



# ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ UNIVERSITY OF PIRAEUS

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΠΛΗΡΟΦΟΡΙΚΗΣ - ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΤΕΧΝΗΤΗΣ  
ΝΟΗΜΟΣΥΝΗΣ

## ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ

ΑΝΑΛΥΣΗ ΚΑΙ ΠΡΟΓΝΩΣΗ ΑΘΛΗΤΙΚΩΝ ΓΕΓΟΝΟΤΩΝ ΜΕ ΧΡΗΣΗ ΑΛΓΟΡΙΘΜΩΝ  
ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

---

Ομάδα ανάπτυξης:

Πατέρας Νικόλας – ΜΠΣΠ21043

Ζάρτηλας Βασίλειος – ΜΠΣΠ21015

*Πειραιάς, Απρίλιος 2022*

## Περιεχόμενα

1 Εισαγωγή.....	3
1.1 Αρχεία CSV.....	3
1.2 Main.py.....	3
1.3 Αρχεία προγράμματος.....	3
2 Ερώτημα πρώτο.....	4
2.1 Ζητούμενα.....	4
2.2 Υλοποίηση.....	4
Γενικά.....	4
Η συνάρτηση πρόβλεψης.....	4
Παράμετροι $w$ και $b$ .....	4
Ελαχιστοποίηση της συνάρτησης απώλειας.....	5
Ψευδοκώδικας.....	5
2.3 Αποτελέσματα.....	5
3 Ερώτημα δεύτερο και τρίτο.....	6
3.1 Ζητούμενα.....	6
Ερώτημα δεύτερο.....	6
Ερώτημα τρίτο.....	6
3.2 Υλοποίηση.....	6
Αποτελέσματα δεύτερου ερωτήματος.....	7
Αποτελέσματα τρίτου ερωτήματος.....	7
4 Ερώτημα τέταρτο.....	8
4.1 Ζητούμενα.....	8
4.2 Υλοποίηση.....	8
Ψευδοκώδικας.....	8
Αποτελέσματα.....	9
Στοιχηματική $b365$ .....	9
Στοιχηματική $bw$ .....	10
Στοιχηματική $iw$ .....	11
Στοιχηματική $lb$ .....	12
5 Εργαλεία.....	13
6 Βιβλιοθήκες.....	13

---

# 1 Εισαγωγή

---

## 1.1 Αρχεία CSV

Από το αρχείο *EuropeanSoccerDatabaseRetriever.m* έχουμε πάρει τα ωφέλιμα δεδομένα και τα έχουμε χωρίσει σε δύο αρχεία csv, το αρχείο Data.csv και Matches.csv. Με αυτόν το τρόπο θα μπορούσαμε να έχουμε καλύτερη διαχείριση των δεδομένων μας εφόσον είναι ογκώδες. Έγινε λοιπόν μία μικρή προεργασία.

## 1.2 Main.py

Για την ομαλή και γρήγορη απόδοση του κώδικα τοποθετήσαμε κάθε ερώτημα σε σχόλια. Αφαιρέστε τα σχόλια από το ερώτημα που θέλετε να τρέξετε για να εκτελεστεί.

## 1.3 Αρχεία προγράμματος

- Αρχείο linearNN.py → Γραμμικό νευρωνικό δίκτυο.
- Αρχείο multilayerNN.py → Πολυστρωματικό νευρωνικό δίκτυο.
- Αρχείο readData.py → Συνάρτηση φόρτωσης δεδομένων.
- Αρχείο main.py → Κύριο αρχείο προγράμματος.
- Αρχείο Data.csv και Matches.csv → Αρχεία δεδομένων τύπου CSV.
- Αρχείο simplePerceptron → Απλή υλοποίηση αλγορίθμου Perceptron για καλύτερη κατανόηση των νευρωνικών δικτύων.

---

## 2 Ερώτημα πρώτο

---

### 2.1 Ζητούμενα

Να υλοποιήσετε ένα γραμμικό νευρωνικό δίκτυο, ώστε ο εκπαιδευόμενος ταξινομητής να υλοποιεί μια συνάρτηση διάκρισης της μορφής  $g_k(\psi_k(m)): \mathbb{R}^3 \rightarrow \{H, D, A\}$  για κάθε στοιχηματική εταιρεία. Να αναγνωρίσετε την στοιχηματική εταιρεία τα προγνωστικά της οποίας οδηγούν σε μεγαλύτερη ακρίβεια ταξινόμησης.

