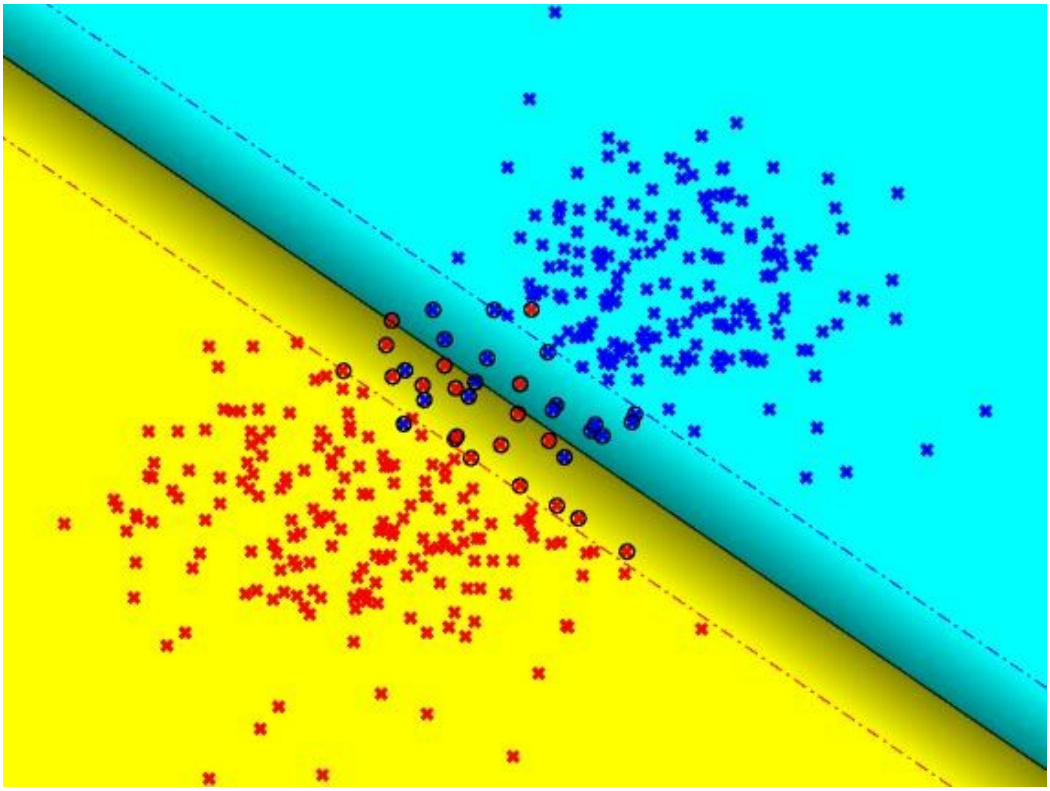
```
########################
          C = 0.2
########################
Pe_tr = 0.0200
Pe_te = 0.0300
sup_vec = 60
marg = 0.8148
```



