2021

ΕΡΓΑΣΙΑ ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ 2020-2021

Κορώνης Ευάγγελος(Π17050)

Περιεχόμενα

Εισαγωγή

Η παρούσα εργασία έχει υλοποιηθεί σε γλώσσα python. Απαραίτητες βιβλιοθήκες για την λειτουργία των προγραμμάτων είναι οι εξής:

- numpy
- sqlite3
- sklearn
- padasip
- matplotlib

Επίσης μέσα στον ίδιο φάκελο με τα προγράμματα θα πρέπει να βρίσκεται και το αρχείο database.sqlite .

10-fold cross validation

Σε όλα τα ερωτήματα χρησιμοποιείται η τεχνική 10-fold cross validation. Σύμφωνα με αυτή την μέθοδο τα δεδομένα χωρίζονται σε 10 ισομερή γκρουπ. Σε κάθε επανάληψη 9 από τα 10 γκρουπ μπαίνουν στα δεδομένα εκπαίδευσης ,ενώ τα δεδομένα του ενός γκρουπ που απομένει, γίνονται τα δεδομένα του test. Έτσι ο αλγόριθμος ταξινόμησης εκπαιδεύεται και τεστάρεται 10 φορές, ώστε το κάθε γκρουπ να έχει μπει μια φορά στην διαδικασία του testing. Στο τέλος της διαδικασίας βρίσκεται ο μέσος όρος της ταξινομικής ακρίβειας όλων των επαναλήψεων.

Ερώτημα 1

Το ερώτημα υλοποιείται από το πρόγραμμα **lms.py** .

Σε αυτό το ερώτημα αντλούμε δεδομένα από τον πίνακα Match από το αρχείο database.sqlite. Συγκεκριμένα για το διάνυσμα χαρακτηριστικών παίρνουμε τις τιμές από τις στήλες **kH, kD, kA** ,όπου $\boldsymbol{k} \boldsymbol{\epsilon}\{\boldsymbol{B}365, \boldsymbol{BW}, \boldsymbol{IW}, \boldsymbol{LB}\}$**.**

Το αποτέλεσμα του κάθε αγώνα υπολογίζεται από την διαφορά των γκολ που παίρνουμε από τις στήλες **home_team_goal** και **away_team_goal** του πίνακα Match .

Ο αλγόριθμος που χρησιμοποιείται είναι αυτός του Ελάχιστου Μέσου Τετραγωνικού Σφάλματος (**Least Mean Squares**) ,ώστε ο ταξινομητής να υλοποιεί την συνάρτηση διάκρισης

$$\boldsymbol{g(kH, kD, kA)}: \boldsymbol{R^3 \rightarrow \{H, D, A\}}$$

για κάθε στοιχηματική εταιρεία.

Η συνάρτηση διάκρισης είναι της μορφής **$g(x) = w^T x + w_0 = 0$.** Στην δικιά μας περίπτωση τα διανύσματα είναι της μορφής $\boldsymbol{x \in R^3, w \in R^3, w_0 \in R}$.

Ο στόχος του αλγορίθμου είναι να βρει το διάνυσμα βαρών **W(w και w_0)** για το οποίο η συνάρτηση κόστους $\boldsymbol{J(w) = E[|y - x^T w|^2]}$ (το μέσο τετραγωνικό σφάλμα ανάμεσα στην επιθυμητή έξοδο και την έξοδο του ταξινομητή) να ελαχιστοποιείται.