### 2.2 Υλοποίηση

#### Γενικά

Για το γραμμικό νευρωνικό δίκτυο έχουμε ένα αλγόριθμο εποπτευόμενης μάθησης. Με άλλα λόγια έχουμε ένα σύνολο δεδομένων με  $X$  χαρακτηριστικά και  $y$  ετικέτες όπου θέλουμε να τα τοποθετήσουμε σε μία ευθεία γραμμή.

Αρχικά, έχουμε τα δεδομένα εκπαίδευσης που βρίσκονται στον αρχείο Matches.csv. Με αυτά τα δεδομένα θα τροφοδοτήσουμε τον αλγόριθμο μας. Πρακτικά αυτό που έχει να κάνει ο αλγόριθμος είναι να εξάγει μία συνάρτηση πρόβλεψης (ή υπόθεση) ( $h(x)$ ) όπου στην συνέχεια θα την χρησιμοποιήσει για να κάνει προβλέψεις.

#### Η συνάρτηση πρόβλεψης

Εφόσον θέλουμε μία ευθεία γραμμή θα έχουμε  $h(x) = wX + b$ , όπου  $w \rightarrow \text{weights}$  και  $b \rightarrow \text{bias}$ . Εάν μπορούμε να αναπαραστήσουμε τα δεδομένα μας σε 2 διαστάσεις τότε μπορούμε να πούμε ότι τα weights πρόκειται να είναι η κλίση της γραμμής μας και bias θα είναι η τομή  $y$ . Αλλά αν έχουμε δύο χαρακτηριστικά στα δεδομένα μας αντί για ένα, τότε θα αναπαραστήσουμε τα δεδομένα μας σε 3 διαστάσεις και έχουμε ένα επίπεδο για να χωρίζει τα δεδομένα μας στον τρισδιάστατο χώρο. Άρα, η υπόθεση μας δεν θα είναι μία ευθεία γραμμή αλλά ένα επίπεδο, καθώς ο αριθμός των χαρακτηριστικών αυξάνεται, οι διαστάσεις μας weights και bias αυξάνονται αντίστοιχα. Να συμπληρώσουμε ότι τα weights και bias είναι διανύσματα και οι διαστάσεις τους είναι ίσες με τον αριθμό των χαρακτηριστικών. Στόχος μας λοιπόν είναι να βρούμε τις τιμές weights και bias τέτοιες ώστε η  $h(x)$  να πλησιάζει όσο πιο κοντά γίνεται στο  $y$ .

#### Παράμετροι $w$ και $b$

Θα προσπαθήσουμε να επιλέξουμε τα  $w$  και  $b$  έτσι ώστε  $h(x)$  να είναι όσο το δυνατόν κοντά στο  $y$ . Άρα πρέπει να επιλέξουμε παραμέτρους τέτοιες ώστε ο αλγόριθμος μάθησης μας να υπολογίζει “σκορ” που να είναι κοντά στο σκετ εκπαίδευσης. Έτσι λοιπόν θα ορίσουμε μια συνάρτηση απώλειας ως το μέσο τετραγωνικό σφάλμα. Θέλουμε να βρούμε τις τιμές των  $w$  και  $b$  που ελαχιστοποιούν τη συνάρτηση απώλειας. Η  $h(x)$  είναι η τιμή που προβλέπει ο αλγόριθμος και  $y$  είναι η αληθινή αξία. Η συνάρτηση απώλειας είναι ένα μέτρο του πόσο κοντά είμαστε στην σωστή τιμή όπου είναι κι ο στόχος, είναι ένα μέτρο για το πόσο καλά λειτουργεί ο αλγόριθμος μας. Τέλος, όσο μικρότερη είναι η

απώλεια τόσο το καλύτερο και έτσι πάλι ο στόχος είναι να βρεθούν  $w$  και  $b$  έτσι ώστε να ελαχιστοποιούν την απώλεια.

