

Εισαγωγή

Στην Εισαγωγή της εργασίας θα πραγματοποιηθεί σύντομη περιγραφή του πεδίου των Νευρωνικών Δικτύων, καθώς και περιγραφή εννοιών που χρησιμοποιούνται στα ερωτήματα της εργασίας. Αναλυτικότερα, τα νευρωνικά δίκτυα αποτελούν υποσύνολο της Μηχανικής Μάθησης και λαμβάνουν αποφάσεις μιμούμενα τους βιολογικούς νευρώνες. Κάθε νευρωνικό δίκτυο αποτελείται από απλές μονάδες επεξεργασίας και έχει τη δυνατότητα να αποθηκεύει εμπειρική γνώση. Περιέχει ένα στρώμα εισόδου, ένα στρώμα εξόδου και τουλάχιστον ένα κρυφό στρώμα. Η έξοδος παράγεται έπειτα από την επεξεργασία της εισόδου με τη βοήθεια μιας συνάρτησης ενεργοποίησης. Αν το οπτικοποιήσουμε, κάθε κόμβος συνδέεται με άλλους και έχει το δικό του βάρος και κατώφλι. Βασίζονται σε δεδομένα εκπαίδευσης για να αποκτήσουν γνώση και βελτιώσουν την ακρίβεια τους με την πάροδο του χρόνου. Συχνά χρησιμοποιούνται σε εφαρμογές ταξινόμησης, παλινδρόμησης, πρόβλεψης και ανάλυσης δεδομένων. Τέλος, τα πολυστρωματικά νευρωνικά δίκτυα αποτελούν το κεντρικό σημείο της Βαθιάς Μάθησης. Στη συγκεκριμένη εργασία ασχοληθήκαμε με τα νευρωνικά δίκτυα Perceptron και Adaline.

Επιπροσθέτως, στη συγκεκριμένη εργασία ασχοληθήκαμε με την οπτικοποίηση δεδομένων, δηλαδή τη διαδικασία απεικόνισης πληροφοριών με τη χρήση γραφημάτων. Με αυτό τον τρόπο υπάρχει η δυνατότητα ερμηνείας πολύπλοκων και μεγάλης κλίμακας δεδομένων. Η οπτικοποίηση των δεδομένων είναι υποκατηγορία της Επιστήμης Ανάλυση Δεδομένων. Σε αυτό το πεδίο πραγματοποιείται αρκετά συχνά η χρήση της μέσης και της ενδιάμεσης τιμής κάποιων μεταβλητών, όπως και η χρήση της διασποράς και της τυπικής απόκλισης(μεγεθών που συναντάμε παρακάτω).

Αξίζει να σημειωθεί ότι η υλοποίηση της άσκησης πραγματοποιήθηκε μέσω της εφαρμογής Matlab. Τα .m αρχεία που θα παραθέσουμε παρακάτω, περιέχουν αναλυτικά σχόλια.

Μέρος Α

A1-A2

Στο πρώτο ερώτημα της εργασίας, παράγουμε τυχαία δεδομένα από μία διακριτή και από μία συνεχή κατανομή πιθανότητας. Έπειτα, υπολογίζουμε τα μεγέθη μέση τιμή, ενδιάμεση τιμή, διασπορά και τυπική απόκλιση. Όπως τονίζεται και στην εκφώνηση, ακολουθεί ο κώδικας του ερωτήματος, τα αποτελέσματα των υπολογισμών που πραγματοποιήθηκαν και σχετικό γράφημα μέσα από το περιβάλλον του Matlab με μορφή screenshot.

Κώδικας:

Αρχείο ala2.m

```
% Παραγωγή τυχαίων δεδομένων

% Παράμετροι για τη διακριτή κατανομή (διωνυμική)

n = 9;
p = 0.8;
N = 1000;

% Δημιουργία τυχαίων δεδομένων από διωνυμική κατανομή με χρήση
της randi
data_discrete = randi([0, n], N, 1);
data_discrete = sum(data_discrete <= n * p, 2);

% Παράμετροι για τη συνεχή κατανομή (κανονική)
m = 0;
s = 1;
N_continuous = 1000;

% Δημιουργία τυχαίων δεδομένων από κανονική κατανομή με χρήση
της randn
data_continuous = m + s * randn(N_continuous, 1);

% Ιστόγραμμα για τη διακριτή κατανομή
[height_discrete, centers_discrete] = hist(data_discrete,
unique(data_discrete));
```

```

% Ιστογράμμα για τη συνεχή κατανομή

% Χρήση 20 bins

[height_continuous, centers_continuous] =
hist(data_continuous, 20);

figure;

% Διακριτή κατανομή
subplot(2, 1, 1);
bar(centers_discrete, height_discrete / sum(height_discrete));
xlabel('Value');
ylabel('Probability');
title('Discrete Distribution using randi');

% Συνεχής κατανομή
subplot(2, 1, 2);
bar(centers_continuous, height_continuous /
sum(height_continuous));
xlabel('Value');
ylabel('Probability');
title('Continuous Distribution using randn');

% Υπολογισμοί μέσης τιμής, ενδιάμεσης τιμής, διασποράς και
τυπικής απόκλισης
disp('ΔΙΑΚΡΙΤΗ ΚΑΤΑΝΟΜΗ:');
disp('Εύρεση μέσης τιμής:');
disp(median(data_discrete));

disp('Εύρεση ενδιάμεσης τιμής:');
disp(mean(data_discrete));

variance = var(data_discrete);
disp('Υπολογισμός διασποράς:');
disp(variance);

```

```

ta = std(data_discrete);
disp('Υπολογισμός τυπικής απόκλισης:');
disp(ta);

disp('ΣΥΝΕΧΗΣ ΚΑΤΑΝΟΜΗ:');
disp('Εύρεση μέσης τιμής:');
disp(median(data_continuous));

disp('Εύρεση ενδιάμεσης τιμής:');
disp(mean(data_continuous));

variance2 = var(data_continuous);
disp('Υπολογισμός διασποράς:');
disp(variance);

ta2 = std(data_continuous);
disp('Υπολογισμός τυπικής απόκλισης:');
disp(ta);

```

Αποτελέσματα:

ΔΙΑΚΡΙΤΗ ΚΑΤΑΝΟΜΗ:

Εύρεση μέσης τιμής:

1

Εύρεση ενδιάμεσης τιμής:

0.8290

Υπολογισμός διασποράς:

0.1419

Υπολογισμός τυπικής απόκλισης:

0.3767

ΣΥΝΕΧΗΣ ΚΑΤΑΝΟΜΗ:

Εύρεση μέσης τιμής:

0.0497

Εύρεση ενδιάμεσης τιμής:

0.0321

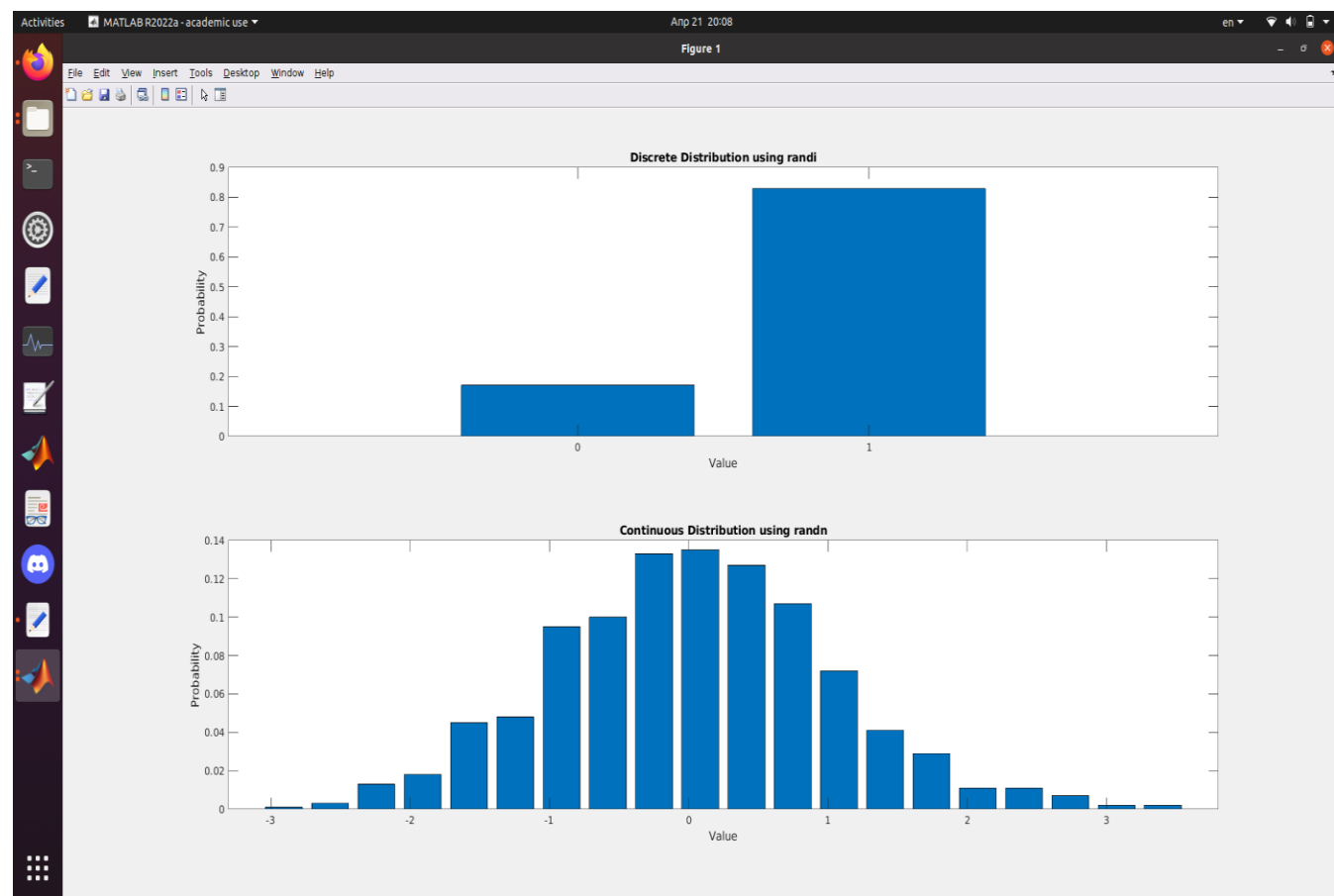
Υπολογισμός διασποράς:

0.1419

Υπολογισμός τυπικής απόκλισης:

0.3767

Γράφημα:



A3

Στο ερώτημα A3, όπως και στο επόμενο(A4) πραγματοποιείται οπτικοποίηση δεδομένων από το dataset cereal, που βρίσκεται στη βιβλιοθήκη του Matlab. Ενώ, για το επόμενο ερώτημα χρησιμοποιήσαμε το dataset Air_Quality.csv, που συλλέχθηκε από το διαδίκτυο. Επίσης, σε μεταβλητές αποθηκεύσαμε τους υπολογισμούς μέσης και ενδιάμεσης τιμής, διασποράς και τυπικής απόκλισης για ένα συγκεκριμένο χαρακτηριστικό του αντίστοιχου dataset.

Κώδικας:

Αρχείο cereal.m

```
load cereal.mat

% Το συγκεκριμένο dataset βρίσκεται στη βιβλιοθήκη του Matlab
% και περιέχει διάφορες πληροφορίες σχετικά με τα δημητριακά.

%Περιεχόμενο του dataset:

    %Name: Ονόματα δημητριακών προϊόντων.
    %Type: Τύπος δημητριακών
    %Calories: Περιεκτικότητα θερμίδων ανά μερίδα.
    %Protein: Περιεκτικότητα πρωτεΐνης ανά μερίδα.
    %Fat: Περιεκτικότητα λίπους ανά μερίδα.
    %Sodium: Περιεκτικότητα νατρίου ανά μερίδα.
    %Fiber: Περιεκτικότητα ινών ανά μερίδα.
    %Carbo: Περιεκτικότητα υδατανθράκων ανά μερίδα.
    %Sugars: Περιεκτικότητα ζάχαρης ανά μερίδα.
    %Potass: Περιεκτικότητα καλίου ανά μερίδα.
    %Vitamins: Περιεκτικότητα βιταμινών ανά μερίδα.
    %Shelf: Ράφι στο κατάστημα όπου τοποθετούνται τα
δημητριακά.
    %Weight: Βάρος ανά μερίδα.
    %Cups: Πλήθος κυπέλλων ανά μερίδα.

% Επιλογή δεδομένων από το dataset
```

```

calories = Calories;
cerealNames = Name;

% Μετατροπή των ονομάτων δημητριακών σε κατηγορίες
(categories)

cerealCategories = categorical(cerealNames);

% Οπτικοποίηση των δεδομένων

figure;
bar(cerealCategories, calories);
xlabel('Cereal Products');
ylabel('Calories (Περιεκτικότητα θερμίδων ανά μερίδα)');
title('Calories σε διάφορα Cereal Products');
xtickangle(45); % Περιστροφή των ονομάτων για ευκολότερη
ανάγνωση
grid on;

% -----

proteinContent = Protein;

figure;
histogram(proteinContent, 'BinWidth', 1);
xlabel('Protein (Περιεκτικότητα πρωτεΐνης ανά μερίδα)');
ylabel('Frequency');
title('Κατανομή Περιεχομένου Πρωτεϊνών σε Προϊόντα
Δημητριακών');
grid on;

% -----

figure;

% Επιλογή των πρώτων 5 στοιχείων από 3 στήλες του πίνακα
selectedNames = Name(1:5);
selectedSugars = Sugars(1:5);
selectedVitamins = Vitamins(1:5);

```

```

% 2 διαγράμματα σε κοινό παράθυρο
subplot(1, 2, 1);
bar(categorical(selectedNames), selectedSugars);
xlabel('Cereal Products');
ylabel('Sugar(Περιεκτικότητα ζάχαρης ανά μερίδα) σε g');
title('Περιεκτικότητα ζάχαρης σε διάφορα προϊόντα
δημητριακών');
xtickangle(45); % Περιστροφή των ονομάτων για ευκολότερη
ανάγνωση
grid on;

subplot(1, 2, 2);
bar(categorical(selectedNames), selectedVitamins);
xlabel('Cereal Products');
ylabel('Vitamin(Περιεκτικότητα βιταμινών ανά μερίδα)
ποσοστό%');
title('Περιεκτικότητα βιταμινών σε διάφορα προϊόντα
δημητριακών');
xtickangle(45); % Περιστροφή των ονομάτων για ευκολότερη
ανάγνωση
grid on;

% -----
pot = Potass;
% Δημιουργία διαγράμματος boxplot για τη μεταβλητή Potass
figure;
boxplot(pot, 'Labels', {'Potass'});
ylabel('Potass( Περιεκτικότητα καλίου ανά μερίδα)');
title('Box Plot');

% -----
%Περιεκτικότητα υδατανθράκων ανά μερίδα
carbohydrates = Carbo;

