



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Αλέξανδρος Αριστόβουλος

1063199

Εργαστηριακή άσκηση 3

Ερώτημα 1:

Διαβάζουμε τα παραδείγματα με τη συνάρτηση ReadLiver ()

```
%Ερώτημα 1
%Ερώτημα 1
Tot=345;
[x,c] = ReadLiver(Tot);
```

Ερώτημα 2:

Τροποποιούμε τη συνάρτηση Perceptron(x1,x2,Lr,MaxRep) έτσι ώστε να μας επιστρέφει και τα βάρη που υπολογίζει. Έτσι ο ορισμός της γίνεται:

```
function [Rc,Rep,Weights] = Perceptron(x1,x2,Lr,MaxRep)
```

Παίρνουμε τους συντελεστές ως εξής:

```
%Ερώτημα 2
%Ερώτημα 2
% μετρητής για υγιείς
y_counter=1;
%μετρητής για αρρώστους
a_counter=1;

for i=1:Tot
    %υγιείς
    if c(i)==1
        yg(:,y_counter)=x(:,i);
        y_counter=y_counter+1;

        %άρρωστοι
    else
        ar(:,a_counter)=x(:,i);
        a_counter=a_counter+1;
    end
end

[Rc1,Rep1,Weights1] = Perceptron(yg,ar,1,10000);
disp("Διάγραμμα των συντελεστών της γραμμικής συνάρτησης απόφασης")
disp(Weights1)
```

Ακολουθεί το αποτέλεσμα του παραπάνω κώδικα.

```
Διάνυσμα των συντελεστών της γραμμικής συνάρτησης απόφασης
242.9770
-14.4860
392.1816
-893.1177
-127.7447
713.5877
-26.8213
```

Ερώτημα 3:

Χρησιμοποιούμε τη συνάρτηση Perceptron(x1,x2,Lr,MaxRep) για να βρούμε τις σωστές ταξινομήσεις και υπολογίζουμε το σφάλμα.

```
%Ερώτημα 3
error = 1-((Rc1(1)+ Rc1(2))/(Rep1(1)+Rep1(2)));
disp("Το σφάλμα είναι:")
disp(error)
```

```
Το σφάλμα είναι:
0.5333
```

Ερώτημα 4:

Καλούμε τον αλγόριθμο για 10.000 επαναλήψεις με συντελεστή εκπαίδευσης α μεταξύ του 0.5 και του 20 με βήμα 0.5. Για κάθε διαφορετικό συντελεστή καλούμε τη συνάρτηση 4 φορές και υπολογίζουμε το μέσο σφάλμα για να μειώσουμε την τυχαιότητα του αλγορίθμου.

```
counter = 1;
number_of_repeats = 4;
all_reliabilities = zeros(1,40);
coefficient = zeros(1,40);
errors = zeros(number_of_repeats);

for a=0.5:0.5:20
    average_error = 0;

    for i=1:number_of_repeats
        [Rc,Rep,~] = Perceptron(yg,ar,a,10000);
        average_error = average_error + 1-
        ((Rc(1)+Rc(2))/(Rep(1)+Rep(2)));
    end

    average_error = average_error/number_of_repeats;
    all_reliabilities(1,counter)= 1 - average_error;
    coefficient(counter) = a;
    counter=counter+1;
end

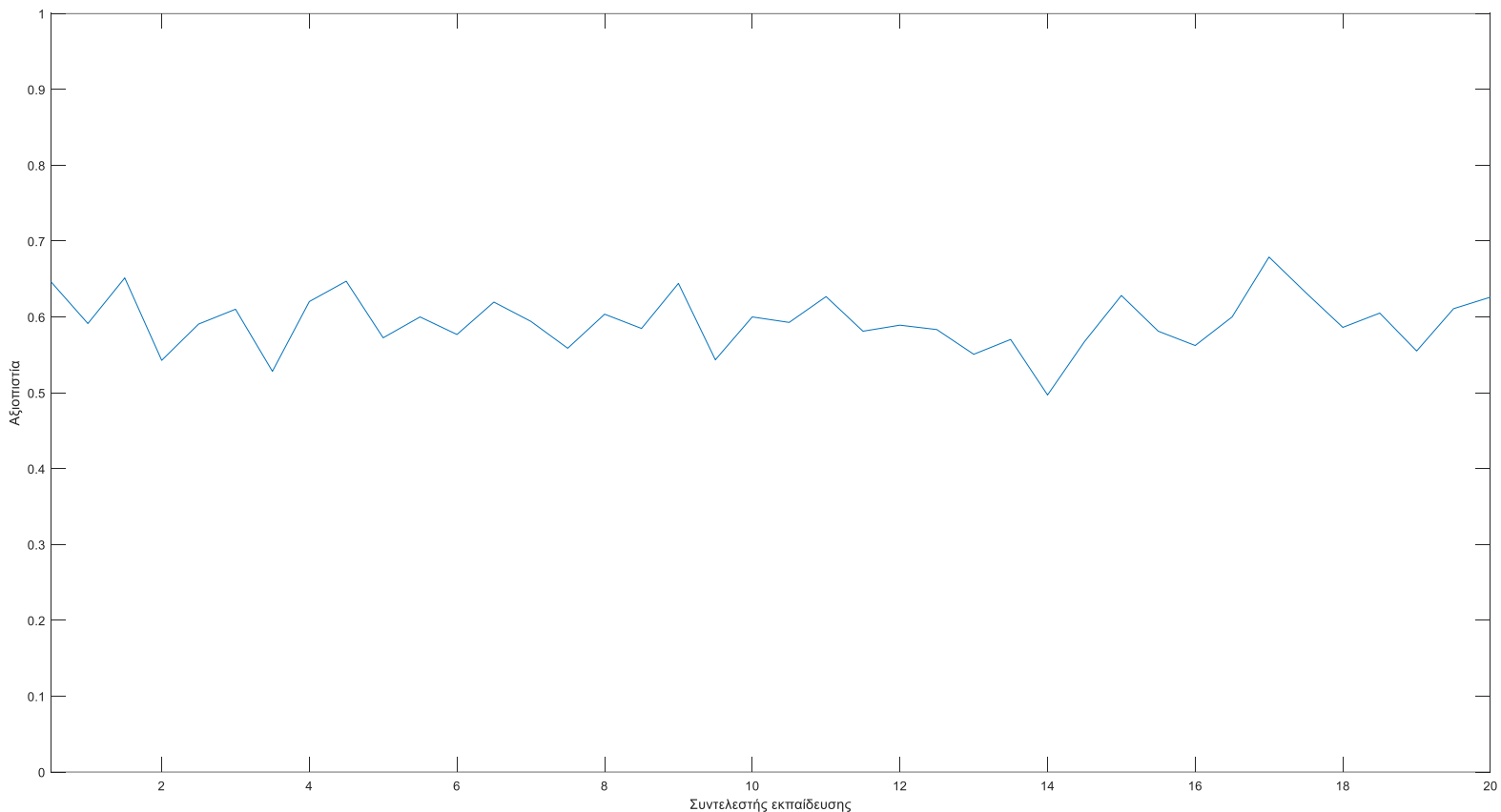
figure('Name','Αξιοπιστία του γραμμικού συστήματος ταξινόμησης
συναρτήσεων του συντελεστή εκπαίδευσης','NumberTitle','off')
```

```

plot(coefficient,all_reliabilities)
xlabel('Συντελεστής εκπαίδευσης')
ylabel('Αξιοπιστία')
xlim([coefficient(1) coefficient(counter-1)])
ylim([0 1])

```

Αξιοπιστία του γραμμικού συστήματος ταξινόμησης συναρτήσει του συντελεστή εκπαίδευσης