## Ελαχιστοποίηση της συνάρτησης απώλειας

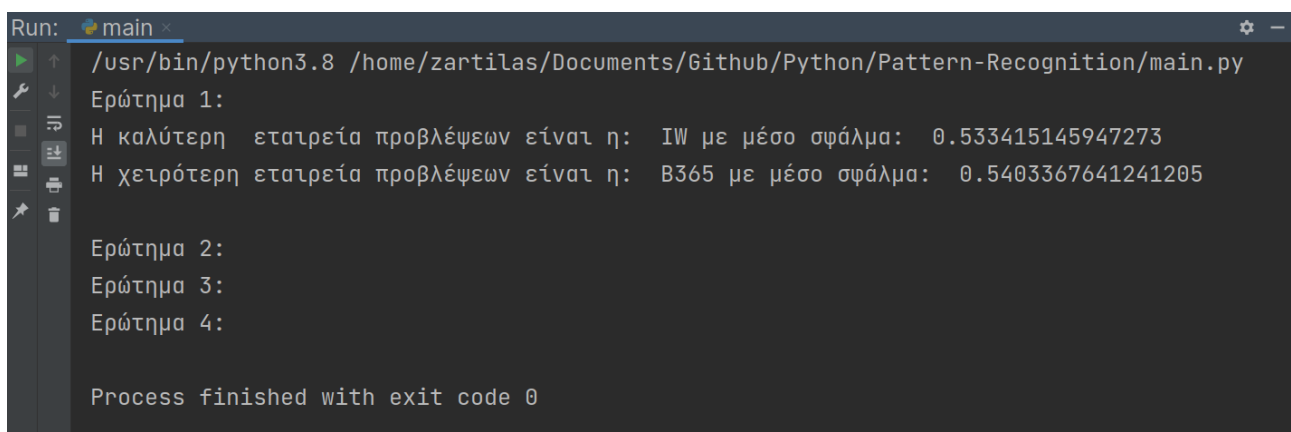
Η ελαχιστοποίηση της συνάρτησης απώλειας θα γίνει με τον αλγόριθμο Gradient Descent. Ας υποθέσουμε ότι έχουμε μία συνάρτηση κόστους/απώλειας  $J(b,w)$  και προσπαθούμε να βρούμε τις τιμές  $b$  και  $w$  που ελαχιστοποιούν την  $J$ . Αρχικά ο αλγόριθμος Gradient Descent αρχικοποιεί τις τιμές των  $b$  και  $w$  (συνήθως με μηδέν ή τυχαία). Μετέπειτα σε κάθε βήμα μας πριν προχωρήσουμε “κοιτάμε” 360 μοίρες για να δούμε αν θα κάνουμε ένα μικρό βήμα για να κατηφορήσουμε όσο το δυνατόν γρηγορότερα γίνεται. Το σημείο που θα κατηφορήσουμε θα είναι το χαμηλότερο σημείο της συνάρτησης αρά θα βρούμε το ελάχιστο  $J$  και έτσι θα έχουμε τις βέλτιστες τιμές των  $w$  και  $b$ . Γιαυτό σε κάθε βήμα ο αλγόριθμος Gradient Descent αναζητά το πιο απότομο σκαλί.

## Ψευδοκώδικας

1. Αρχικοποίηση βαρών  $w$  και  $b$
2. Ανάγνωση δεδομένων
3. Διαχωρισμός σε  $X$  (χαρακτηριστικά) και  $y$  (ετικέτες)
4. Υπολογισμός της  $h(x)$ 
  1. Υπολογισμός τις κλίσεις απώλειας ως προς τις παραμέτρους  $w$  και  $b$
  2. Ενημέρωση των  $w$  και  $b$
  3. Επανάλαβε τα παραπάνω σημεία όσο με το πλήθος των βημάτων που έγιναν στην κατηφόρα.

## 2.3 Αποτελέσματα

Βλέπουμε στην πιο κάτω φωτογραφία τα αποτελέσματα του κώδικα μας:



```
Run: main x
/usr/bin/python3.8 /home/zartilas/Documents/Github/Python/Pattern-Recognition/main.py
Ερώτημα 1:
Η καλύτερη εταιρεία προβλέψεων είναι η: IW με μέσο σφάλμα: 0.533415145947273
Η χειρότερη εταιρεία προβλέψεων είναι η: B365 με μέσο σφάλμα: 0.5403367641241205

Ερώτημα 2:
Ερώτημα 3:
Ερώτημα 4:

Process finished with exit code 0
```