Το βέλτιστο διάνυσμα βαρών υπολογίζεται επαναληπτικά για κάθε διάνυσμα εισόδου που δίνουμε στον ταξινομητή από τον τύπο:

$$\boldsymbol{\hat{w}(k) = \hat{w}(k-1) + \rho_k x_k (y_k - x_k^T \hat{w}(k-1))}$$

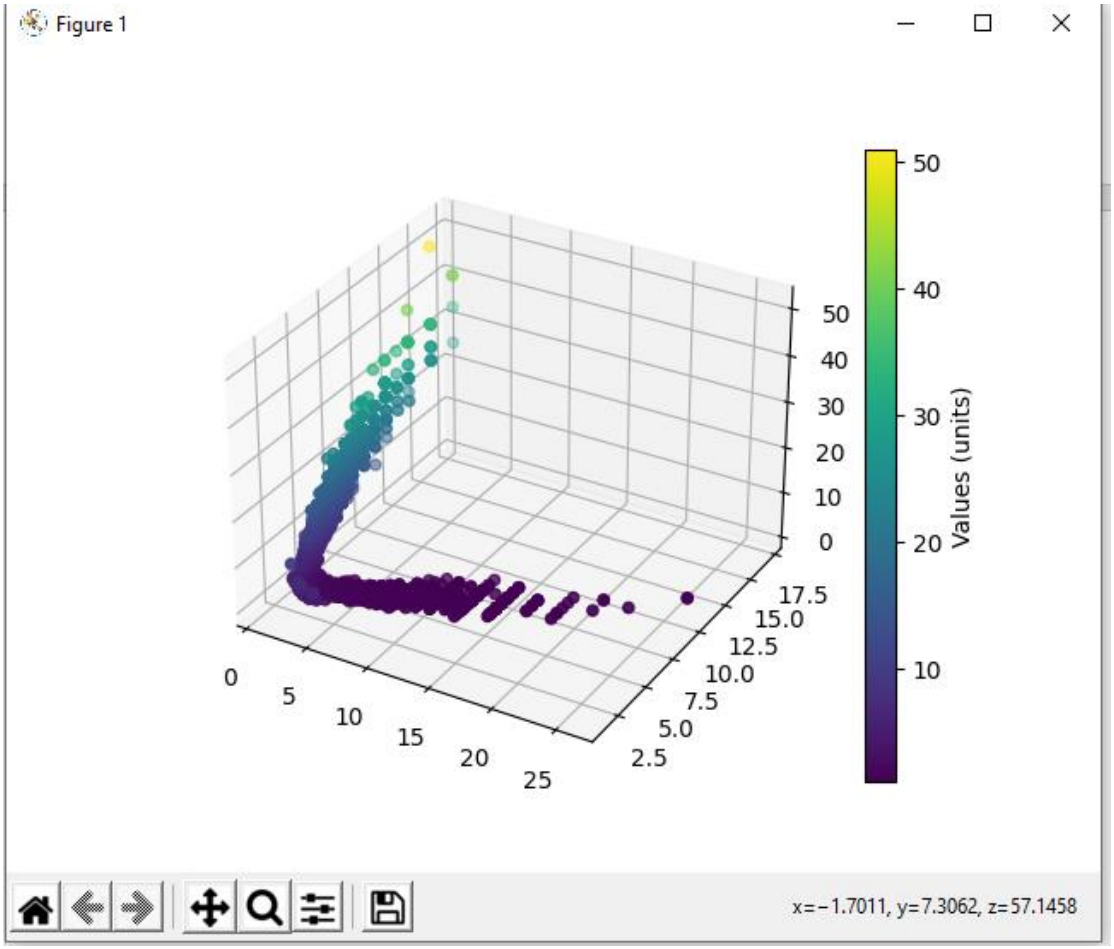
Όπου

- $x_k^T \hat{w}(k-1)$ η έξοδος του ταξινομητή για το διάνυσμα χαρακτηριστικών της παρούσας επανάληψης και για το διάνυσμα βαρών της προηγούμενης επανάληψης.
- ρ_k μια μηδενική ακολουθία θετικών όρων όπου η σειρά της συγκλίνει στο άπειρο και η σειρά του τετραγώνου της συγκλίνει.
- y_k η επιθυμητή έξοδος για το διάνυσμα χαρακτηριστικών της παρούσας επανάληψης.

Ο αλγόριθμος εκτελείται για κάθε στοιχηματική εταιρεία ξεχωριστά. Επιπλέον χρησιμοποιείται η τεχνική **10 fold cross validation**.

