# ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ

Κείμενο τεκμηρίωσης τελικής εργασίας

Αντώνιος Ελευθέριος Καρναβάς – Π18063
Ελένη Τζάρα – Π18151

# Ερώτημα 1

Στο πρώτο ερώτημα προσεγγίσαμε το πρόβλημα της ταξινόμησης σύμφωνα με τον αλγόριθμο του ***Ελάχιστου Μέσου Τετραγωνικού Σφάλματος***. Γνωρίζουμε ότι ο βασικός, γενικός τύπος ενημέρωσης των βαρών κατά την διαδικασία του training είναι ο εξής : $w(n+1) = w(n) + \mu x(n)[d^*(n) - x^h(n)w(n)] = w(n) + \mu x(n)e^*(n)$ . Η παραπάνω προσέγγιση είναι γενικευμένη καθώς αφορά πρόβλημα ταξινόμησης δύο κλάσεων (δυαδικό πρόβλημα ταξινόμησης ).Επομένως στη δική μας περίπτωση της ταξινόμησης 3 κλάσεων (Home,Draw,Away) έπρεπε να κάνουμε μια τροποποίηση στον παραπάνω τύπο. Έτσι λοιπόν χρησιμοποιήσαμε το one hot vector encoding ώστε να μας βοηθήσει να αντιμετωπίσουμε το πρόβλημα ως ένα τριταξικό πρόβλημα ταξινόμησης. Αυτό μας οδηγεί επίσης να ορίσουμε τον πίνακα των βαρών ως ένα πίνακα διαστάσεων 3 γραμμών και 4 στηλών ο οποίος πίνακας αναλύεται περαιτέρω ως ένας πίνακας 3x3 βαρών και 3x1 βαρών κατωφλίου (bias terms).Ο πίνακας d είναι αυτός ο οποίος θα φτιαχτεί βάση του one hot vector encoding με 3 στήλες που η κάθε γραμμή θα έχει ως εξής: 1 0 0 (Home win),0 1 0 (Draw),0 0 1 (Away win). Το μ είναι το learning rate το οποίο κάθε φορά που θα ανανεώνονται τα βάρη αυτό θα μικραίνει τείνοντας προς το 0 έτσι ώστε η διαδικασία υπολογισμού των βαρών κάποτε να παγώσει καθώς το error( $d^*(n) - x^h(n)w(n)$ ) θα μηδενιστεί. Λόγω όλων των παραπάνω ο τελικός τύπος θα χρησιμοποιηθεί 12 φορές σε κάθε επανάληψη έτσι ώστε να ανανεώνονται τα 3x4=12 βάρη του πίνακα.

```
w(1,1)=w(1,1)+a*betarray(i,1)*(r(i,1) -(betarray(i,1)'* w(1,1)));
w(1,2)=w(1,2)+a*betarray(i,2)*(r(i,1) -(betarray(i,2)'* w(1,2)));
w(1,3)=w(1,3)+a*betarray(i,3)*(r(i,1) -(betarray(i,3)'* w(1,3)));
w(1,4)=w(1,4)+a*betarray(i,4)*(r(i,1) -(betarray(i,4)'* w(1,4)));

w(2,1)=w(2,1)+a*betarray(i,1)*(r(i,2) -(betarray(i,1)'* w(2,1)));
w(2,2)=w(2,2)+a*betarray(i,2)*(r(i,2) -(betarray(i,2)'* w(2,2)));
w(2,3)=w(2,3)+a*betarray(i,3)*(r(i,2) -(betarray(i,3)'* w(2,3)));
w(2,4)=w(2,4)+a*betarray(i,4)*(r(i,2) -(betarray(i,4)'* w(2,4)));

w(3,1)=w(3,1)+a*betarray(i,1)*(r(i,3) -(betarray(i,1)'* w(3,1)));
w(3,2)=w(3,2)+a*betarray(i,2)*(r(i,3) -(betarray(i,2)'* w(3,2)));
w(3,3)=w(3,3)+a*betarray(i,3)*(r(i,3) -(betarray(i,3)'* w(3,3)));
w(3,4)=w(3,4)+a*betarray(i,4)*(r(i,3) -(betarray(i,4)'* w(3,4)));
```

Όπου betarray είναι ο πίνακας x, r είναι ο πίνακας d και το a είναι το μ.