## 3 Ερώτημα δεύτερο και τρίτο

### 3.1 Ζητούμενα

#### Ερώτημα δεύτερο

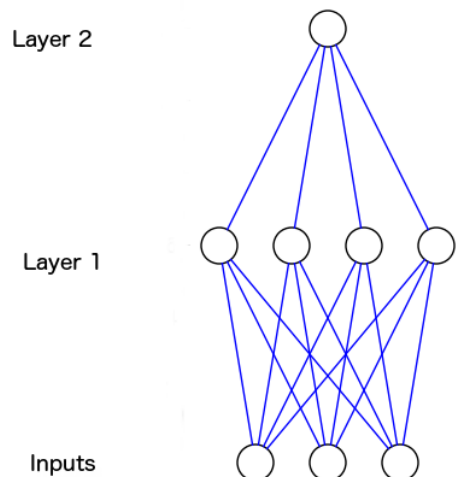
Να υλοποιήσετε ένα πολυστρωματικό νευρωνικό δίκτυο, ώστε ο εκπαιδευόμενος ταξινομητής να υλοποιεί μια συνάρτηση διάκρισης της μορφής  $g_k(\psi_k(m)): \mathbb{R}^3 \rightarrow \{H, D, A\}$  για κάθε στοιχηματική εταιρεία. Να αναγνωρίσετε την στοιχηματική εταιρεία τα προγνωστικά της οποίας οδηγούν σε μεγαλύτερη ακρίβεια ταξινόμησης.

#### Ερώτημα τρίτο

Να υλοποιήσετε ένα πολυστρωματικό νευρωνικό δίκτυο, ώστε ο εκπαιδευόμενος ταξινομητής να υλοποιεί μια συνάρτηση διάκρισης της μορφής  $g(\Phi(m)): \mathbb{R}^{28} \rightarrow \{H, D, A\}$ , όπου το  $\Phi(m) \in \mathbb{R}^{28}$  αντιστοιχεί στο πλήρες διάνυσμα χαρακτηριστικών του κάθε αγώνα που δίνεται από την σχέση:  $\Phi(m) = [\varphi(h), \varphi(a), \psi_{B365}(m), \psi_{BW}(m), \psi_{IW}(m), \psi_{LW}(m)]$

### 3.2 Υλοποίηση

Σε αυτά τα δύο υποερωτήματα έχουμε πιο περίπλοκα προβλήματα να λύσουμε. Το πολυστρωματικό νευρωνικό δίκτυο, όπως κι ονομάζεται, μας επιτρέπει να προσθέσουμε κρυφά στρώματα. Αυτά τα πρόσθετα επίπεδα επιτρέπουν στο νευρωνικό δίκτυο να σκεφτεί συνδυασμούς εισόδων. Στην εικόνα που ακολουθεί βλέπουμε ένα διάγραμμα όπου η έξοδος του Layer 1 τροφοδοτείται στο Layer. Έχοντας ένα επιπλέον στρώμα το νευρωνικό δίκτυο μπορεί να ανακαλύψει συσχετίσεις μεταξύ του Layer 1 και της εξόδου στο σύνολο εκπαίδευσης. Έτσι λοιπόν κατά την διάρκεια της εκπαίδευσης το νευρωνικό δίκτυο θα ενισχύσει αυτούς τους συσχετισμούς προσαρμόζοντας τα βάρη και στα δύο στρώματα.



## Αποτελέσματα δεύτερου ερωτήματος

Στην κλήση της συνάρτησης έχουμε βάλει την τιμή 3 στο όρισμα `input_layer` για να εξυπηρετεί την συνάρτηση διάκρισης.

```
Run: main x
/usr/bin/python3.8 /home/zartilas/Documents/Github/Python/Pattern-Recognition/main.py
Ερώτημα 1:
Ερώτημα 2:
Η καλύτερη εταιρεία προβλέψεων είναι η : BW με accuracy: 0.45726988308908284
Η χειρότερη εταιρεία προβλέψεων είναι η : B365 με accuracy: 0.4657225972885449

Ερώτημα 3:
Ερώτημα 4:

Process finished with exit code 0
```

## Αποτελέσματα τρίτου ερωτήματος

Στην κλήση της συνάρτησης έχουμε βάλει την τιμή 28 στο όρισμα `input_layer` για να εξυπηρετεί την συνάρτηση διάκρισης.

```
Run: main x
/usr/bin/python3.8 /home/zartilas/Documents/Github/Python/Pattern-Recognition/main.py
Ερώτημα 1:
Ερώτημα 2:
Ερώτημα 3:
Η συνάρτηση επιστέφει accuracy: 0.2535716420587005 και lowest loss: 1.0985038

Ερώτημα 4:

Process finished with exit code 0
```