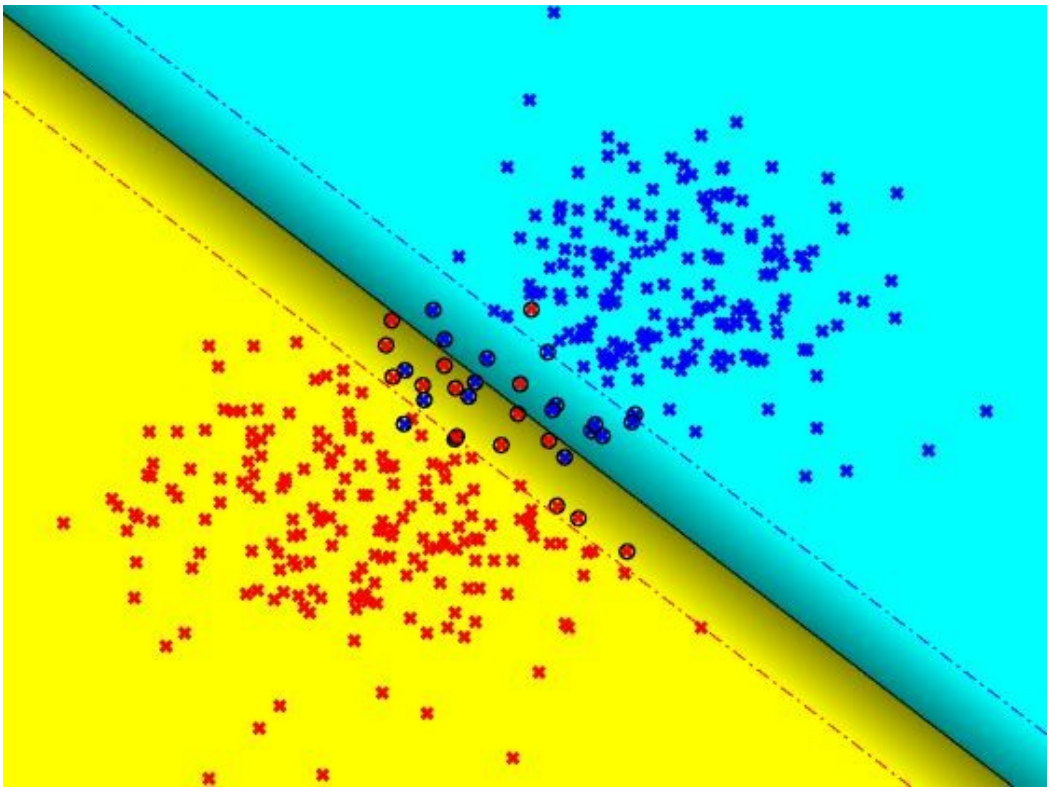
```
########################
          C = 0.5
########################
Pe_tr = 0.0200
Pe_te = 0.0325
sup_vec = 44
marg = 0.7085
```



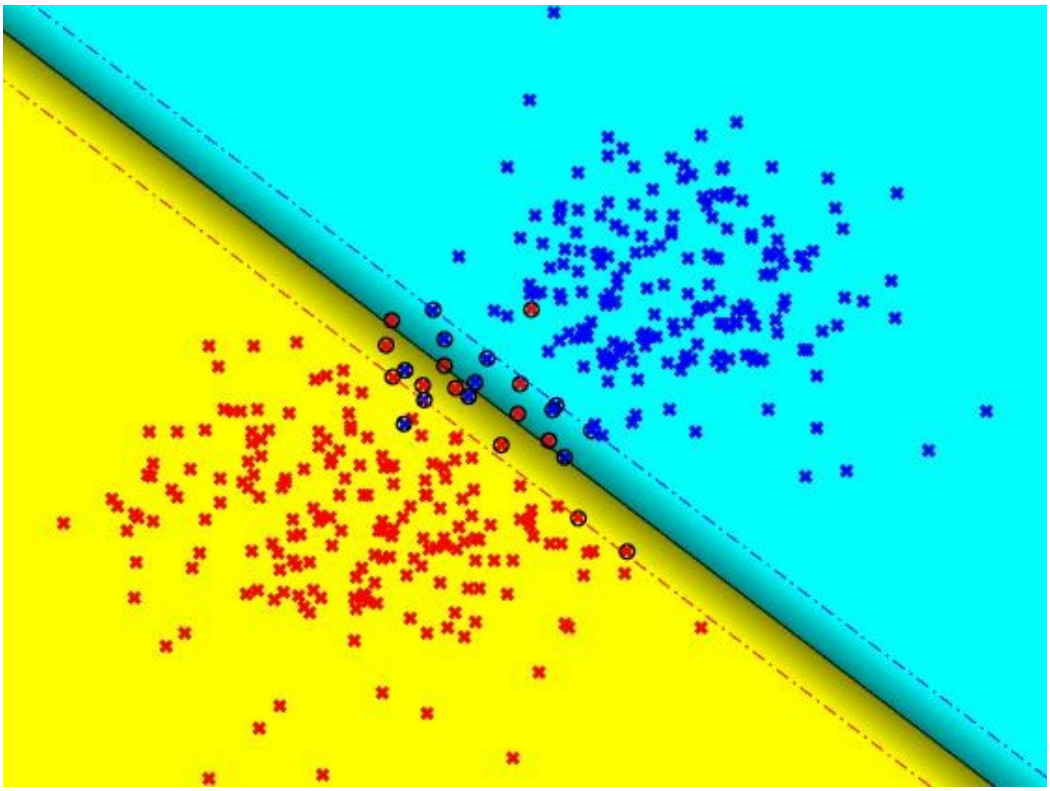
```
########################
          C = 1
########################
Pe_tr = 0.0225
Pe_te = 0.0325
sup_vec = 37
marg = 0.6318
```



```
########################
          C = 2
########################
Pe_tr = 0.0200
Pe_te = 0.0325
sup_vec = 33
marg = 0.5700
```



```
########################
          C = 20
########################
Pe_tr = 0.0250
Pe_te = 0.0350
sup_vec = 25
marg = 0.3573
```