Για τον υπολογισμό των επαναλήψεων συναρτήσει του σφάλματος καλούμε τη συνάρτηση με τον ίδιο συντελεστή εκπαίδευσης και για επαναλήψεις από 10.000 έως 20.000 με βήμα 1.000.

```

repeats = zeros(1,11);
counter = 1;
average_errors = zeros(1,11);
for rep=10000:1000:20000
    average_error = 0;

    for i=1:number_of_repeats
        [Rc,Rep,~] = Perceptron(yg,ar,1,rep);
        average_error = average_error + 1-
        ((Rc(1)+Rc(2))/(Rep(1)+Rep(2)));
    end

    average_error = average_error/number_of_repeats;
    average_errors(counter) = average_error;

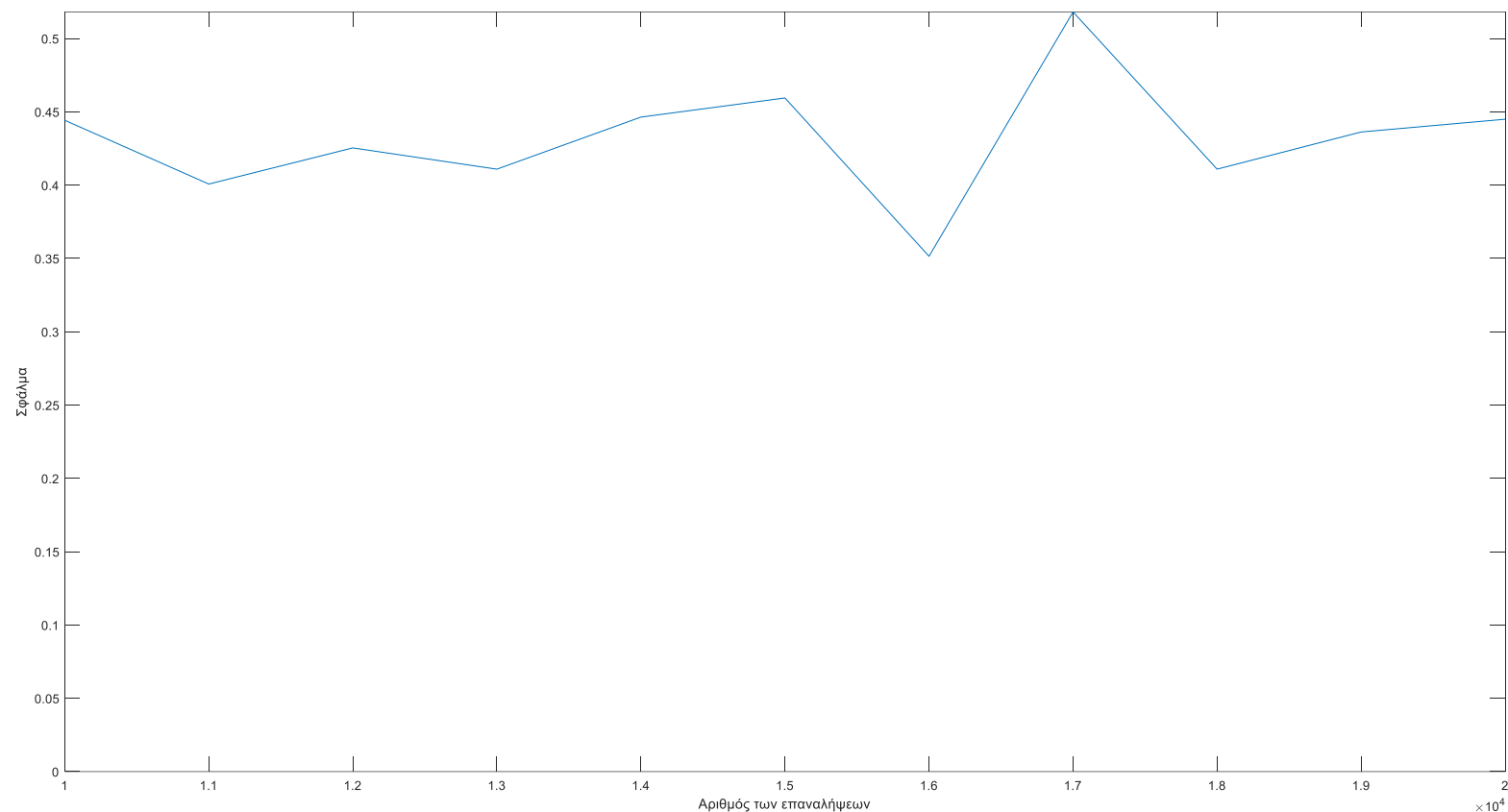
```

```

        repeats(counter) = rep;
        counter=counter+1;
    end

    figure('Name','Αριθμός των επαναλήψεων του αλγόριθμου συναρτήσει του
    σφάλματος','NumberTitle','off')
    plot(repeats,average_errors)
    xlabel('Αριθμός των επαναλήψεων')
    ylabel('Σφάλμα')
    xlim([repeats(1) repeats(counter-1)])
    ylim([0 inf])

```



Ερώτημα 5:

- Παρατηρούμε ότι μεταβάλλοντας τον συντελεστή εκπαίδευσης δεν υπάρχει κάποια ουσιαστική διαφορά στην αξιοπιστία του αλγόριθμου. Αυτό συμβαίνει διότι ο αλγόριθμος δεν είναι αρκετά καλός και βασίζεται πολύ στην τύχη. Επαναλαμβάνοντας την ίδια διαδικασία για να μειώσουμε την επίδραση της τύχης βλέπουμε ότι δεν μπορούμε να καταλήξουμε σε κάποιο «επιθυμητό» συντελεστή εκπαίδευσης.
- Για τον ίδιο λόγο βλέπουμε ότι αυξάνοντας τις επαναλήψεις του αλγόριθμου δεν υπάρχει κάποια ουσιαστική διαφορά.