---

## 4 Ερώτημα τέταρτο

---

### 4.1 Ζητούμενα

Να εφαρμόσετε τον αλγόριθμο ομαδοποίησης  $c$  – means επάνω στο σύνολο των διανυσμάτων προγνωστικών  $\Psi_k = \{\psi_k(\mathbf{m}) \in \mathbb{R}^3 : \mathbf{m} \in \mathbf{M}\}$  για κάθε στοιχηματική εταιρεία  $k \in \mathbf{B}$ , θέτοντας την τιμή του  $c$  ίση με 3. Με το τρόπο αυτό, θα παράξετε μια διαφορετική διαμέριση του συνόλου των αγώνων  $\mathbf{M}$  σε τρεις συστάδες για κάθε στοιχηματική εταιρεία. Λαμβάνοντας υπόψιν το αποτέλεσμα του κάθε αγώνα να υπολογίσετε την κατανομή των τριών αποτελεσμάτων εντός της κάθε συστάδας για κάθε στοιχηματική εταιρεία. Υπάρχει κάποιο αποτέλεσμα που να επικρατεί σε συχνότητα εντός της κάθε συστάδας;

### 4.2 Υλοποίηση

Ο αλγόριθμος K-Means είναι ένας πολύ δημοφιλής αλγόριθμος ομαδοποίησης. Η ομαδοποίηση με K-means ανήκει σε μία κατηγορία αλγορίθμων μάθησης χωρίς επίβλεψη και χρησιμοποιείται για την εύρεση συστάδων σε ένα σύνολο δεδομένων.

### Ψευδοκώδικας

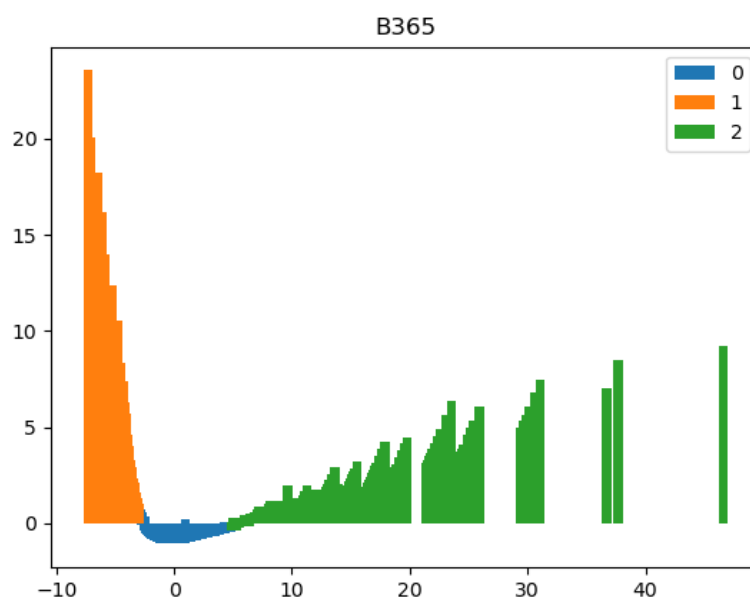
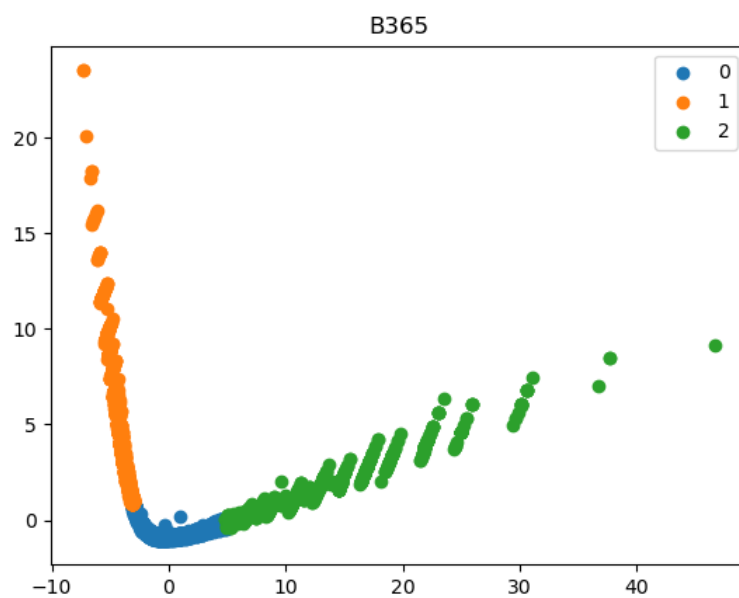
1. Επιλογή τυχαίων  $k$  σημείων για τα κεντροειδή
2. Υπολογισμός Ευκλείδειας απόστασης μεταξύ όλων των σημείων (από τα δεδομένα μας) από το σετ εκπαίδευσης με τα  $k$  κεντροειδή.
3. Αντιστοίχιση κάθε σημείου στο πλησιέστερο κεντροειδή σύμφωνα με την Ευκλείδεια απόσταση που υπολογίστηκε.
4. Ενημέρωση της θέσης του κέντρου μέσω του μέσου όρου από τα σημεία σε κάθε συστάδα.
5. Επανάληψη των βημάτων 2, 3 και 4 μέχρι τα κεντροειδή μας να μην αλλάζουν.