6η Άσκηση

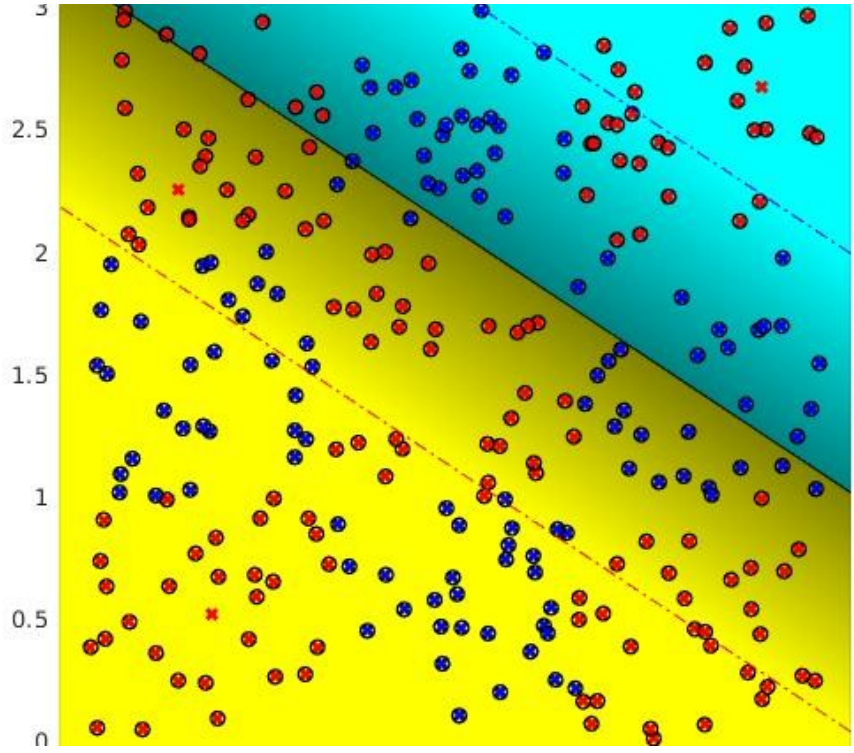
(1)

Για την παραγωγή των συνόλων καθώς και της κατηγοριοποίησης αυτών έγινε χρήση του δοθέντος κώδικα.

(2)

α)

```
kernel = 'linear'
marg = 1.3261
Pe_train = 0.4222
Pe_test = 0.4444
sup_vec = 269
```