```



```
% Υπολογισμός μέσης τιμής
mean_carbo = mean(carbohydrates);

% Υπολογισμός ενδιάμεσης τιμής
median_carbo = median(carbohydrates);

% Υπολογισμός διασποράς
variance_carbo = var(carbohydrates);

% Υπολογισμός τυπικής απόκλισης
std_dev_carbo = std(carbohydrates);

% Εκτύπωση των αποτελεσμάτων
fprintf('Μέση τιμή των Carbo: %.2f\n', mean_carbo);
fprintf('Ενδιάμεση τιμή των Carbo: %.2f\n', median_carbo);
fprintf('Διακύμανση των Carbo: %.2f\n', variance_carbo);
fprintf('Τυπική απόκλιση των Carbo: %.2f\n', std_dev_carbo);
```

Αποτελέσματα:

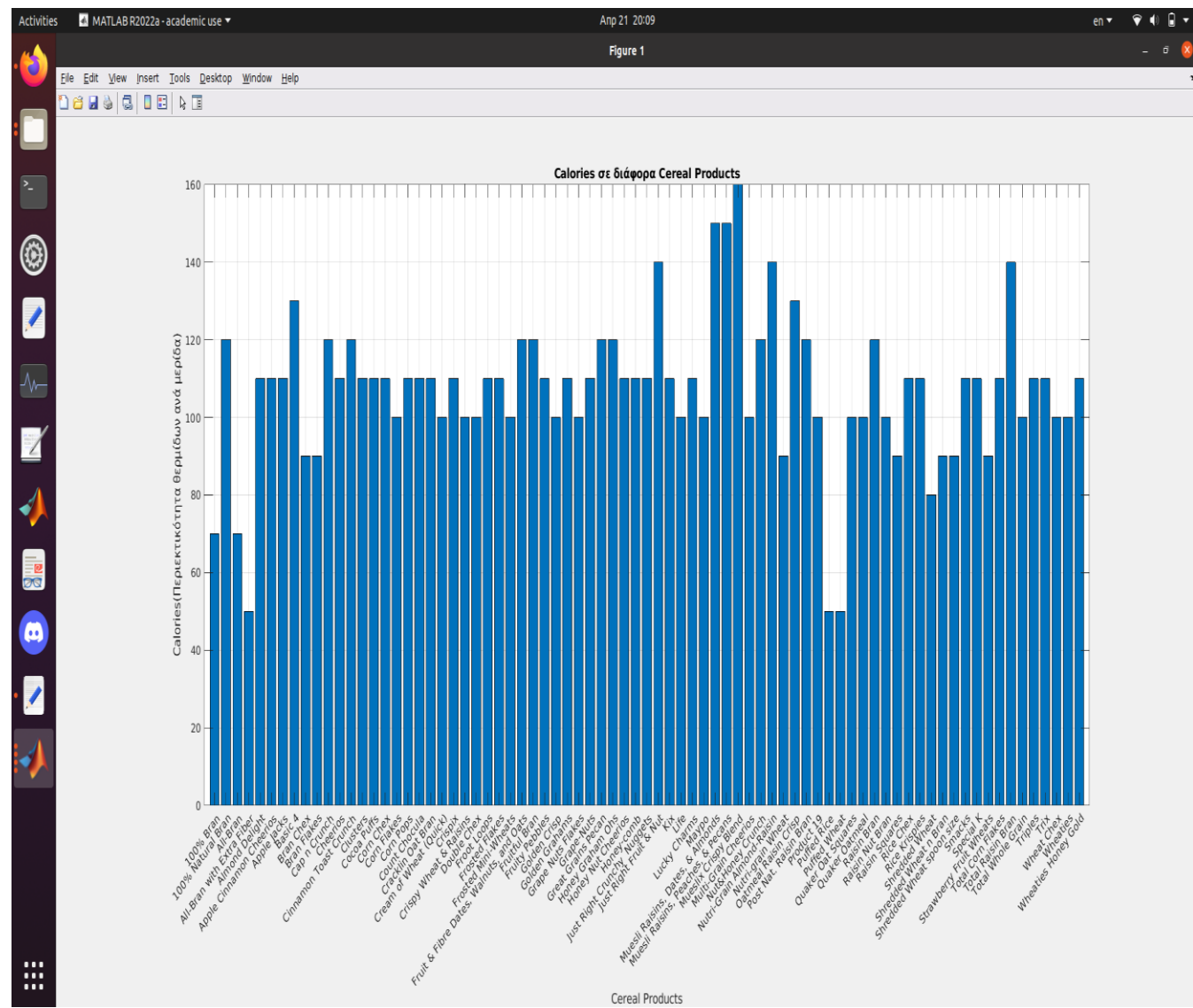
Μέση τιμή των Carbo: 14.60

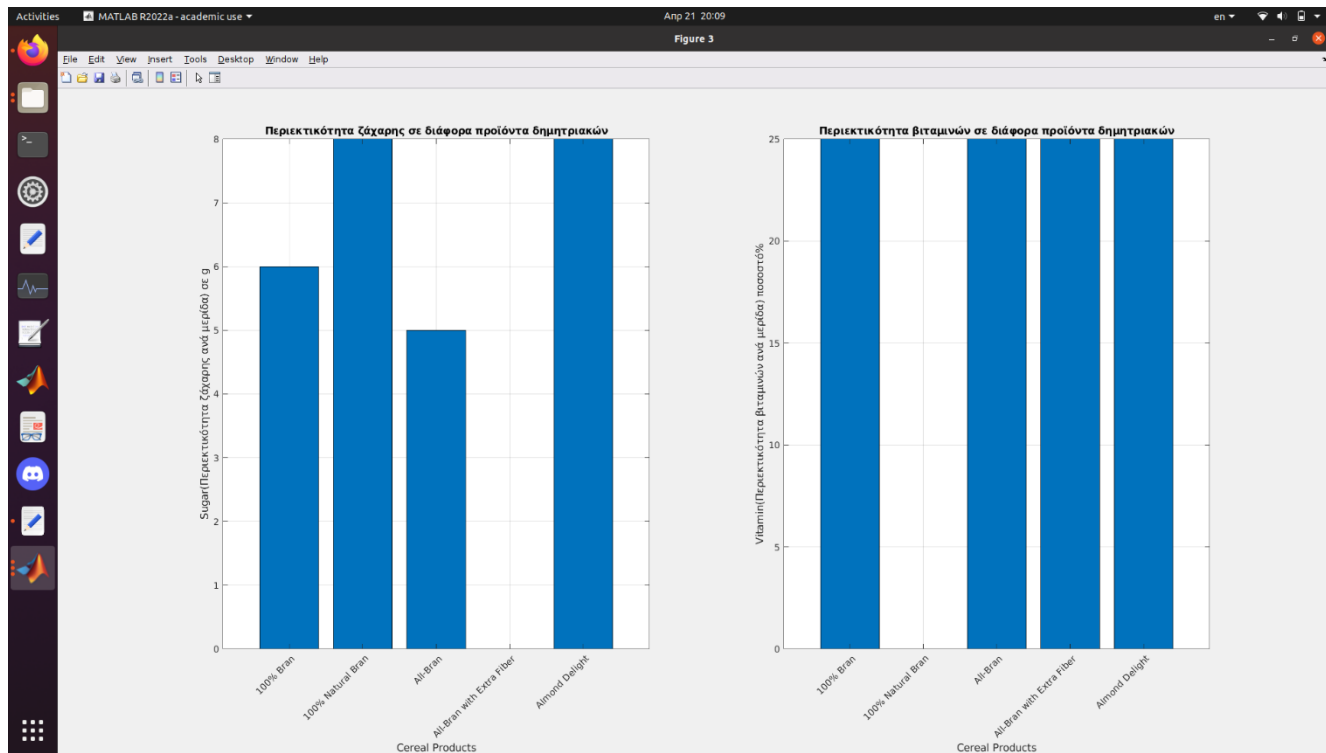
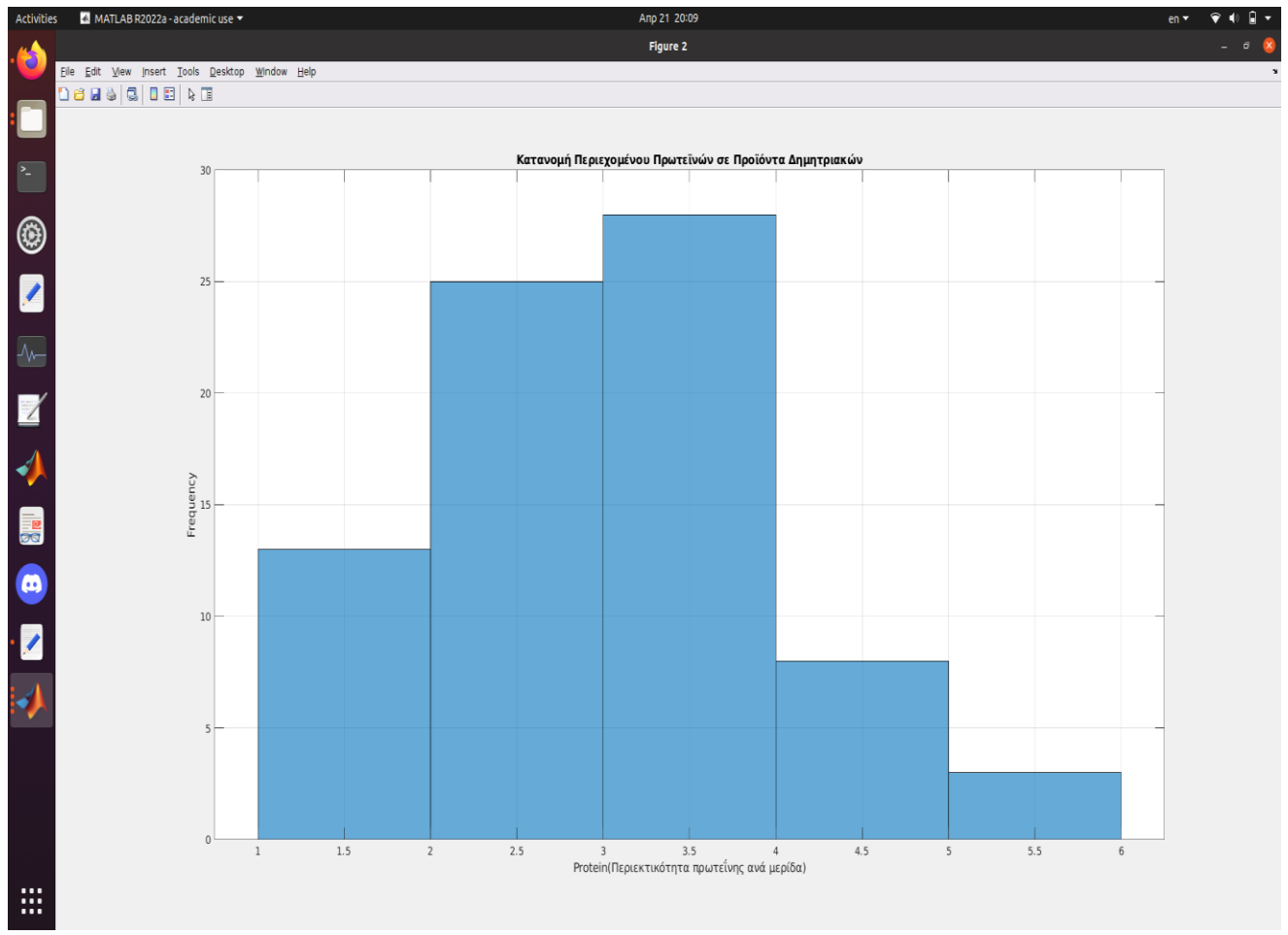
Ενδιάμεση τιμή των Carbo: 14.00

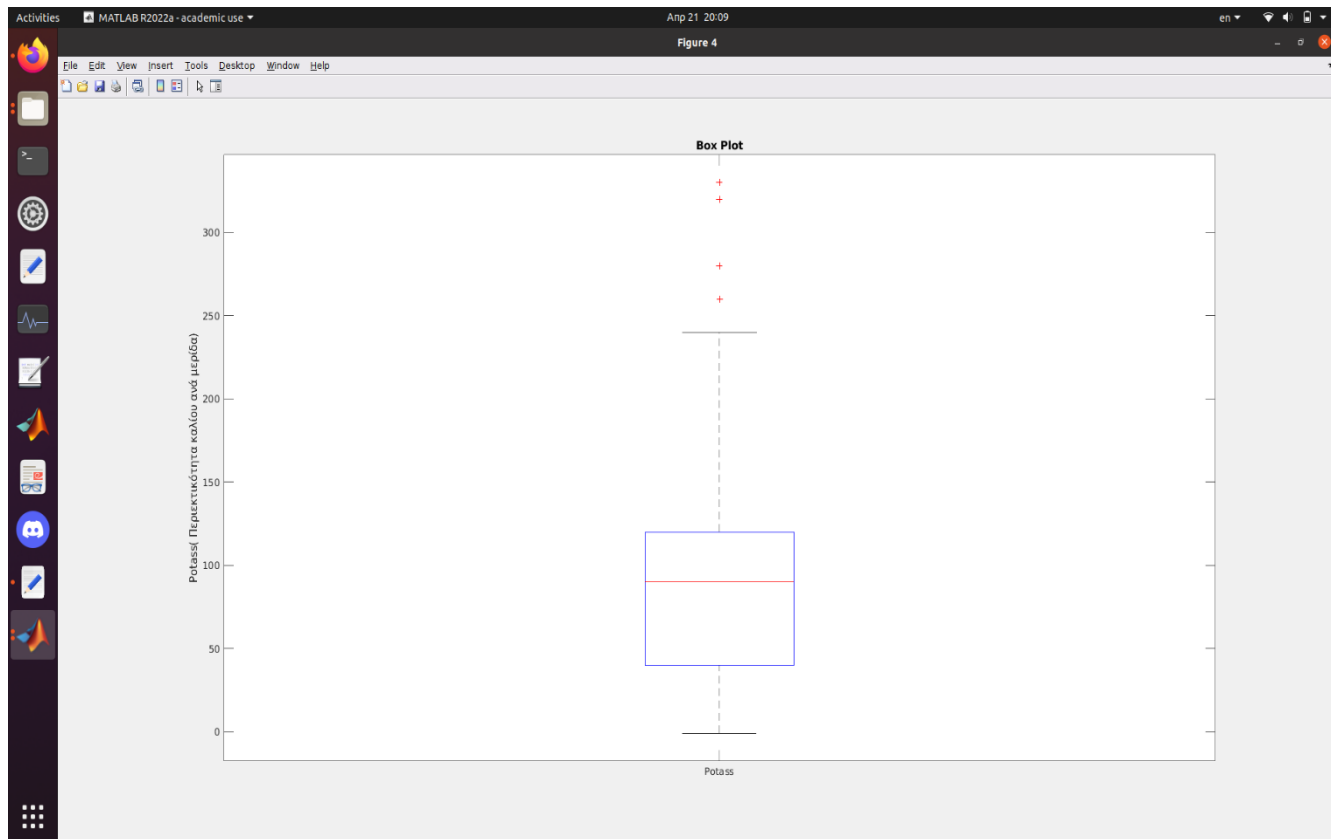
Διακύμανση των Carbo: 18.31

Τυπική απόκλιση των Carbo: 4.28

Γραφήματα:







A4

Κώδικας:

Αρχείο airquality.m

```
filename = 'Air_Quality.csv';
```

```
data = readtable(filename);
```

```
head(data)
```

% Το συγκεκριμένο σύνολο δεδομένων περιέχει πληροφορίες για τα δεδομένα επιτήρησης

% της ποιότητας του αέρα της Νέας Υόρκης.