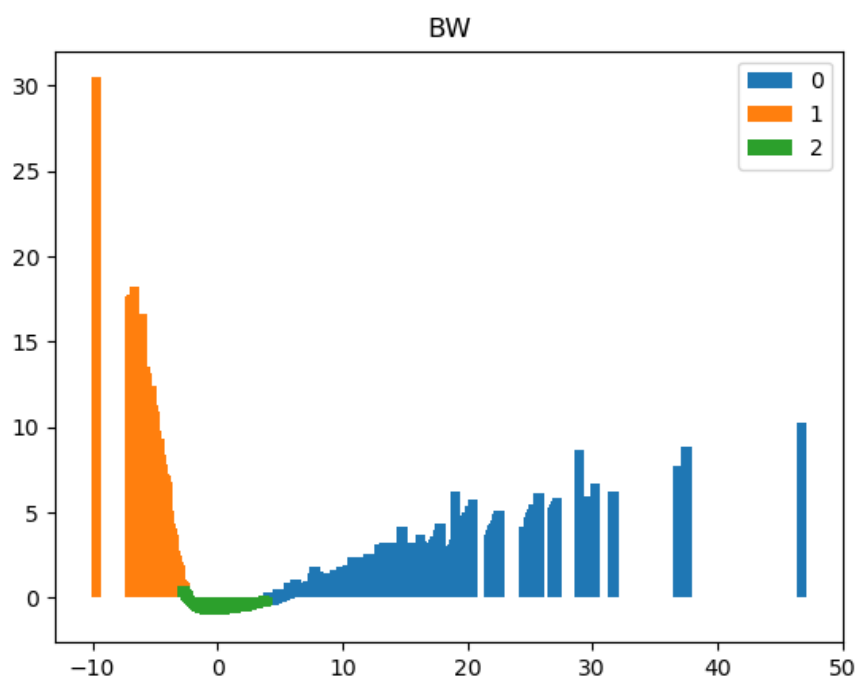
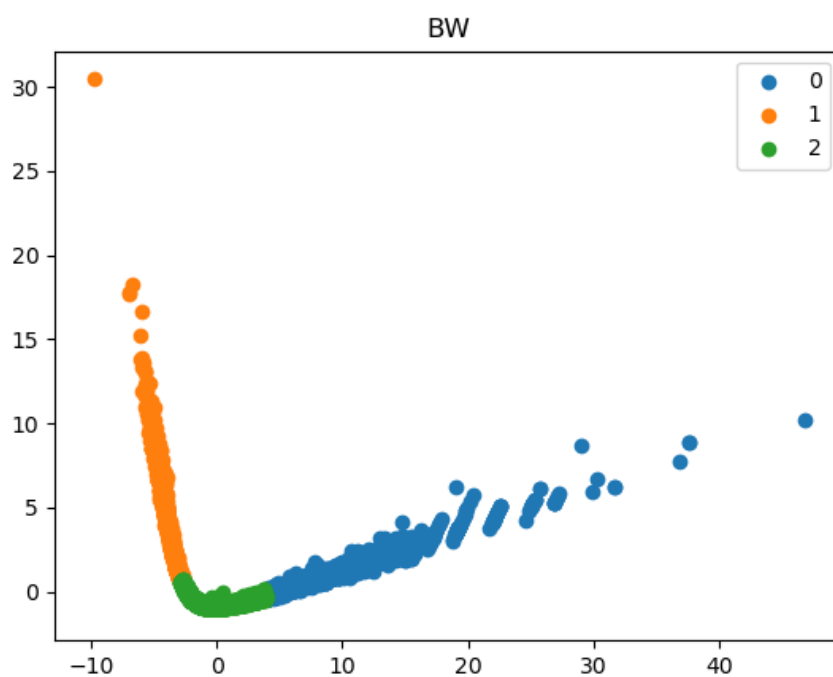