Επίσης σε αυτό το ερώτημα χρησιμοποιούμε την μέθοδο **one vs all**,δηλαδή υπολογίζουμε για κάθε κλάση την συνάρτηση διάκρισης μεταξύ του εαυτού της και των άλλων δύο κλάσεων.

Για κάθε στοιχηματική εταιρεία παίρνουμε τις 3 αποδόσεις που δίνει για κάθε αγώνα (για νίκη γηπεδούχου ,ισοπαλία και νίκη φιλοξενούμενου).Με βάση αυτά τα χαρακτηριστικά εκπαιδεύουμε τον ταξινομητή.Η κατανομή των χαρακτηριστικών στον τρισδιάστατο χώρο φαίνεται στο παρακάτω γράφημα.



Επειδή βρίσκουμε 3 συναρτήσεις διακρίσεων η έξοδος για το αποτέλεσμα του κάθε αγώνα αντιστοιχίζεται στην συνάρτηση που επέστρεψε το μεγαλύτερο αποτέλεσμα.

Χρησιμοποιούμε τον υλοποιημένο ,από την **padasip** βιβλιοθήκη, αλγόριθμο **FilterNLMS**.

Τα αποτελέσματα από την διαδικασία του training και του testing φαίνεται στην παρακάτω εικόνα.

```
========== RESTART: C:\Users\user\Desktop\pattern recognition\lms.py =======
1. Computing for B365 odds and results...
Mean training score: 52.08234931959502 %
Successfull predictions mean rate 52.12465183513732 %

2. Computing for BW odds and results...
Mean training score: 52.063494444824315 %
Successfull predictions mean rate 52.079992841874095 %

3. Computing for IW odds and results...
Mean training score: 51.713045194395114 %
Successfull predictions mean rate 51.69626998223801 %

4. Computing for LB odds and results...
Mean training score: 51.96991460781252 %
Successfull predictions mean rate 51.99530004245884 %
```

Όπως βλέπουμε το ποσοστό των σωστών προβλέψεων είναι περίπου το ίδιο για όλες τις στοιχηματικές εταιρίες, με μεγαλύτερη ακρίβεια να έχει η B365.

Ερώτημα 2

Το ερώτημα υλοποιείται από το πρόγραμμα **least_squares.py** .

Σε αυτό το ερώτημα αντλούμε δεδομένα από τον πίνακα Match από το αρχείο database.sqlite. Συγκεκριμένα, για το διάνυσμα χαρακτηριστικών παίρνουμε τις τιμές από τις στήλες **kH, kD, kA** , όπου $\boldsymbol{k} \in \{\boldsymbol{B}365, \boldsymbol{BW}, \boldsymbol{IW}, \boldsymbol{LB}\}$**.**

Το αποτέλεσμα του κάθε αγώνα υπολογίζεται από την διαφορά των γκολ που παίρνουμε από τις στήλες **home_team_goal** και **away_team_goal** του πίνακα Match .

Ο αλγόριθμος που χρησιμοποιείται είναι αυτός του Ελάχιστου Τετραγωνικού Σφάλματος (**Least Squares**) ,ώστε ο ταξινομητής να υλοποιεί την συνάρτηση διάκρισης $\boldsymbol{g}(\boldsymbol{kH}, \boldsymbol{kD}, \boldsymbol{kA})$: $\boldsymbol{R}^3 \rightarrow \{\boldsymbol{H}, \boldsymbol{D}, \boldsymbol{A}\}$, για κάθε στοιχηματική εταιρεία.