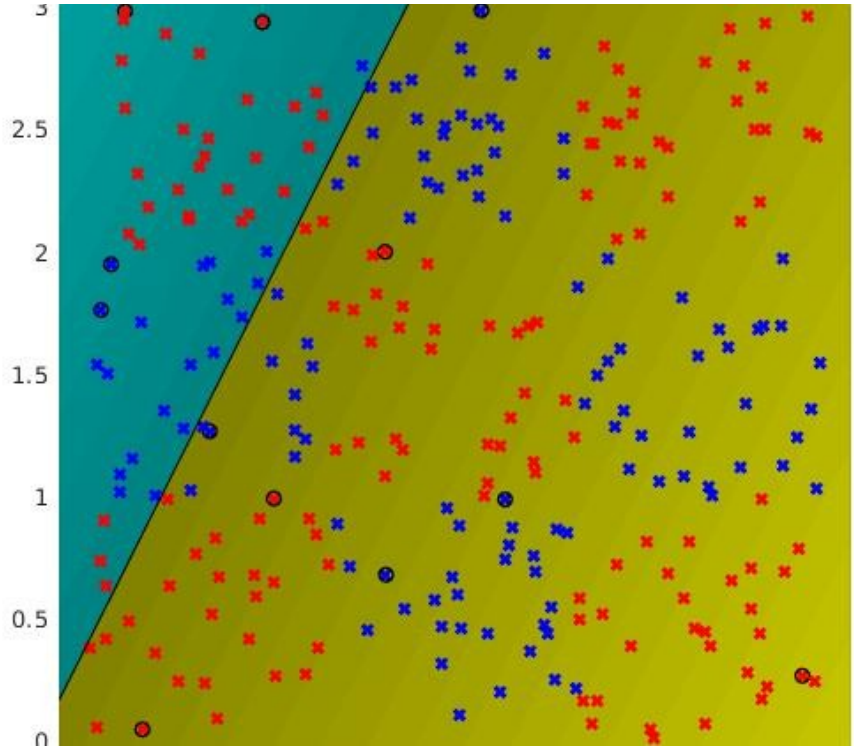


kernel = 'poly'