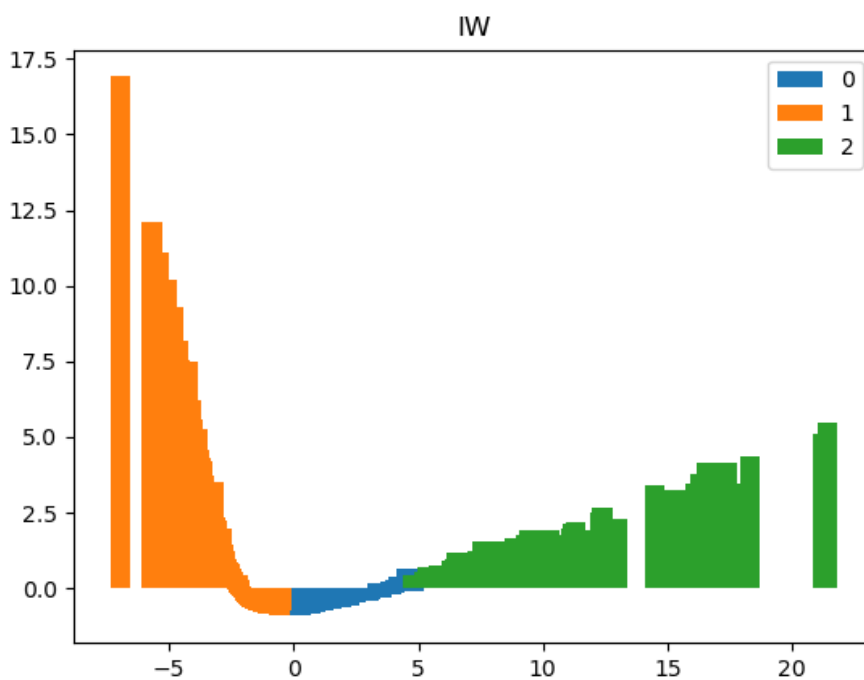
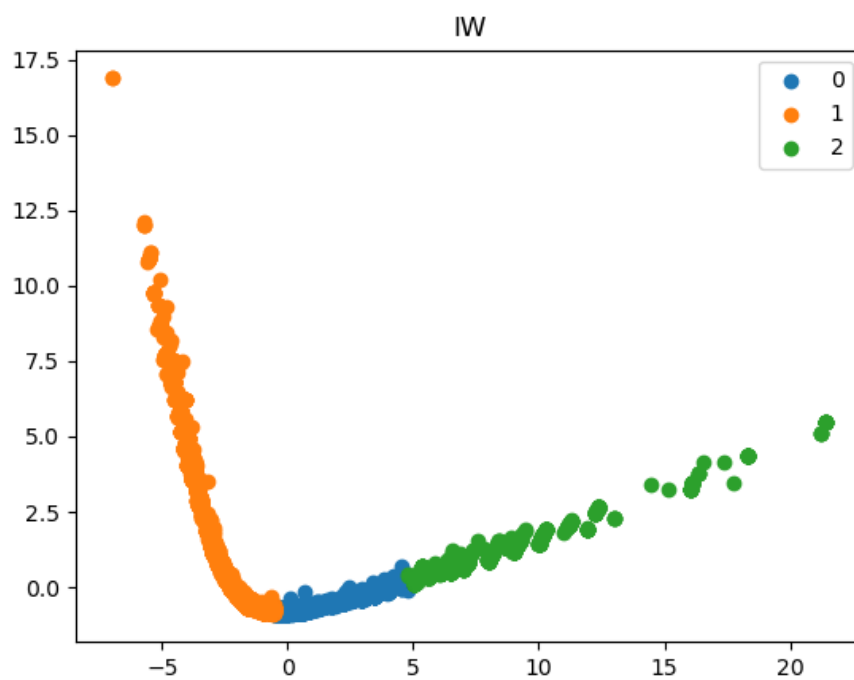
## Αποτελέσματα