Η συνάρτηση διάκρισης είναι της μορφής $\mathbf{g(x) = w^T x + w_0 = 0}$. Στην δικιά μας περίπτωση τα διανύσματα είναι της μορφής $\boldsymbol{x} \in \boldsymbol{R}^3$,$\boldsymbol{w} \in \boldsymbol{R}^3$,$\boldsymbol{w}_0 \in \boldsymbol{R}$

Ο στόχος του αλγορίθμου είναι να βρει το διάνυσμα βαρών **W(w και $\mathbf{w_0}$)** έτσι ώστε η συνάρτηση κόστους $\boldsymbol{J}(\boldsymbol{w}) = \sum_{i=1}^{N} (\boldsymbol{y_i} - \boldsymbol{x_i^T w})^2$

(δηλαδή το άθροισμα των τετραγωνικών σφαλμάτων ανάμεσα στην επιθυμητή έξοδο και στην έξοδο του ταξινομητή για κάθε διάνυσμα χαρακτηριστικών εκπαίδευσης) να ελαχιστοποιείται.

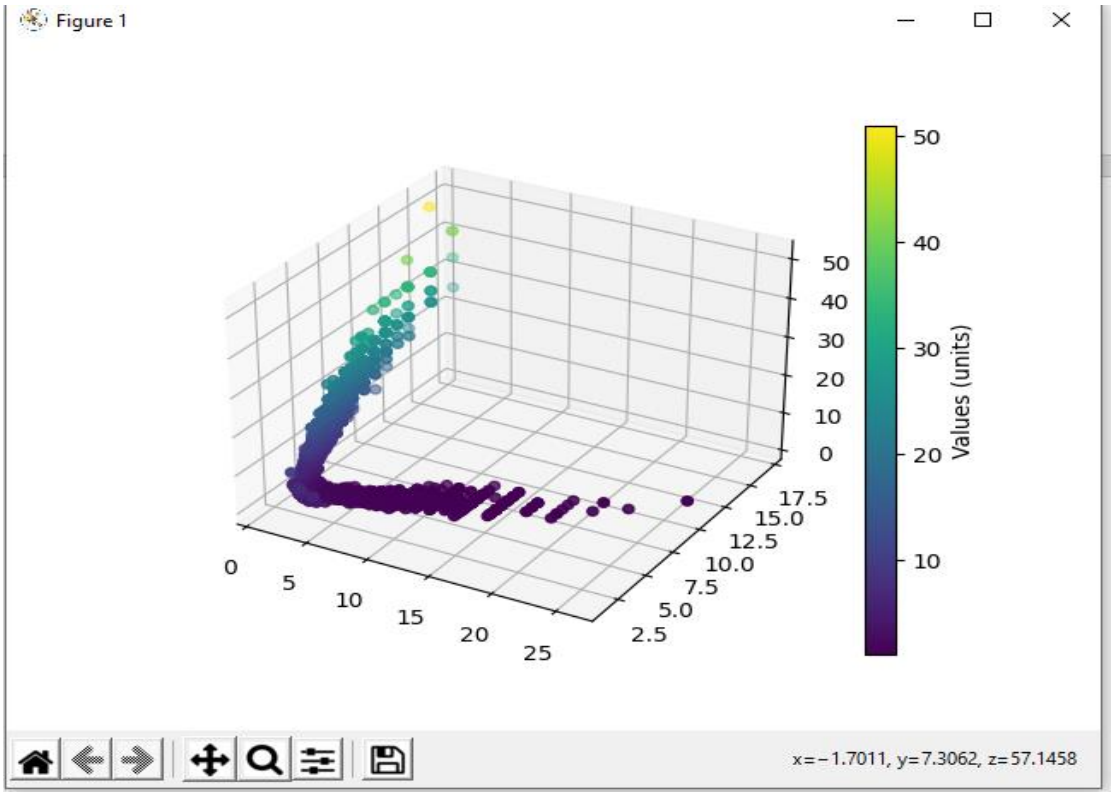
Το σημείο που ελαχιστοποιείται αυτή η συνάρτηση είναι το σημείο όπου μηδενίζεται η μερική παράγωγος της πρώτης τάξης ως προς το **W.**

Έτσι καταλήγουμε ότι το βέλτιστο διάνυσμα βαρών υπολογίζεται από την εξίσωση :

$$\widehat{\boldsymbol{w}} = (\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y}$$

Ο αλγόριθμος εκτελείται για κάθε στοιχηματική εταιρεία ξεχωριστά. Επιπλέον χρησιμοποιείται η τεχνική **10 fold cross validation**.