Pe_train = 0.4852

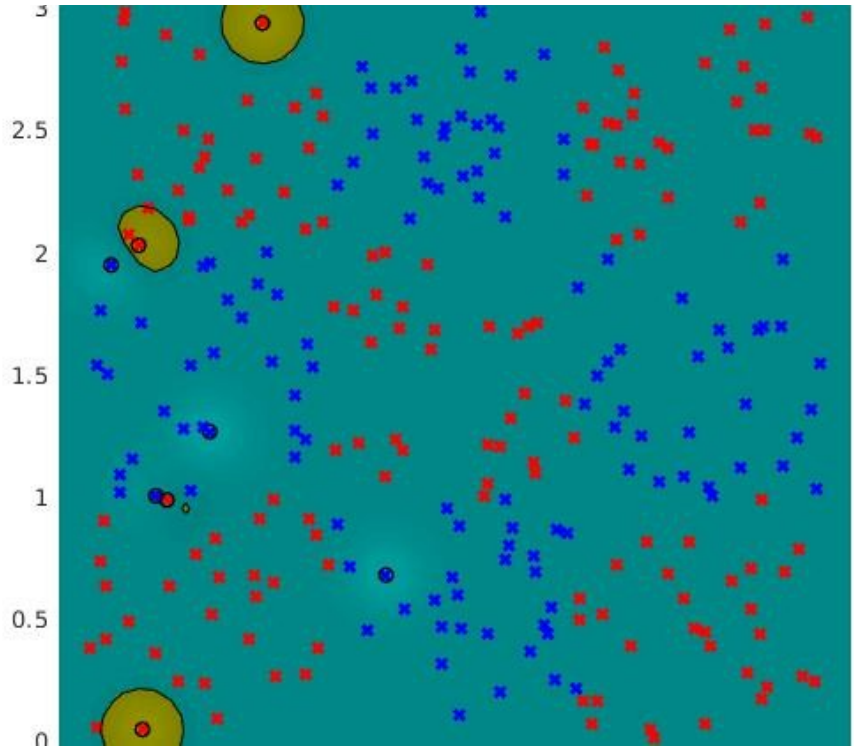
Pe_test = 0.5111

sup_vec = 12

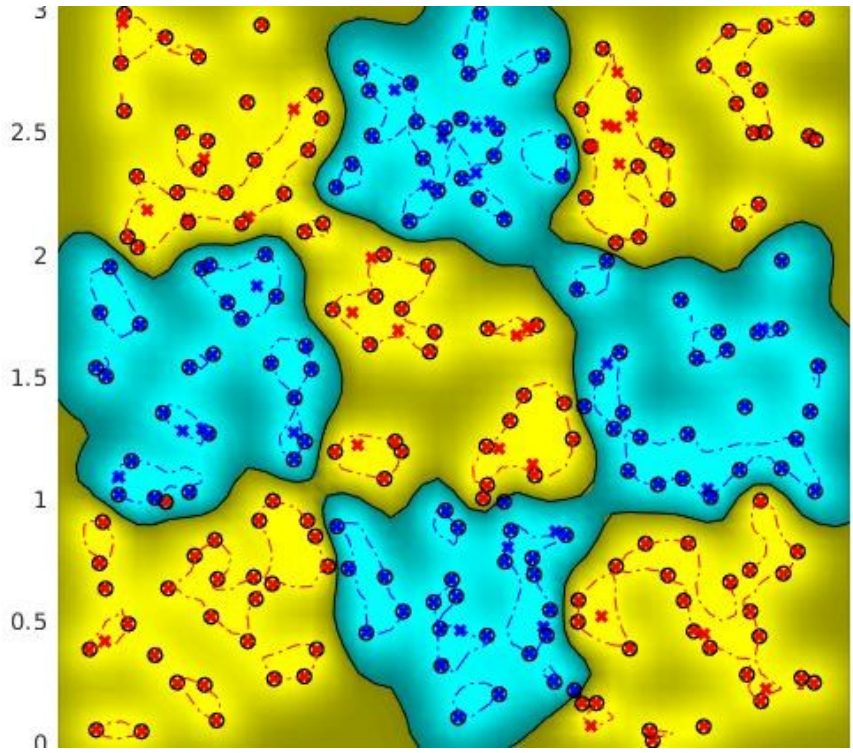


β)

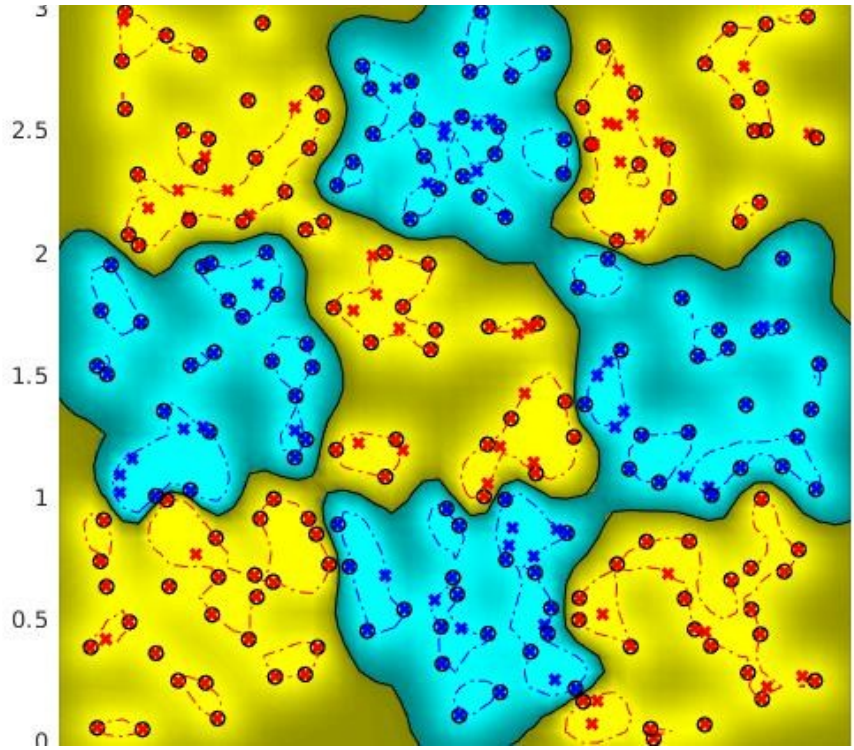
```
########################
          C = 0.2
########################
kernel = 'rbf'
Pe_train = 0.5370
Pe_test = 0.5370
sup_vec = 8
```