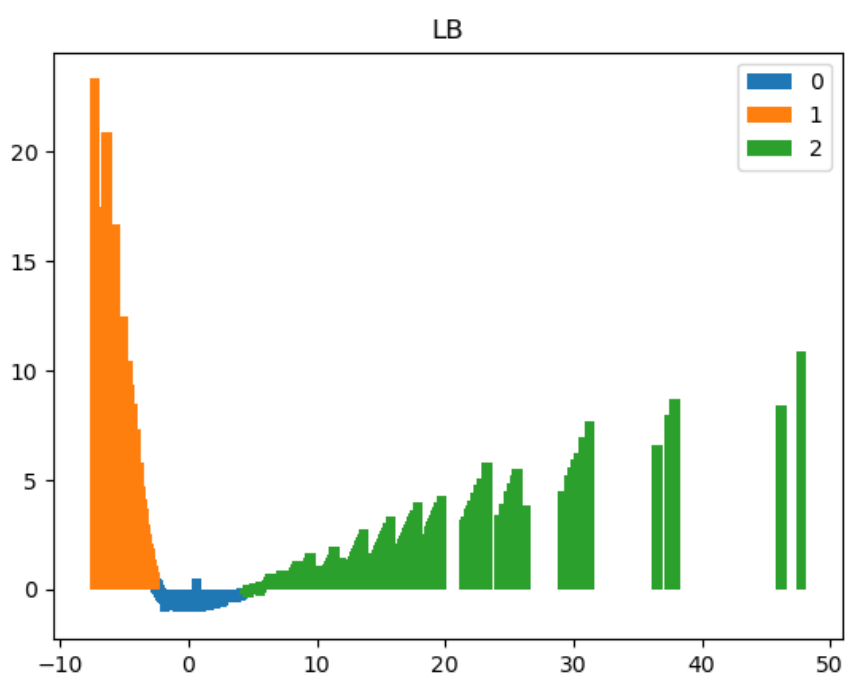
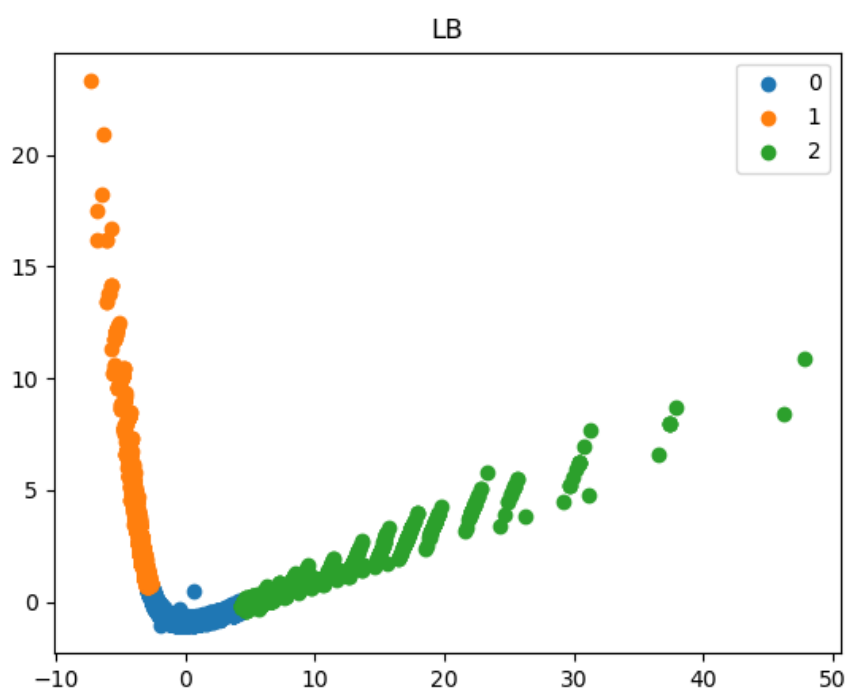
```
Run: main x
/usr/bin/python3.8 /home/zartilas/Documents/Github/Python/Pattern-Recognition/main.py
Ερώτημα 1:
Ερώτημα 2:
Ερώτημα 3:
Ερώτημα 4: Αναμονή για εμφάνιση γραφημάτων...
```

### Στοιχηματική b365



**Στοιχηματική bw**

**Στοιχηματική  $iW$** 

**Στοιχηματική lb**

---

## 5 Εργαλεία

---

Όνομα	Έκδοση
PyCharm	1.19.2
Python	3.8.10

---

## 6 Βιβλιοθήκες

---

Όνομα	Έκδοση
numpy	1.22
nnfs	0.5.1
sklearn	1.0.2
pandas	1.4.2
scipy	1.8.0
matplotlib	3.1.2