% Οι παρακάτω δείκτες παρέχουν μια προοπτική για το χρόνο και τη γεωγραφία

```

% της Νέας Υόρκης για τον καλύτερο χαρακτηρισμό της ποιότητας
και

% της υγείας του αέρα στη Νέα Υόρκη.


% Οπτικοποίηση των δεδομένων του dataset
% Προεπεξεργασία των δεδομένων
dataSubset = data(1:10, :);
categoricalNames = categorical(dataSubset.Name);
categoryCounts = countcats(categoricalNames);


figure;
bar(categoryCounts);
xlabel('Είδη ρύπων');
ylabel('Συχνότητα εμφάνισης');
title('Συχνότητα εμφάνισης του κάθε είδους ρύπου');
xticklabels(categories(categoricalNames)); % Ορισμός ετικετών
x με τα ονόματα κατηγοριών
xtickangle(45); % Περιστροφή των ονομάτων (διευκόλυνση στην
ανάγνωση)
grid on;


% -----

% Υπολογισμός συχνότητας εμφάνισης των γεωγραφικών τοποθεσιών
(GeoPlaceName)
placeCounts = countcats(categorical(data.GeoPlaceName));
uniquePlaces = categories(categorical(data.GeoPlaceName));


% Δημιουργία pie chart
figure;
pie(placeCounts, uniquePlaces);
title('Ποσοστό κάθε γεωγραφικής τοποθεσίας στο dataset');


% -----

```

```
% Υπολογισμός μέσης τιμής, ενδιάμεσης τιμής, διασποράς και
τυπικής απόκλισης για τη DataValue

mean_data = mean(data.DataValue);
median_data = median(data.DataValue);
variance_data = var(data.DataValue);
std_dev_data = std(data.DataValue);

% Εκτύπωση των αποτελεσμάτων
fprintf('Μέση τιμή της DataValue: %.2f\n', mean_data);
fprintf('Ενδιάμεση τιμή της DataValue: %.2f\n', median_data);
fprintf('Διακύμανση της DataValue: %.2f\n', variance_data);
fprintf('Τυπική απόκλιση της DataValue: %.2f\n',
std_dev_data);
```

Αποτελέσματα:

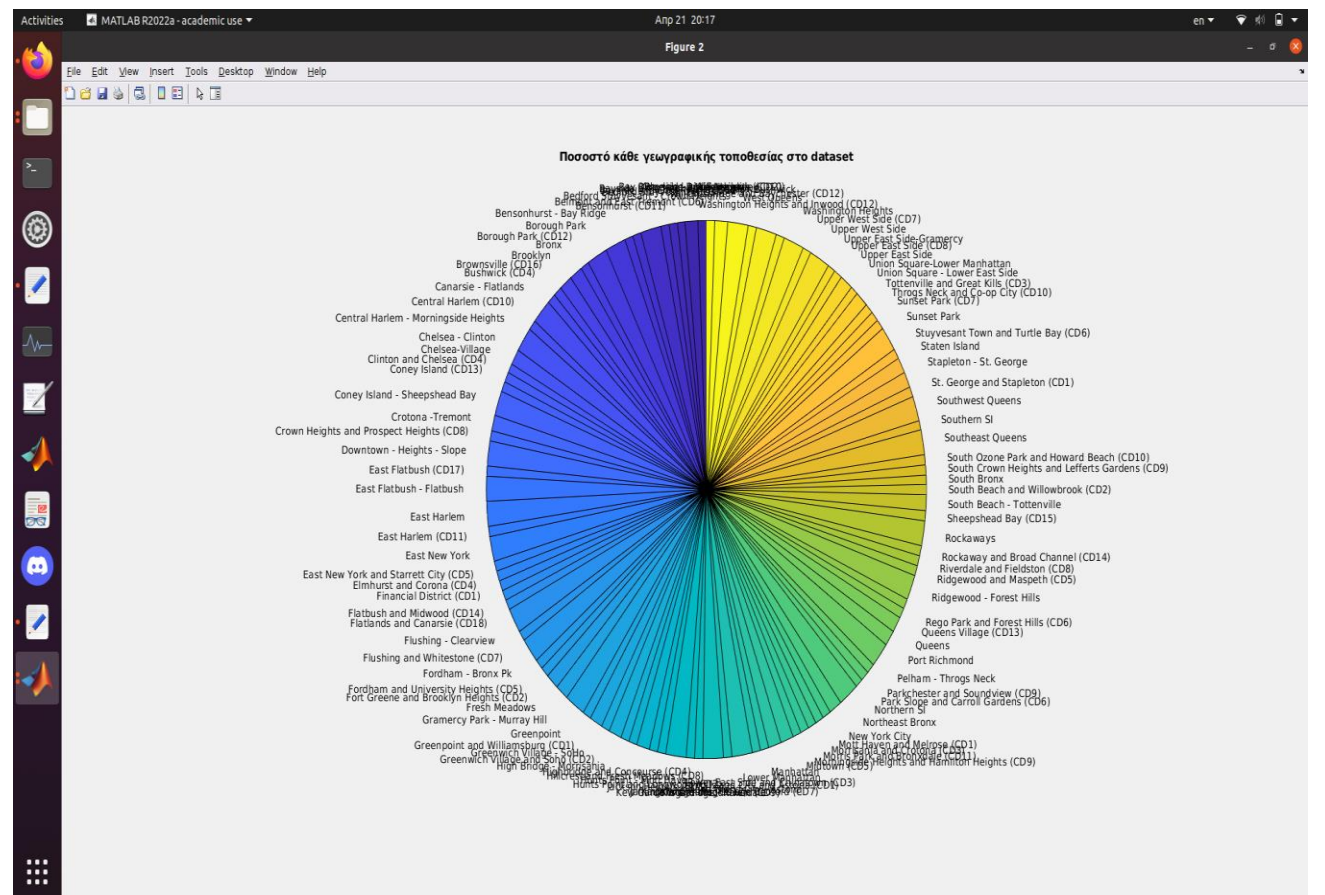
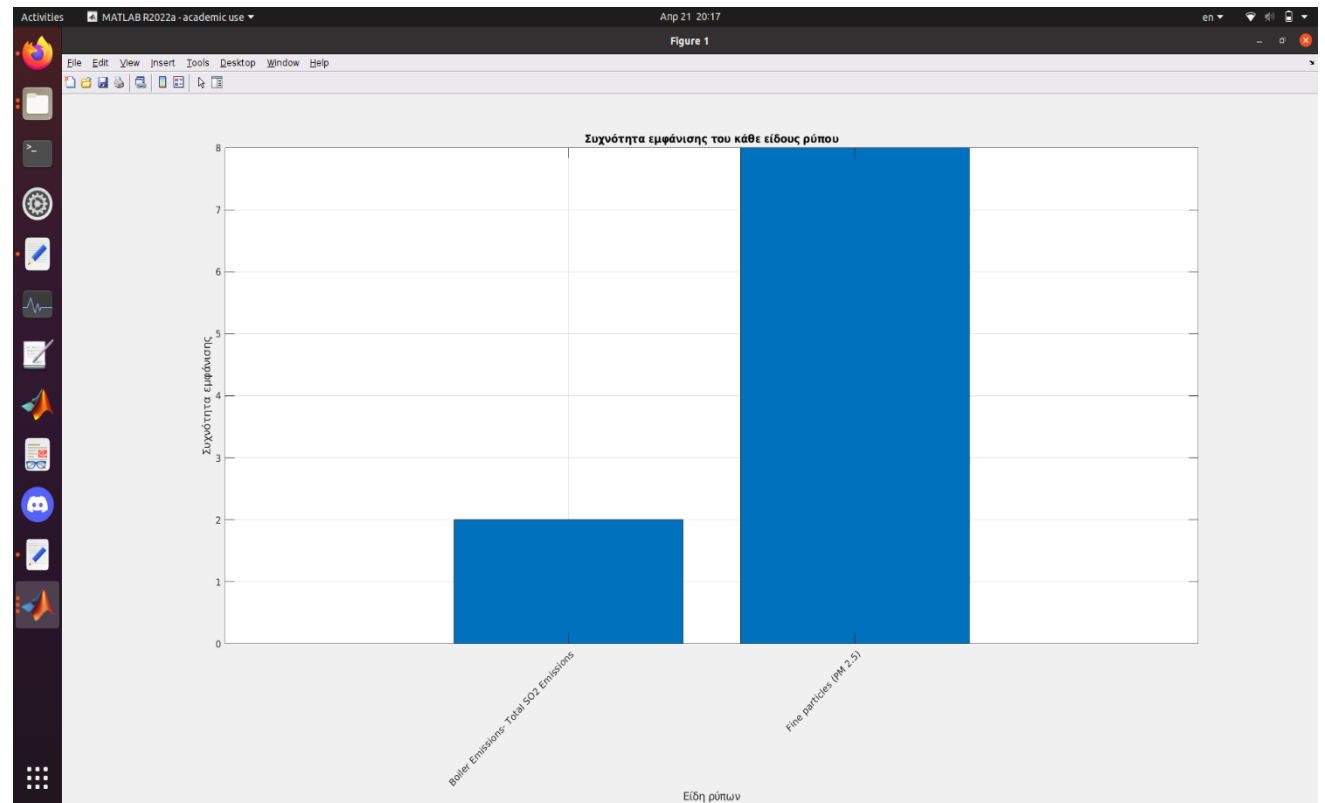
Μέση τιμή της DataValue: 21.43