Όσον αφορά το training και το testing θα χρησιμοποιήσουμε την μέθοδο του 10-fold-cross validation.Η διαδικασία αυτή έχει ως εξής:

Χωρίζουμε το σύνολο των δεδομένων μας σε 10%(testing) και 90%(training), όπου κάθε φορά το σύνολο του testing θα αφορά διαφορετικό 10% των συνολικών δεδομένων. Αυτό σημαίνει ότι ο LMS θα τρέξει 10 φορές λύνοντας το ίδιο πρόβλημα με διαφορετικά trainingκαι testing σετ 10 φορές. Αυτό το κάνουμε διότι το αποτέλεσμα μας θα είναι πιο ακριβές από ότι θα ήταν εάν ανακατεύαμε τα δεδομένα και τρέχαμε μια φορά των LMS πάλι με το 10% για testing και το 90% για training.

Για το testing πολλαπλασιάζουμε το εκάστοτε διάνυσμα βαρών του πίνακα x στη δική μας περίπτωση το betarray , με τον ανεστραμμένο πίνακα των βαρών, με αποτέλεσμα να έχουμε έναν πίνακα διαστάσεων 1x3 ο οποίος θα είναι η πρόβλεψη που μας έδωσε ο αλγόριθμος LMS έχοντας έτσι εφαρμόσει το one hot vector encoding επιτυχώς.

# Ερώτημα 2

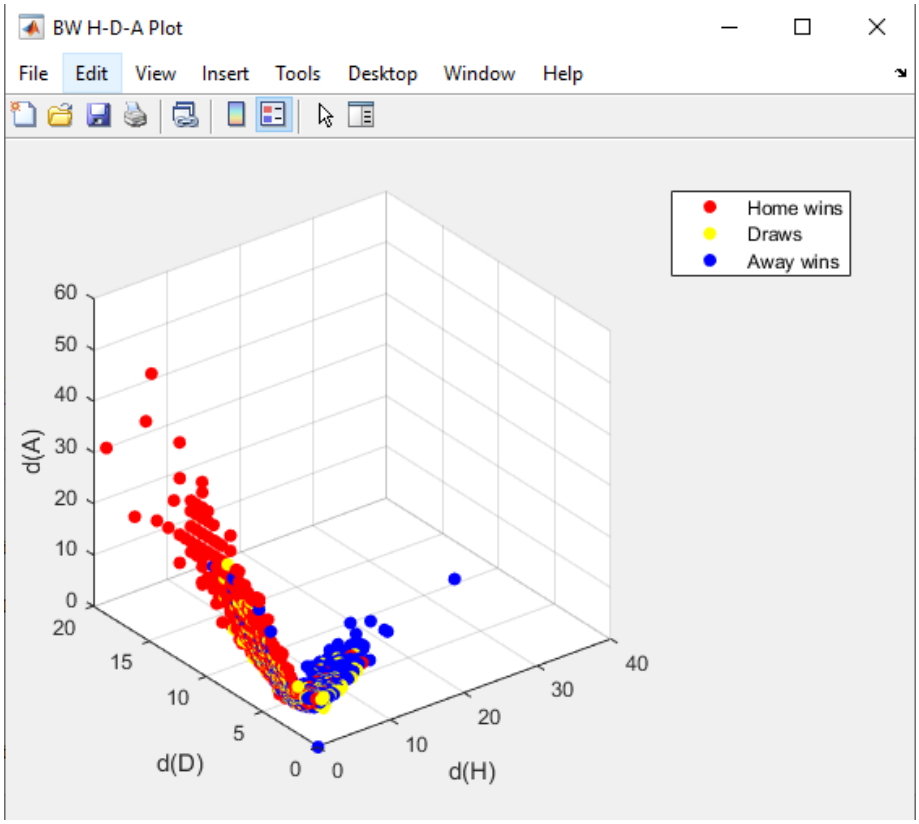
Στο δεύτερο ερώτημα προσεγγίσαμε το πρόβλημα της ταξινόμησης σύμφωνα με τον αλγόριθμο του ***Ελάχιστου Τετραγωνικού Σφάλματος***. Γνωρίζουμε πως ο παραπάνω αλγόριθμος είναι το άθροισμα των τετραγωνικών σφαλμάτων δηλαδή το άθροισμα του error( $d^*(n) - x^h(n)w(n)$ ) στο τετράγωνο. Αν ελαχιστοποιήσουμε το παραπάνω άθροισμα δηλαδή το ορίσουμε ίσο με 0 και λύσουμε ως προς το w τότε ο τύπος για να το ορίσουμε θα είναι ο εξής: ( $(x^h * x)^{-1} * x^h * d$ ). Η διαδικασία που ακολουθούμε είναι ίδια με το Ερώτημα 1 όσον αφορά το one hot vector encoding και το 10-fold-cross-validation αλλά η μονή διαφορά είναι ότι ο υπολογισμός του 3x4 πίνακα των βαρών δεν γίνεται επαναληπτικά για κάθε διάνυσμα χαρακτηριστικών του πίνακα x (betarray) αλλά υπολογίζεται με τη μια σύμφωνα με τον παραπάνω τύπο, δηλαδή ο τύπος αυτός εμπλέκει ολόκληρο τον πίνακα των διανυσμάτων των χαρακτηριστικών και όλο τον πίνακα των πραγματικών αποτελεσμάτων y(r). Τέλος, εννοείται πως δεν υπάρχει λόγος ύπαρξης του learning rate, καθώς η χρήση του αφορά αποκλειστικά το πάγωμα της επαναληπτικής ανανέωσης του πίνακα των βαρών .