Για κάθε στοιχηματική εταιρεία παίρνουμε τις 3 αποδόσεις που δίνει για κάθε αγώνα (για νίκη γηπεδούχου ,ισοπαλία και νίκη φιλοξενούμενου).Με βάση αυτά τα χαρακτηριστικά εκπαιδεύουμε τον ταξινομητή. Η κατανομή των χαρακτηριστικών στον τρισδιάστατο χώρο φαίνεται στο παρακάτω γράφημα.



Κάθε διάνυσμα χαρακτηριστικών του αγώνα , αντιστοιχίζεται σε ένα διάνυσμα τριών στοιχείων ανάλογα με το αποτέλεσμα του αγώνα. Έτσι η νίκη της γηπεδούχου ομάδας αντιστοιχίζεται στο διάνυσμα [1,0,0] , η ισοπαλία στο διάνυσμα [0,1,0] και η νίκη της φιλοξενούμενης ομάδας στο διάνυσμα [0,0,1].

Χρησιμοποιούμε τον υλοποιημένο από την **sklearn** βιβλιοθήκη αλγόριθμο **LinearRegression**.

Τα αποτελέσματα από την διαδικασία του training και του testing φαίνεται στην παρακάτω εικόνα.

```
====== RESTART: C:\Users\user\Desktop\pattern recognition\least_squares.py =====
1. Computing for B365 odds and results...
Train score:  0.0859527176036176
Mean squared error: 0.20
Coefficient of determination: 0.08
Successfull predictions mean rate 52.73549655850542 %

2. Computing for BW odds and results...
Train score:  0.08583876111291774
Mean squared error: 0.20
Coefficient of determination: 0.08
Successfull predictions mean rate 52.68682061085029 %

3. Computing for IW odds and results...
Train score:  0.08625270573250253
Mean squared error: 0.19
Coefficient of determination: 0.08
Successfull predictions mean rate 52.45559502664298 %

4. Computing for LB odds and results...
Train score:  0.08342637456858894
Mean squared error: 0.20
Coefficient of determination: 0.08
Successfull predictions mean rate 52.4784894875061 %
```

Όπως βλέπουμε το train score και το coefficient of determination είναι πολύ χαμηλότερα από τον υπολογισμό του μέσου ποσοστού επιτυχίας. Αυτό συμβαίνει επειδή η έξοδος που επιστρέφει ο ταξινομητής δεν είναι ένα διάνυσμα όπου τα στοιχεία του είναι είτε 1 είτε 0 , αλλά πραγματικοί αριθμοί. Για αυτό τον λόγο ερμηνεύουμε το αποτέλεσμα του ταξινομητή με βάση την θέση μέσα στο διάνυσμα όπου βρίσκεται το στοιχείο με τη μεγαλύτερη τιμή.

Αν για παράδειγμα ο ταξινομητής επέστρεφε για ένα διάνυσμα χαρακτηριστικών την τιμή [0.2,-0.3,0.5] εμείς ερμηνεύουμε το αποτέλεσμα ως [0,0,1] ,δηλαδή νίκη της φιλοξενούμενης ομάδας.

Στο training score και στο coefficient of determination δεν κάνουμε αυτή την αντιστοίχιση για αυτό και το score βγαίνει τόσο χαμηλό.

Όπως βλέπουμε το ποσοστό των σωστών προβλέψεων είναι περίπου το ίδιο για όλες τις στοιχηματικές εταιρίες, με μεγαλύτερη ακρίβεια να έχει η B365.

Ερώτημα 3

Το ερώτημα υλοποιείται από το πρόγραμμα **mlp.py** .