Ενδιάμεση τιμή της DataValue: 15.20

Διακύμανση της DataValue: 575.97

Τυπική απόκλιση της DataValue: 24.00

Γραφήματα:



Μέρος Β

B1

Στο συγκεκριμένο ερώτημα λαμβάνει χώρα η δημιουργία ενός νευρωνικού δικτύου Perceptron. Στη συνέχεια, εκπαιδεύουμε το νευρωνικό δίκτυο, προσομοιώνουμε την έξοδο του και υπολογίζουμε το σφάλμα του. Όπως σημειώνεται και στην εκφώνηση της εργασίας, ακολουθήσαμε τα βήματα και τις οδηγίες που βρίσκονται στο φυλλάδιο ERGASTHRIO_NEURAL_NETS, και συγκεκριμένα την άσκηση 2.2 που βρίσκεται στη σελίδα 12.

Κώδικας:

Αρχείο Perceptron.m

```
% Διαχωρισμός Γραμμικά Διαχωρίσιμων Κλάσεων με Perceptron

% Το νευρωνικό δίκτυο Perceptron αποτελείται από ένα νευρώνα
τύπου McCulloch-Pitts

% με n εισόδους και μία έξοδο

% Χρησιμοποιεί τη Βηματική Συνάρτηση Ενεργοποίησης.

% Είναι ένα δίκτυο επιβλεπόμενης μάθησης,

% για αυτό το λόγο χρησιμοποιούμε και τους στόχους d, δηλαδή
τις επιθυμητές εξόδους.

% w :το διάνυσμα των συναπτικών βαρών

% x :το πρότυπο που εισάγεται κάθε φορά

% d :ο στόχος, η κλάση στην οποία ανήκει το πρότυπο με τιμές
0, 1.

% y :η έξοδος του νευρώνα με τιμές 0,1.

% Βήμα 1: Διάβασμα αριθμού προτύπων n (άρτιος αριθμός)
n = input('Εισάγετε τον αριθμό των προτύπων (άρτιος αριθμός):
');

% Βήμα 2: Διάβασμα συντελεστή εκπαίδευσης (Learning Rate) beta
beta = input('Εισάγετε τον συντελεστή εκπαίδευσης (Learning
Rate): ');

% Βήμα 3: Διάβασμα μέγιστου αριθμού επαναλήψεων
max_num_of_epochs
```



```

max_num_of_epochs = input('Εισάγετε το μέγιστο αριθμό
επαναλήψεων: ');

% Βήμα 4: Δημιουργία τυχαίων τιμών για τις συνάψεις w
w = randn(n + 1, 1);

% Βήμα 5: Δημιουργία τυχαίων τιμών για τα πρότυπα
p = zeros(n, 2);

% Δημιουργία προτύπων για την πρώτη κλάση λαμβάνοντας υπόψη
τις οδηγίες του φυλλαδίου
%  $(1 \leq x, y \leq 3)$ 
p(1:n/2, :) = rand(n/2, 2) * 2 + 1;

% Δημιουργία προτύπων για τη δεύτερη κλάση λαμβάνοντας υπόψη
τις οδηγίες του φυλλαδίου
%  $(7 \leq x, y \leq 9)$ 
p(n/2+1:n, :) = rand(n/2, 2) * 2 + 7;

% Βήμα 6: Δίνει τις τιμές 0, 1 για τους στόχους d
d = zeros(n, 1);
d(1:n/2) = 1; % Στόχοι (επιθυμητοί έξοδοι) για την 1η κλάση
d(n/2+1:n) = -1; % Στόχοι (επιθυμητοί έξοδοι) για τη 2η κλάση

% Βήμα 7: Εμφάνιση γραφήματος των προτύπων των 2 κλάσεων
figure;
plot(p(1:n/2, 1), p(1:n/2, 2), 'ro'); % Η πρώτη κλάση
εμφανίζεται με κόκκινο
hold on;
plot(p(n/2+1:n, 1), p(n/2+1:n, 2), 'go'); % Η δεύτερη κλάση
εμφανίζεται με πράσινο
xlabel('Χαρακτηριστικό 1');
ylabel('Χαρακτηριστικό 2');
title('Dataset');