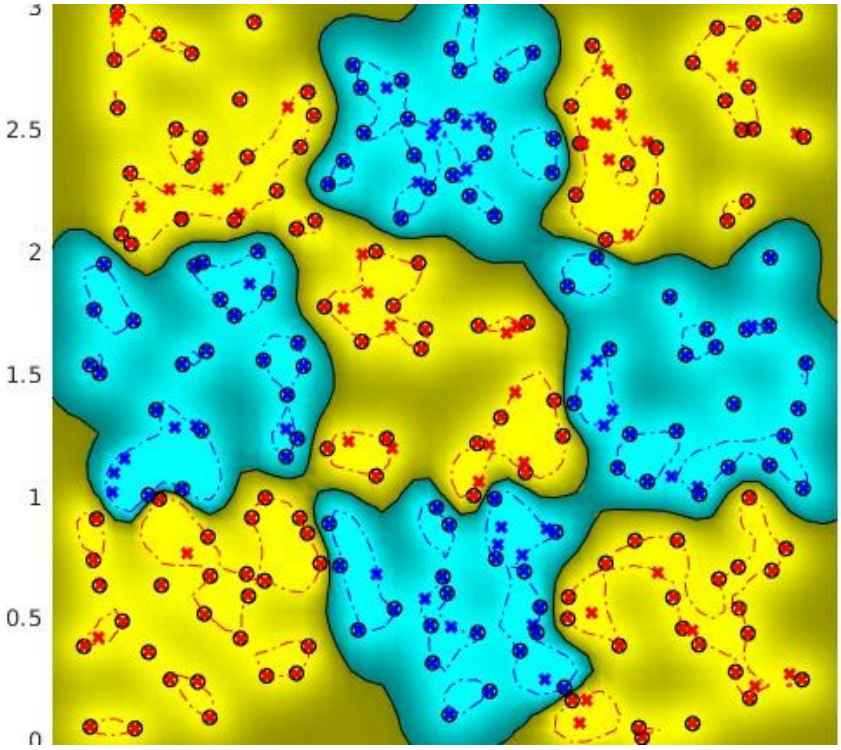
```
########################
          C = 2
########################
kernel = 'rbf'
Pe_train = 0.0037
Pe_test = 0.1074
sup_vec = 228
```



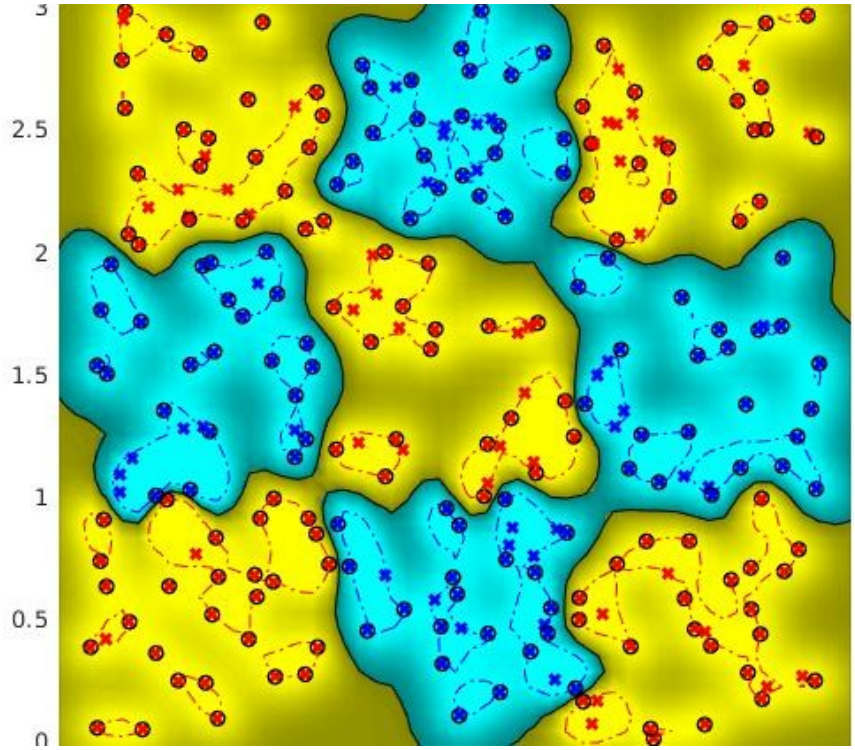
```
######################
          C = 20
######################
kernel = 'rbf'
Pe_train = 0
Pe_test = 0.1000
sup_vec = 215
```