# Οπτικοποίηση Δεδομένων

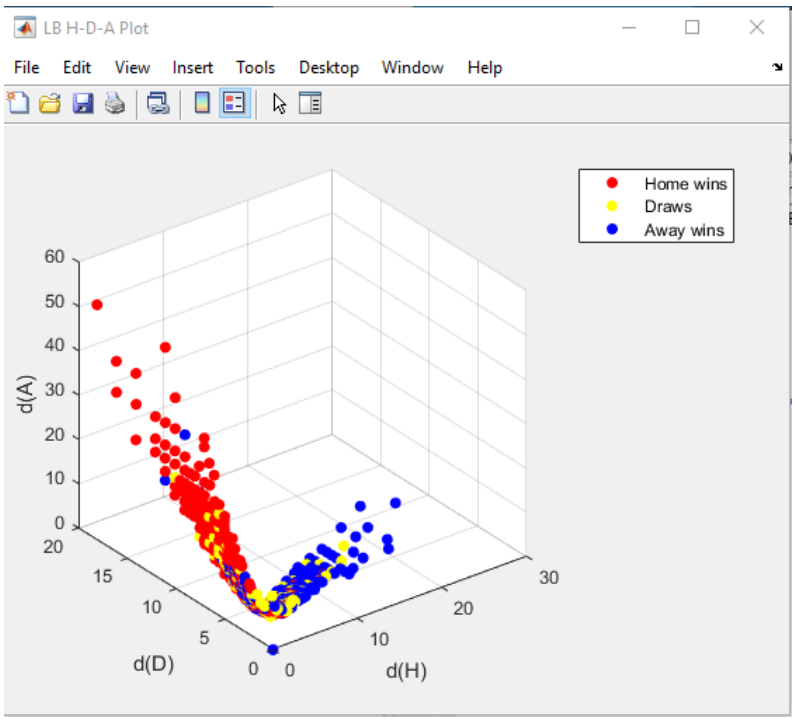
Η οπτικοποίηση που ακολουθεί αφορά και τα 2 ερωτήματα εφόσον μας δείχνει το εξής: σχεδιάζουμε ένα καρτεσιανό 3 διαστάσεων ( x,y,z) όπου x είναι η τιμή της απόδοσης για Home win, y της ισοπαλίας και z του Away win, του εκάστοτε διανύσματος χαρακτηριστικών. Τα σημεία που προκύπτουν από τις παραπάνω συντεταγμένες είναι χρώμα κόκκινο αν η πραγματική έκβαση του αγώνα είναι home win, χρώμα κίτρινο αν η πραγματική έκβαση του αγώνα είναι draw και χρώμα μπλε αν η πραγματική έκβαση του αγώνα είναι away win.

Έτσι για τις 4 στοιχηματικές έχουμε τα παρακάτω 4 διαγράμματα:

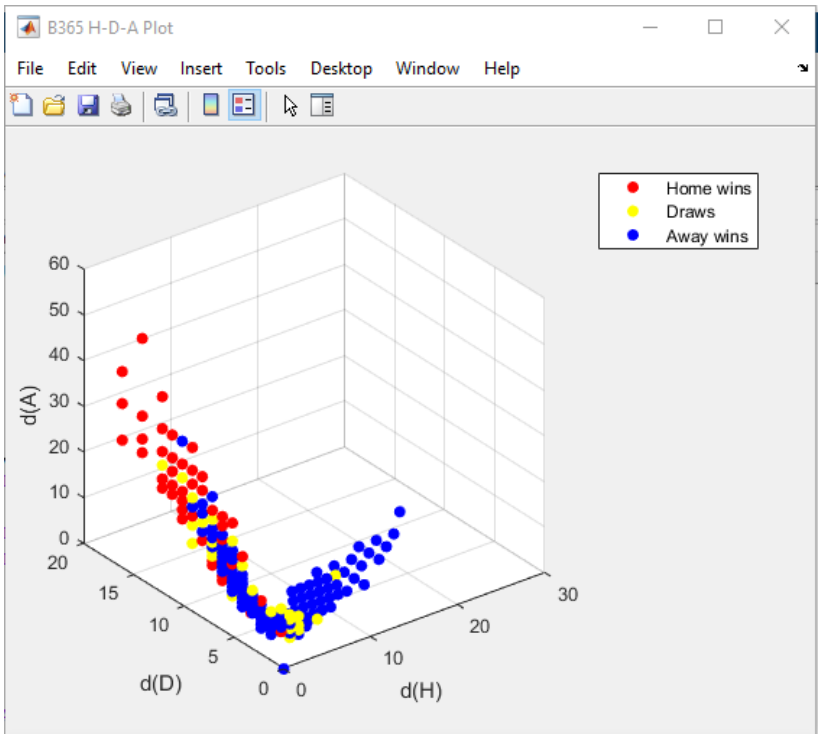
Για την BW:



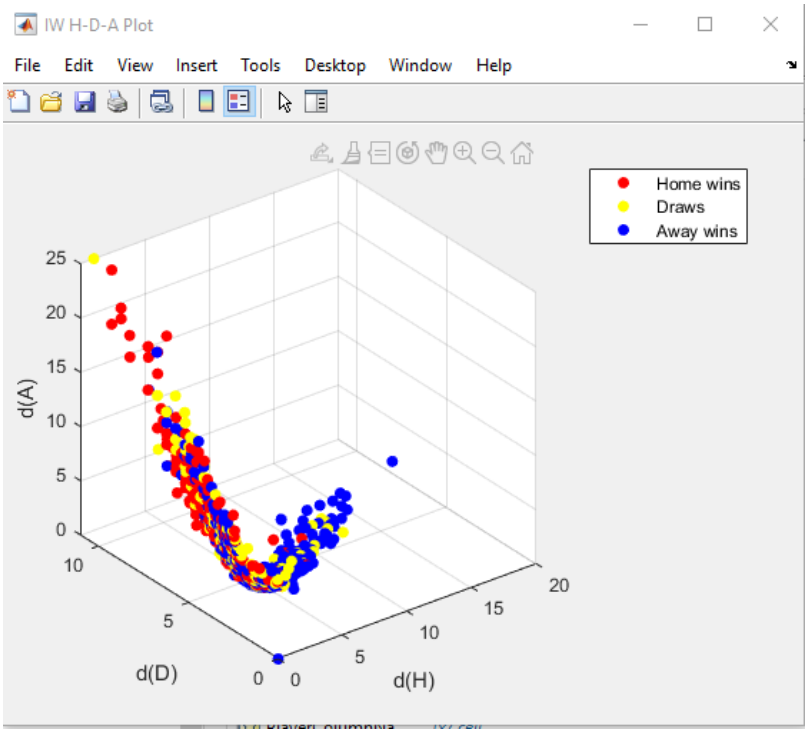
Για την LB:



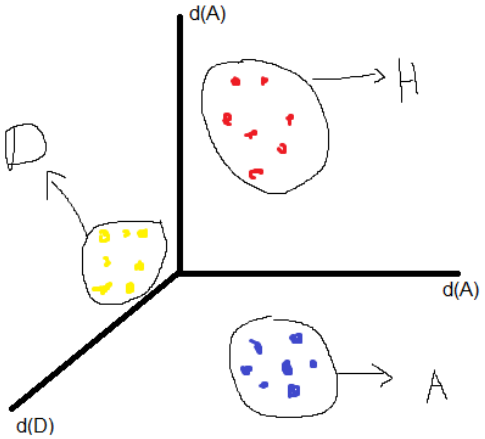

Για την B365:



Για την IW:




Η ουσία των παραπάνω σχημάτων συνοψίζεται εννοιολογικά στο παρακάτω σχήμα αλλά η διαχωρισιμότητα δεν υπάρχει τόσο έντονη στα 4 παραπάνω καθώς οι αποδόσεις των στοιχηματικών δεν αντιπροσωπεύουν πάντα την πραγματική έκβαση του αγώνα αφού κάνουν πολλές φορές λάθος!



# Ταξινομητική ακρίβεια

Τις 4 ακρίβειες των 4 στοιχηματικών και των 2 αλγορίθμων (lms,mse) θα τις αναπαραστήσουμε στον παρακάτω πίνακα. Για τον lms η ταξινομητική ακρίβεια προέρχεται από την αρχική τιμή του learning rate στο 0.5 και κάθε φορά ανανέωσης των βαρών το learning rate γίνεται learning rate/k ,όπου k είναι k-οστή φορά που ανανεώνονται τα βάρη.

| Στοιχηματικές/Αλγόριθμοι | Least mean square | Mean square |
|---|---|---|
| B365 | 49.1158% | 55.3492% |
| BW | 48.6295% | 55.0398% |
| IW | 50.4863% | 54.863% |
| LB | 50% | 54.9514% |