```

```

% Βήμα 8: Αρχικοποίηση των εποχών σε 0
epochs = 0;

flag = 0; % Μεταβλητή για εντοπισμό αλλαγών στις συνάψεις

% Βήμα 9A:
while (flag == 0 && epochs < max_num_of_epochs)
    flag = 1;
    for i = 1:n
        % Υπολογισμός εξόδου
        y = w(1) * p(i, 1) + w(2) * p(i, 2) + w(3);

        % Έλεγχος σφάλματος και διόρθωση συνάψεων
        if sign(y) ~= sign(d(i))
            % Διόρθωση των συνάψεων με τον Κανόνα του Δέλτα
            w(1) = w(1) + beta * d(i) * p(i, 1);
            w(2) = w(2) + beta * d(i) * p(i, 2);
            w(3) = w(3) + beta * d(i);
            flag = 0;
        end
    end
end

% Ενημέρωση εποχών
epochs = epochs + 1;
end

% Βήμα 9B: Εμφανίζει το γράφημα των προτύπων των 2 κλάσεων,
ανάλογα με την κλάση στην
% οποία κατατάσσεται
% Χρήση συνάρτησης find: επιστρέφει στον πίνακα classa τους
δείκτες των στοιχείων
% του πίνακα y που είναι 0
Class1 = find(d == 1);

```

```

Class2 = find(d == -1);

figure;
plot(p(Class1, 1), p(Class1, 2), 'ro'); % 1η κλάση:κόκκινο
hold on;
plot(p(Class2, 1), p(Class2, 2), 'go'); % 2η κλάση:πράσινο
xlabel('Χαρακτηριστικό 1');
ylabel('Χαρακτηριστικό 2');
title('Γράφημα των προτύπων των 2 κλάσεων, ανάλογα με την
κλάση στην οποία κατατάσσεται');

% Οπτικοποίηση του γραμμικού διαχωριστικού ορίου των προτύπων
των δύο κλάσεων
x_line = linspace(min(p(:, 1)), max(p(:, 1)), 100);
y_line = -(w(1) * x_line + w(3)) / w(2);

hold on;
plot(x_line, y_line, 'k--'); % Σχεδίαση του ορίου με μαύρη
διακεκομμένη γραμμή
legend('Πρώτη κλάση', 'Δεύτερη κλάση', 'Γραμμικό διαχωριστικό
όριο των 2 κλάσεων');

% Βήματα 10,11: Πρόβλεψη εξόδου για τυχαία πρότυπα και
υπολογισμός ποσοστού σφάλματος
% - Προσομοίωση εξόδου -
test_samples = rand(2, 2); % Δύο τυχαία πρότυπα (ένα από κάθε
κλάση)
output = sign(w(1) * test_samples(:, 1) + w(2) *
test_samples(:, 2) + w(3));
disp('Προσομοίωση εξόδου:');
disp(output);

% - Υπολογισμός του ποσοστού σφάλματος -
error_percentage = sum(output' ~= [1; -1]) / numel(output) *
100;

```

```

disp(['Ποσοστό σφάλματος: ', num2str(error_percentage), '%']);

% - Εκπαίδευση του νευρωνικού δικτύου -

% Η εκπαίδευση του νευρωνικού δικτύου Perceptron έχει
υλοποιηθεί παραπάνω,

% με συγκεκριμένες εντολές. Στο σημείο αυτό παρουσιάζονται τα
βήματα που

% εκτελούνται επαναληπτικά ώστε να πραγματοποιηθεί η
εκπαίδευση.

% Βήματα:

% 1) Αρχικοποίηση παραμέτρων
% 2) Υπολογισμός εξόδου
% 3) Διόρθωση βαρών
% 4) Ενημέρωση αριθμού εποχών

% Αν χρησιμοποιούσαμε τη συνάρτηση network, θα το κάναμε με
αυτό τον τρόπο:

% net = network(1, 2, [1; 0], [1; 0], [0 0; 1 0], [0 1]);

% Ωστόσο χρησιμοποιήθηκε ο τρόπος του φυλλαδίου (Σελ.12),
% όπως τονίζεται στην εκφώνηση της Άσκησης.

```

Αποτελέσματα:

Εισάγετε τον αριθμό των προτύπων (άρτιος αριθμός): 100

Εισάγετε τον συντελεστή εκπαίδευσης (Learning Rate): 0.1

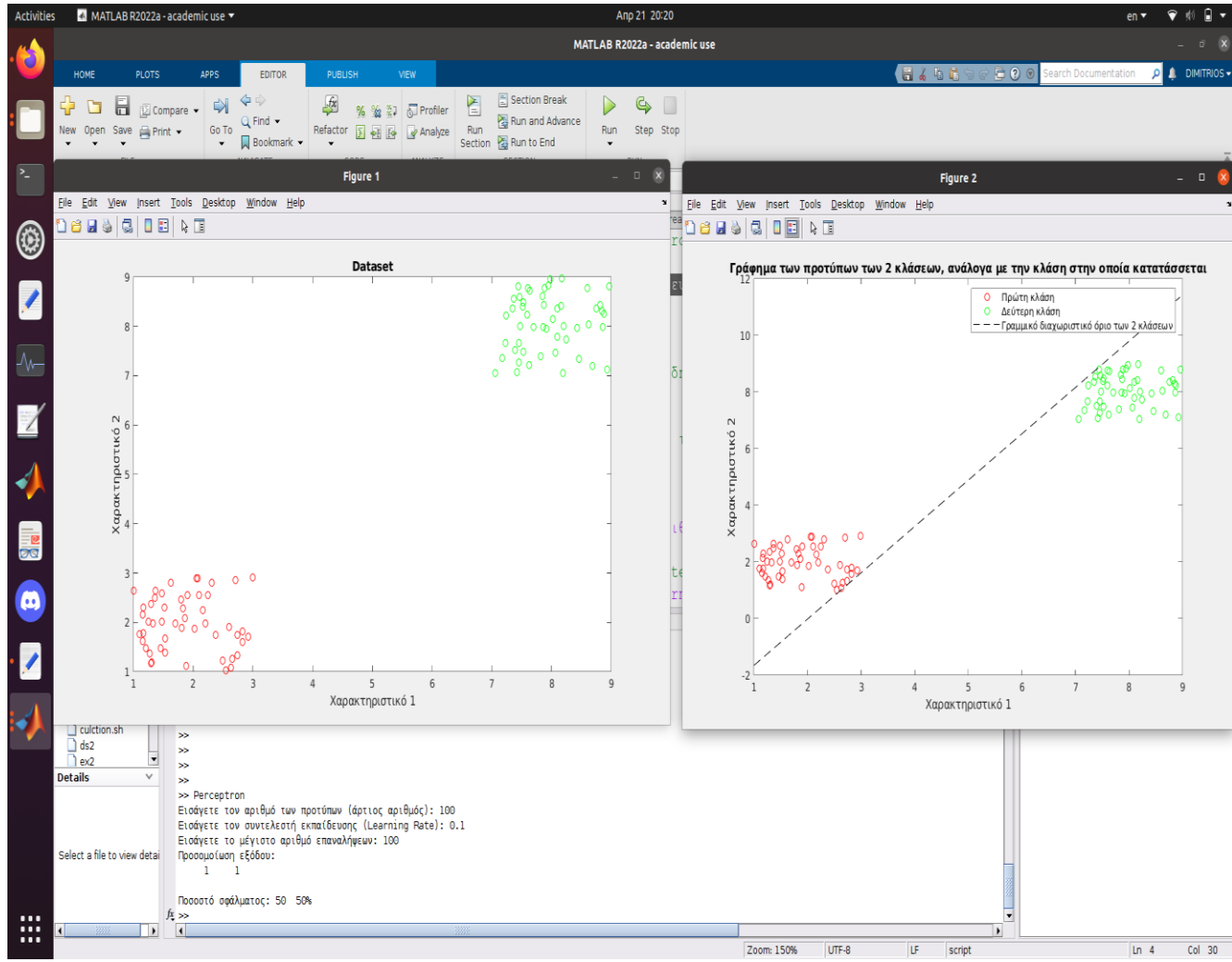
Εισάγετε το μέγιστο αριθμό επαναλήψεων: 100

Προσομοίωση εξόδου:

1 1

Ποσοστό σφάλματος: 50 50%

Γράφημα:



B2

Στο συγκεκριμένο ερώτημα λαμβάνει χώρα η δημιουργία ενός νευρωνικού δικτύου Adaline. Στη συνέχεια, εκπαιδεύουμε το νευρωνικό δίκτυο, προσομοιώνουμε την έξοδο του και υπολογίζουμε το σφάλμα του. Όπως σημειώνεται και στην εκφώνηση της εργασίας, ακολουθήσαμε τα βήματα και τις οδηγίες που βρίσκονται στο φυλλάδιο ERGASTHRIO_NEURAL_NETS, και συγκεκριμένα την άσκηση 3.2 που βρίσκεται στη σελίδα 18.