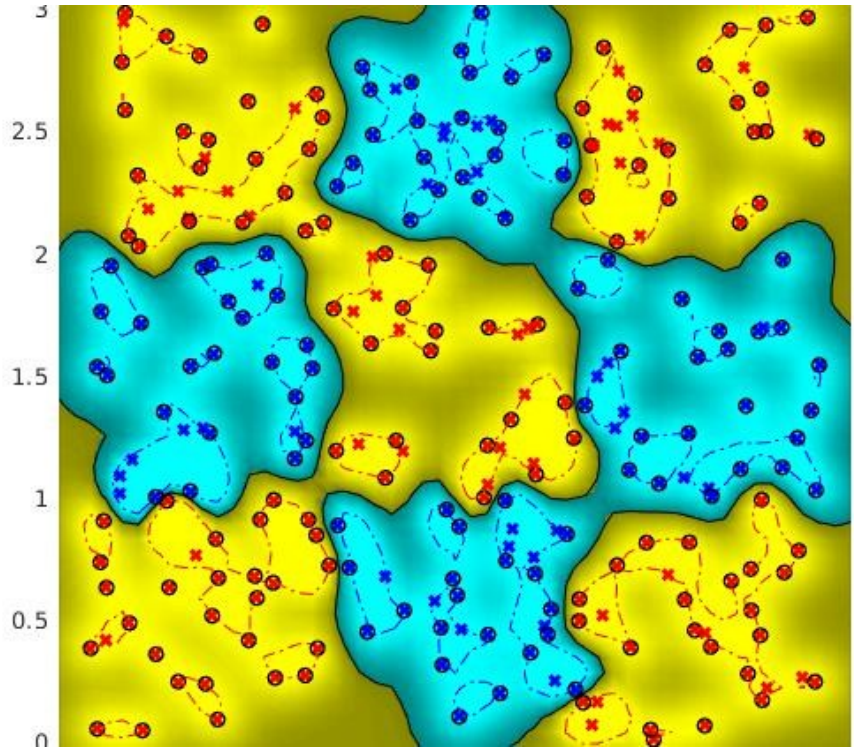
```
#######################
          C = 200
#######################
kernel = 'rbf'
Pe_train = 0
Pe_test = 0.1000
sup_vec = 215
```



```
######################
        C = 2000
######################
kernel = 'rbf'
Pe_train = 0
Pe_test = 0.1000
sup_vec = 215
```



```
######################
        C = 20000
######################
kernel = 'rbf'
Pe_train = 0
Pe_test = 0.1000
sup_vec = 215
```



(3)

Παρατηρούμε ότι οι ο γραμμικός SVM και ο πολυωνυμικός δεν κατηγοριοποιούν το σύνολο τόσο καλά όσο ο μη γραμμικός SVM με συναρτήσεις πυρήνα ακτινωτής βάσης. Επίσης παρατηρείται ότι για τον μη γραμμικό SVM όσο αυξάνεται το C τόσο αυξάνεται και η ακρίβεια κατηγοριοποίησης αλλά και ότι μετά από μια συγκεκριμένη τιμή του C η κατηγοριοποίηση παραμένει σταθερά η ίδια (στη συγκεκριμένη περίπτωση για C>20).

Ο κώδικας της εργασίας βρίσκεται στον προσωπικό μου λογαριασμό στο Github.