Σε αυτό το ερώτημα αντλούμε δεδομένα από τους πίνακες Match και Team_Attributes από το αρχείο database.sqlite. Συγκεκριμένα, για το διάνυσμα χαρακτηριστικών παίρνουμε τις τιμές από τις στήλες **B365H, B365D, B365A, BWH, BWD, BWA, IWH, IWD, IWA, LBH, LBD, LBA** του πίνακα Match και από τις στήλες **buildUpPlaySpeed,buildUpPlayPassing, chanceCreationPassing, chanceCreationCrossing, chanceCreationShooting, defencePressure,defenceAggression, defenceTeamWidth** του πίνακα Team_Atrributes (τα τελευταία τα παίρνουμε 2 φορές, μια για κάθε ομάδα) .Έτσι για κάθε ματς εξάγουμε 28 στο σύνολο χαρακτηριστικά

Το αποτέλεσμα του κάθε αγώνα υπολογίζεται από την διαφορά των γκολ που παίρνουμε από τις στήλες **home_team_goal**, **away_team_goal** του πίνακα Match .

Για αυτό το ερώτημα χρησιμοποιούμε έναν πολυστρωματκό νευρωνικό δίκτυο **(Multilayer Perceptron)**,ώστε ο ταξινομητής να υλοποιεί την συνάρτηση διάκρισης $\boldsymbol{g(\Phi(M)): R^{28} \rightarrow \{H, D, A\}}$ **,** όπου Φ(Μ) το διάνυσμα των 28 χαρακτηριστικών του κάθε αγώνα.

Η συνάρτηση διάκρισης είναι της μορφής $\mathbf{g(x) = w^T x + w_0 = 0}$. Στην δικιά μας περίπτωση τα διανύσματα είναι της μορφής $\boldsymbol{x \in R^{28}, w \in R^{28}, w_0 \in R}$

Ο σκοπός του perceptron είναι να βρει το διάνυσμα χαρακτηριστικών για το οποίο όλα τα διαθέσιμα διανύσματα εκπαίδευσης χαρακτηριστικών ταξινομούνται στην σωστή κλάση.

Το διάνυσμα χαρακτηριστικών υπολογίζεται επαναληπτικά (σε εποχές) .

Επιπλέον χρησιμοποιείται η τεχνική **10 fold cross validation**.

Κάθε διάνυσμα χαρακτηριστικών του αγώνα , αντιστοιχίζεται σε ένα διάνυσμα τριών στοιχείων ανάλογα με το αποτέλεσμα του αγώνα. Έτσι η νίκη της γηπεδούχου ομάδας αντιστοιχίζεται στο διάνυσμα [1,0,0] , η ισοπαλία στο διάνυσμα [0,1,0] και η νίκη της φιλοξενούμενης ομάδας στο διάνυσμα [0,0,1].

Χρησιμοποιούμε τον υλοποιημένο από την **sklearn** βιβλιοθήκη αλγόριθμο **MLPClassifier**.

Τα αποτελέσματα από την διαδικασία του training και του testing φαίνεται στην παρακάτω εικόνα.

```
========== RESTART: C:\Users\user\Desktop\pattern recognition\mlp.py ==
 making Dataset...
 training and testing multilayer perceptron...
1 )Successfull predictions rate 0.2825597115817936
2 )Successfull predictions rate 0.3019378098242452
3 )Successfull predictions rate 0.24064894096439837
4 )Successfull predictions rate 0.27985579089680035
5 )Successfull predictions rate 0.30509238395673727
6 )Successfull predictions rate 0.29652996845425866
7 )Successfull predictions rate 0.34474988733663814
8 )Successfull predictions rate 0.3704371338440739
9 )Successfull predictions rate 0.37990085624155023
10 )Successfull predictions rate 0.3106402164111812
Mean training score: 0.3099185666988791
Successfull predictions mean rate 31.12352699511677 %

>>>
```

Στην παραπάνω εικόνα φαίνονται και τα αποτελέσματα του ταξινομητή για κάθε επανάληψη του 10-fold cross validation.