Κώδικας:

Αρχείο Adaline.m

```
% Διαχωρισμός Γραμμικά Διαχωρίσιμων Κλάσεων με Adaline

% Το Adaline αποτελείται από ένα νευρώνα τύπου McCulloch-Pitts
με n εισόδους και μία έξοδο.

% w :το διάνυσμα των συναπτικών βαρών

% x :το πρότυπο που εισάγεται κάθε φορά

% d :ο στόχος, η κλάση στην οποία ανήκει το πρότυπο με τιμές
0, 1.

% y :η έξοδος του νευρώνα με συνεχείς τιμές, σε αντίθεση με το
Perceptron.

% Το Adaline χρησιμοποιεί τη Γραμμική Συνάρτηση Ενεργοποίησης.

% Βήμα 1: Διάβασμα αριθμού προτύπων n (άρτιος αριθμός)
n = input('Εισάγετε τον αριθμό των προτύπων (άρτιος αριθμός):
');

% Βήμα 2: Διάβασμα συντελεστή εκπαίδευσης (Learning Rate) beta
beta = input('Εισάγετε τον συντελεστή εκπαίδευσης (Learning
Rate): ');
```

```

% Βήμα 3: Διάβασμα μέγιστου αριθμού επαναλήψεων
max_num_of_epochs

max_num_of_epochs = input('Εισάγετε το μέγιστο αριθμό
επαναλήψεων: ');

% Βήμα 4: Διάβασμα ελάχιστου μέσου τετραγωνικού σφάλματος
min_mean_squared_error

min_mean_squared_error = input('Εισάγετε το ελάχιστο μέσο
τετραγωνικό σφάλμα: ');

% Βήμα 5: Δημιουργία τυχαίων τιμών για τις συνάψεις w
w = randn(3, 1); % 2 είσοδοι και 1 bias

% Βήμα 6: Δημιουργεί με τη συνάρτηση rand τυχαίες τιμές για τα
πρότυπα p
p = zeros(n, 2);

% Δημιουργία προτύπων για την πρώτη κλάση ( $1 \leq x, y \leq 3$ )
p(1:n/2, :) = rand(n/2, 2) * 2 + 1;

% Δημιουργία προτύπων για τη δεύτερη κλάση ( $7 \leq x, y \leq 9$ )
p(n/2+1:n, :) = rand(n/2, 2) * 2 + 7;

% Βήμα 7: Δίνει τις τιμές 0, 1 για τους στόχους (επιθυμητοί
έξοδοι) d
d = zeros(n, 1);
d(1:n/2) = 1; % Στόχοι για την 1η κλάση
d(n/2+1:n) = -1; % Στόχοι για τη 1η κλάση

% Βήμα 8: Εμφάνιση γραφήματος των προτύπων
figure;

plot(p(1:n/2, 1), p(1:n/2, 2), 'ro'); % Η πρώτη κλάση
εμφανίζεται με κόκκινο

hold on;

plot(p(n/2+1:n, 1), p(n/2+1:n, 2), 'go'); % Η δεύτερη κλάση
εμφανίζεται με πράσινο

```

```

xlabel('Χαρακτηριστικό 1');
ylabel('Χαρακτηριστικό 2');
title('Dataset');

% Βήμα 9: Αρχικοποίηση εποχών και μέσου τετραγωνικού σφάλματος
epochs = 0;
mse = inf;

% Βήμα 10: Εκπαίδευση νευρωνικού δικτύου Adaline
%Η εκπαίδευση τελειώνει, όταν το Μέσο Τετραγωνικό Σφάλμα των
Προτύπων πάρει μια
% επιθυμητή τιμή.
while mse > min_mean_squared_error && epochs <
max_num_of_epochs
    mse = 0;
    for i = 1:n
        % Υπολογισμός εξόδου
        y = w(1) * p(i, 1) + w(2) * p(i, 2) + w(3);

        % Υπολογισμός του delta
        delta = d(i) - y;

        % Υπολογισμός του τετραγωνικού σφάλματος
        mse = mse + delta^2;

        % Ενημέρωση συνάψεων βάσει του κανόνα του Δέλτα
        w = w + beta * delta * [p(i, 1); p(i, 2); 1];
    end

    % Αύξηση των εποχών
    epochs = epochs + 1;

    % Εμφάνιση των προτύπων και του μέσου τετραγωνικού
    σφάλματος σε κάθε εποχή

```



```

figure;
subplot(2, 1, 1);
plot(p(1:n/2, 1), p(1:n/2, 2), 'ro');
hold on;
plot(p(n/2+1:n, 1), p(n/2+1:n, 2), 'go');
xlabel('Χαρακτηριστικό 1');
ylabel('Χαρακτηριστικό 2');
title('Γράφημα των προτύπων των 2 κλάσεων, ανάλογα με την
κλάση στην οποία κατατάσσονται');

```

```

subplot(2, 1, 2);
plot(1:epochs, mse/n, 'b*');
xlabel('Εποχή');
ylabel('MSE');
title('Γράφημα του Μέσου Τετραγωνικού Σφάλματος στην κάθε
εποχή');
drawnow;

```

```
end
```

```
% Βήμα 11: Ανάκληση(Δημιουργεί δύο τυχαία πρότυπα, ένα απ' την
κάθε κλάση)
```

```
test_samples = rand(2, 2);
```

```
% Βήμα 12:
```

```
for i = 1:2
```

```
    % Υπολογισμός εξόδου y
```

```
    y = w(1) * test_samples(i, 1) + w(2) * test_samples(i, 2)
    + w(3);
```

```
    % Κατηγοριοποίηση του προτύπου
```

```
    if y > 0
```

```
        fprintf('Το πρότυπο [%f, %f] ανήκει στην κλάση 1.\n',
test_samples(i, :));
```

```
    else
```

```

        fprintf('Το πρότυπο [%f, %f] ανήκει στην κλάση 2.\n',
test_samples(i, :));
    end
end
end

```

Αποτελέσματα:

Εισάγετε τον αριθμό των προτύπων (άρτιος αριθμός): 100

Εισάγετε τον συντελεστή εκπαίδευσης (Learning Rate): 0.1

Εισάγετε το μέγιστο αριθμό επαναλήψεων: 100

Εισάγετε το ελάχιστο μέσο τετραγωνικό σφάλμα: 0.01

Το πρότυπο [0.639317, 0.647311] ανήκει στην κλάση 2.

Το πρότυπο [0.544716, 0.543886] ανήκει στην κλάση 2.

Γραφήματα:

