Εξόρυξη Δεδομένων και Αλγόριθμοι Μάθησης

Εργαστηριακή Άσκηση Εαρινό Εξάμηνο 2023-2024

ΑΜΙΤΣΗΣ ΕΛΕΥΘΕΡΙΟΣ 1072464

ΠΑΠΑΧΡΙΣΤΟΦΙΛΟΥ ΣΑΡΑΝΤΗΣ 1072600

Γλώσσα Υλοποίησης :

Το ide που επιλέξαμε να χρησιμοποιήσουμε για την συγγραφή του κώδικα είναι το Visual Studio Codes. Τα functions των τεχνικών και των αλγορίθμων που χρησιμοποιήθηκαν για την υλοποίηση των ερωτημάτων γράφτηκαν σε python extension (python version 3.10.12) ενώ η συνολική υλοποίηση της άσκησης σε jupyter extension (kernel python 3.10.12)

Έτοιμες Βιβλιοθήκες και σύντομη περιγραφή:

pandas : Βιβλιοθήκη για δημιουργία dataframe. Χρησιμοποιήθηκε για την αποθήκευση και επεξεργασία των δεδομένων.

numpy: Μαθηματική βιβλιοθήκη. Χρησιμοποιήθηκε ύστερα από την διαδικασία δημιουργίας των παραθύρων στο dataset για να γίνει reshape πριν την χρήση του νευρωνικού δικτύου.

matplotlib.pyplot: Βιβλιοθήκη απεικόνισης δεδομένων. Χρησιμοποιήθηκε για την απεικόνιση των δεδομένων μετά από την χρήση τεχνικών που χρησιμοποιήθηκαν.

scipy.stats: Μαθηματική βιβλιοθήκη. Χρησιμοποιήθηκε για την εύρεση του συχνότερου label στην δημιουργία παραθύρων.

sklearn: Βιβλιοθήκη μηχανικής μάθησης. Χρησιμοποιήθηκε για το scaling των δεδομένων, την εύρεση των principal components , την έτοιμη χρήση μετρικών αξιολόγησης και την έτοιμη χρήση του αλγόριθμου random forest. Επιπλέον από την sklearn πήραμε και τις έτοιμες βιβλιοθήκες των αλγορίθμων clustering

seaborn: Βιβλιοθήκη απεικόνισης δεδομένων. Χρησιμοποιήθηκε για δημιουργία box-plot για εύρεση/παρακολούθηση των outliers.

statsmodels.api: Βιβλιοθήκη απεικόνισης δεδομένων. Χρησιμοποιήθηκε για απεικόνιση των κατανομών των δεδομένων καθώς και τον έλεγχο κανονικότητας-qqplot

mpl_toolkits.mplot3d: Βιβλιοθήκη απεικόνισης δεδομένων. Χρησιμοποιήθηκε για τρισδιάστατη απεικόνιση των δεδομένων μετά την διαδικασία του clustering.

pgmpy: Βιβλιοθήκη έτοιμης υλοποίησης και βοηθητικών τεχνικών για Bayesian Network. Χρησιμοποιήθηκε για την εύρεση κατάλληλου structure και δημιουργίας του δικτύου.

Tensorflow: Βιβλιοθήκη τεχνητής νοημοσύνης (machine learning and deep learning). Χρησιμοποιήθηκε για την δημιουργία νευρωνικού δικτύου.

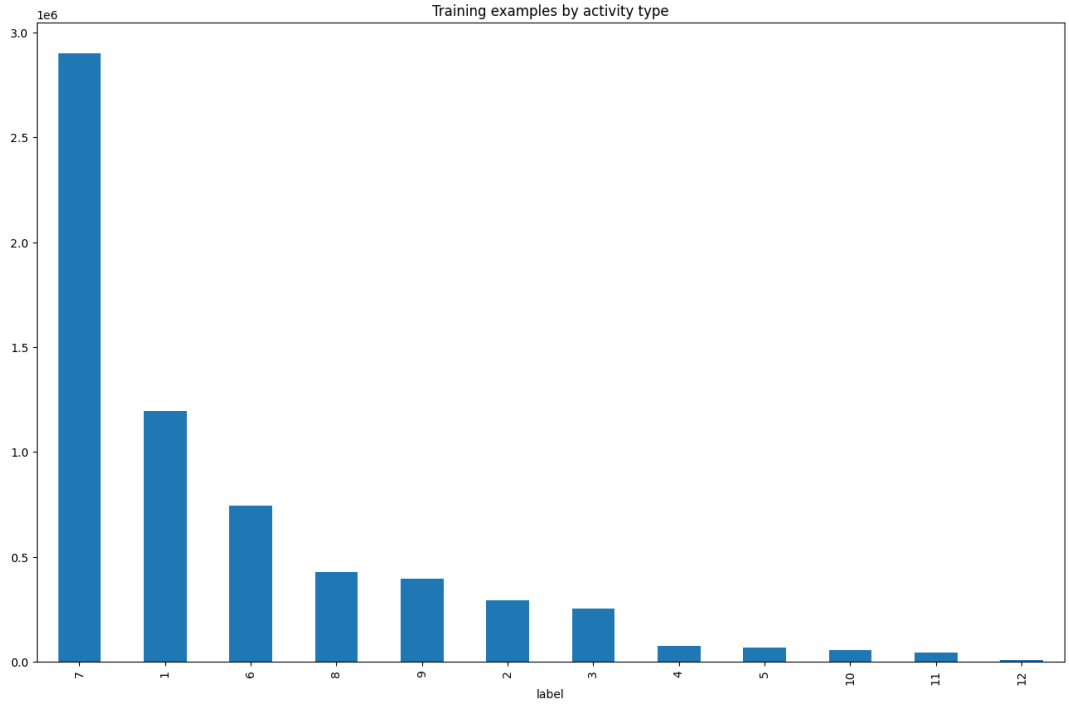
Ερώτημα 1 - Data analysis and preprocessing

1. Το πρώτο βήμα για την κατανόηση του dataset που είχαμε στην κατοχή μας ήταν να κάνουμε μια γρήγορη ανάγνωση στα csv. Εκεί παρατηρήσαμε ότι τα αρχεία που είχαμε είχαν διαφορές στον αριθμό των στηλών, δηλαδή κάποια είχαν παραπάνω στήλες οι οποίες δεν είχαν κάποια χρήση για τις επόμενες διαδικασίες αλλά και λάθη στα timestamps των δειγμάτων. Έτσι για την ευκολότερη και γρηγορότερη ανάλυση και επεξεργασία των δεδομένων ενώσαμε όλα τα csv σε ένα dataframe και δώσαμε στα δεδομένα όλων των αρχείων ίδια μορφή. Επιπλέον για όλες τις κλάσεις αντικαταστήσαμε το όνομα τους με αριθμούς από το 1 εώς το 12 όπως φαίνονται παρακάτω.

   1: walking → 1
   2: running → 2
   3: shuffling → 3
   4: stairs (ascending) → 4
   5: stairs (descending) → 5
   6: standing → 6
   7: sitting → 7
   8: lying → 8
   13: cycling (sit) → 9

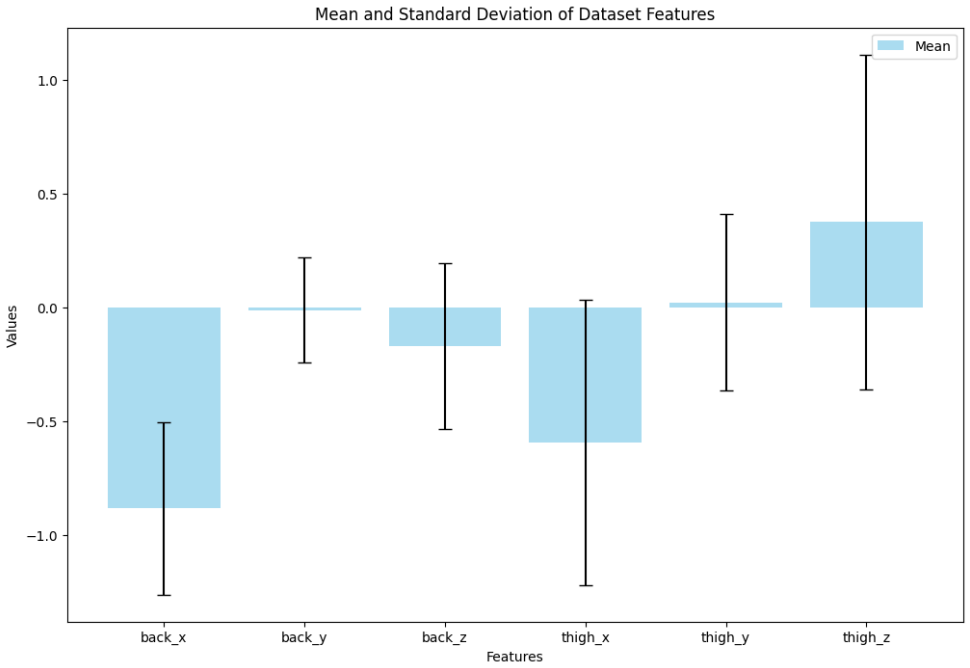
14: cycling (stand) →10
130: cycling (sit, inactive) → 11
140: cycling (stand, inactive) → 12

2. Το δεύτερο βήμα ήταν να δούμε περίπου τον όγκο των δεδομένων που είχαμε και σε ποιες κλάσεις ανήκουν. Όπως φαίνεται και παρακάτω ο όγκος δεδομένων σε κάθε κλάση διαφέρει κατά πολύ. Έτσι καταλήξαμε στο συμπέρασμα ότι ενδεχομένως να χρειαστεί αργότερα να δημιουργήσουμε ένα πιο ισορροπημένο dataset όπου καμία κλάση δεν θα υπερκαλύπτεται από τον μεγάλο όγκο δεδομένων άλλων κλάσεων.

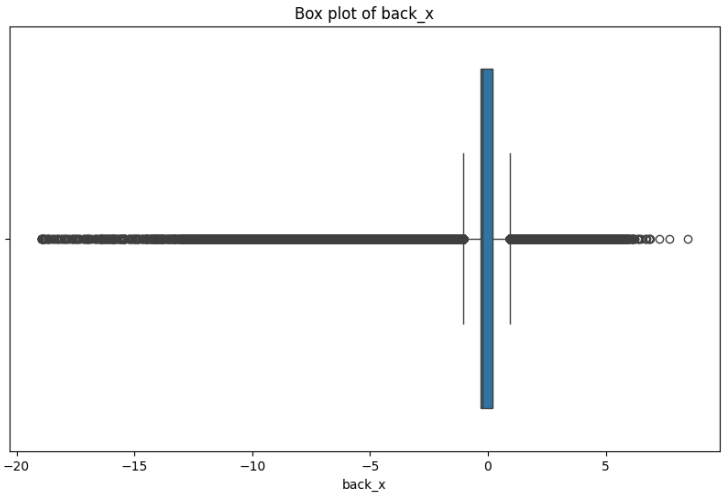


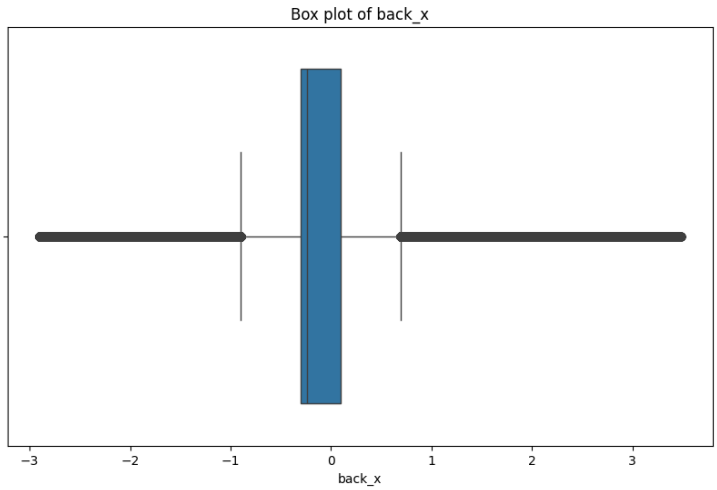
3. Το τρίτο βήμα ήταν να κοιτάξουμε τον μέσο όρο (mean) και την τυπική απόκλιση (standard deviation) για τα 6 features , τις μετρήσεις των 6 αισθητήρων, για να αποκτήσουμε μια πρώτη εικόνα. Αυτό που παρατηρούμε τόσο από την τυπική απόκλιση όσο και από τους μέσους ότι τα features διαφέρουν σημαντικά στο impact που θα έχουν στα μοντέλα προβλέψεων. Αναλυτικότερα είδαμε πως τα χαρακτηριστικά με μεγαλύτερο standard deviation ενδεχομένως να χρειαστούν outliers detection. Επιπλέον βάση των μέσων όρων των χαρακτηριστικών συμπεραίνουμε προς τα που τείνει η κατανομή των δεδομένων (left/right skewed) καθώς και πως τα

χαρακτηριστικά με μεγαλύτερη τυπική απόκλιση και μεγάλο μέσο όρο (το μέτρο του) έχουν μεγαλύτερο impact στα μοντέλα. Όλα αυτά μας οδήγησαν στα εξής συμπεράσματα. Το πρώτο είναι ότι θα πρέπει να κάνουμε scaling στα δεδομένα. Η τεχνική που επιλέξαμε είναι το Standard scaling για να φέρουμε όλα τα features στην ίδια κλίμακα και να ισορροπήσουμε την επιρροή που θα έχουν. Επίσης τα δεδομένα κινήσεων από την φύση τους είναι δεδομένο πως θα έχουν outliers και το standard scaling είναι αρκετά ανεκτικό σε αυτά. Το δεύτερο συμπέρασμα ήταν ότι θα πρέπει να ελεγχτούν τα δεδομένα για outliers.



4. Το τέταρτο βήμα ήταν λοιπόν να κάνουμε standard scaling στα δεδομένα μας και να δημιουργήσουμε ένα box plot για να ελέγξουμε για outliers. Το box plot θεωρεί ως outliers ως outliers τις τιμές που βρίσκονται κάτω από το πρώτο quartile μέχρι και κάποιο σημείο (Q1 - 1.5 * IQR) καθώς και αυτά που βρίσκονται πάνω από το τρίτο quartile (Q3 + 1.5 *IQR) . Εμείς προκειμένου να μην αλλοιώσουμε ενδεχομένως χαρακτηριστικά του κάθε feature αποφασίσαμε να κόψουμε μόνο μέχρι το σημείο που φαίνονται αισθητά το πόσο αραιά βρίσκονται εκεί τα datapoints.





5. Το πέμπτο βήμα ήταν να κάνουμε κάποιες τελικές απεικονίσεις. Αρχικά απεικονίσαμε τις κατανομές των δεδομένων, ύστερα ελέγξαμε για κανονικές κατανομές με qq-plot (τυπικά κυρίως μήπως μας ξέφυγε κάτι αφού ήδη γνωρίζαμε πως δεν είναι κανονικές οι κατανομές) και τέλος δημιουργήσαμε ένα correlation matrix για να δούμε αν υπάρχουν σχέσεις που να συνδέουν τας χαρακτηριστικά του προβλήματος. Γενικά οι σχέσεις μεταξύ των χαρακτηριστικών ήταν πολύ αδύναμες χωρίς να συσχετίζονται σχεδόν καθόλου με εξαίρεση τα χαρακτηριστικά thigh_x – back_x και thigh_x – thigh_z που παρουσίασαν μια μικρή συσχέτιση της τάξης του ~0.45

<u>Έρώτημα 2 – Classification</u>

1. Convolutional 2D Neural Network:

   - Data Preproccesing: Αρχικά για να δώσουμε τα δεδομένα μας στο νευρωνικό δίκτυο πρέπει πάρουν συγκεκριμένη μορφή. Η μορφή που πρέπει να πάρουν είναι ένα array από παράθυρα. Το παράθυρο είναι στην πραγματικότητα ένα τρισδιάστατο array με shape(σχήμα)→ (a,b,c). Το a εκπροσωπεί των αριθμό των δεδομένων μας, στην δική μας περίπτωση πρόκειται για 200 γραμμές του ήδη υπάρχων dataframe οι οποίες αντιστοιχούν σε 4 δευτερόλεπτα. Το b εκπροσωπεί των αριθμό των features, στην δική μας περίπτωση 6 και τέλος το c που εκπροσωπεί τα χρώματα/channels, στην δική μας περίπτωση το 1. Κάθε ένα από αυτά τα

παράθυρα ίδιου μεγέθους πρέπει να αντιστοιχεί σε μία μόνο κλάση του προβλήματος. Για να το πετύχουμε αυτό για κάθε 200 γραμμές των δεδομένων μας δημιουργούμε ένα παράθυρο δίνοντάς του ως label την πιο συχνά εμφανιζόμενη κλάση και για να σιγουρευτούμε ότι δεν θα υπάρξει αλλοίωση των δεδομένων χρησιμοποιούμε και ένα overlap 50 σειρών.

- Conv2D Neural Network construction: Για την κατασκευή του νευρωνικού ξεκινήσαμε με ένα conv2d layer με 32 kernels μεγέθους 2x2 και συνάρτηση ενεργοποίησης relu. Επιλέξαμε έναν μικρό αριθμό kernels για να μάθει κάποια πιο βασικά χαρακτηριστικά για αρχή. Το μικρό σχετικά μέγεθος των φίλτρων(2x2) επιλέχτηκε για να γίνει αναλυτικότερος εντοπισμός χρήσιμων πληροφοριών. Έπειτα λόγω του άνισου αριθμού data κάθε κλάσης προσθέσαμε dropout layers ώστε να αγνοηθούν τυχαία κάποιοι νευρώνες και να αποφευχθεί πιθανό overfitting. Έπειτα κάνουμε το ίδιο με περισσότερους kernels για να αναζητήσουμε περισσότερα και λιγότερο εμφανή χαρακτηριστικά. Έπειτα κάνουμε flatten το δυσδιάστατο matrix που πήραμε από τα 2dconv layers και το κάνουμε μονοδιάστατο για να το βάλουμε σε ένα πλήρως συνδεδεμένο, με το προηγούμενο, layer και τέλος εξάγουμε τα αποτελέσματα μας από το output layer με συνάρτηση ενεργοποίησης softmax.

2. Random Forest Algorithm:

- Data Preproccesing: Λαμβάνοντας υπόψιν την ευαισθησία των αλγορίθμων δέντρων στο overfitting επιλέξαμε να δημιουργήσουμε ένα πιο ισορροπημένο αριθμό από data για κάθε κλάση. Τα δεδομένα δεν χρειάστηκαν περαιτέρω επεξεργασία.
- Random Forest Structure: Οι υπερπαράμετροι που θέσαμε ήταν να δημιουργηθούν τα δέντρα με κριτήριο την εντροπία και τα ελάχιστα data points για να δημιουργηθεί ένα φύλλο να είναι πενήντα δείγματα. Επιπλέον βάλαμε 200 estimators δηλαδή να δημιουργηθούν 200 δέντρα πριν γίνει το majority voting.

3. Bayesian Network:

   - Data Preproccesing: Για το Bayesian Network χρειάστηκε να διακρητοποιήσουμε τις τιμές των 6 features σε very low , low, avg, high και very high βάση της τυπικής τους απόκλισης. Ο συγκεκριμένος αλγόριθμος δεν χρειάστηκε περεταίρω αλλαγές στα δεδομένα.
   - Bayesian Network structure: Λόγω των όχι τόσο καλών σχέσεων μεταξύ των features αποφασίσαμε να αφήσουμε την εύρεση του καλύτερου πιθανού δικτύου στον αλγόριθμο hill climb με μέγιστο αριθμό γονέων για κάθε κόμβο να είναι δύο και ελάχιστη βελτίωση το 0.001 .

4. Σύγκριση – Αξιολόγηση Μοντέλων:

   Τα μοντέλα που υλοποιήθηκαν αξιολογήσαμε με τις μετρικές accuracy και f1 score.

   Conv2D Neural Network: Το καλύτερο με διαφορά μοντέλο. Ταίριαζε απόλυτα στις ανάγκες του προβλήματος , αφού εκπαιδεύτηκε βλέποντας ολόκληρα μέρη της χρονοσειράς . Έδωσε το μεγαλύτερο accucary που άγγιζε το 97% σε όλα τα test sets.

   Bayesian Network: Είχε το αμέσως καλύτερο accuracy με 81% . Ήταν ο λιγότερο ταιριαστός αλγόριθμος αφού έπρεπε και να διακρητοποιηθούν οι τιμές αλλά και να φροντίσουμε να δώσουμε τέτοιες υπερπαραμέτρους ώστε να λειτουργήσει το μοντέλο. Λόγω των πολύ μικρών σχέσεων μεταξύ των 6 features πολλές φορές δημιουργόντουσαν δίκτυα τόσο περίπλοκα που δεν μπορούσαν να υλοποιηθούν. Είχε αρκετά καλό αρμονικό μέσο, 78% που είναι χειρότερος όμως από αυτόν του random forest.

   Random Forest: Είχε το χειρότερο accuracy με 80% αλλά είχει αρμονικό μέσο 80% που ήταν καλύτερο από του Bayesian Network . Με μικρές διαφορές στην αξιοπιστία των αποτελεσμάτων του από τον Bayesian Network αλλά είχε με διαφορά τον χειρότερο χρόνο υλοποίησης. Αφήνοντας εκτός αρκετά μεγάλο

όγκο δεδομένων προκειμένου να αποφύγουμε το overfitting είχαμε χρόνους που άγγιζαν και τα 6 λεπτά.

Έρώτημα 3 – Clustering

Για την καλύτερη ταξινόμηση των δειγμάτων δημιουργήσαμε ένα πιο ισορροπημένο dataset με 10.000 δείγματα ανά κλάση. Χρησιμοποιήσαμε τρεις αλγόριθμους διαφορετικής φύσης τον Mini Batch K-means, τον DBSCAN και τον Birch.

1. Mini Batch K-means Hyperparameters: Αρχικά χρησιμοποιήσαμε kmeans++ για το αρχικό spread των clusters. Έπειτα θέσαμε ως τον μέγιστο αριθμό που θα τρέξει το κάθε batch στο 20 και θα περιέχει 512 data points. Τα centroids θα ανανεωθούν 10 φορές και αν σε 10 mini batches δεν έχει βελτίωση ο αλγόριθμος σταματάει. Ο αριθμός των cluster καθορίστηκε από το silhouette score που το μεγαλύτερο έδωσε για 2 clusters στα 0.38 που είναι αποδεκτό.

2. DBSCAN Hyperparameters: Στον DBSCAN δώσαμε στο min samples για να δημιουργηθεί core point 50 δείγματα και η απόσταση για να θεωρηθούν γείτονες δύο data points 1.

3. Birch Hyperparameters: Για Threshold δώσαμε 0.01 ενώ για branching factor 50. Επιπλέον πάλι με κριτήριο την μετρική silhouette καλύτερο διαχωρισμό μεταξύ clusters είχαμε για 2 clusters.

Πείραμα-Συμπεράσματα:

Πέραν του DBSCAN για τους ιεραρχικούς αλγόριθμους Mini Batch K-means και Birch δοκιμάσαμε να δημιουργήσουμε 12 clusters και να δούμε πόσο καλό είναι το purity , δηλαδή κατά πόσο διαχωρίζονται οι κλάσεις σε μοναδικούς clusters. Και στις δύο περιπτώσεις τα αποτελέσματα δεν ήταν καθόλου ικανοποιητικά αφού έδιναν purity από 30-45% (μη αποδεκτό). Έπειτα αποφασίσουμε να προσεγγίσουμε διαφορετικά το πρόβλημα και να ομαδοποιήσουμε τα δεδομένα όσο το δυνατόν καλύτερα, μεγιστοποιώντας δηλαδή τη μετρική silhouette. Το

silhouette έδινε αποδεκτό ποσοστό περίπου 40% για δύο clusters. Στον πρώτο cluster μπήκαν όλες οι κινήσεις των κλάσεων από 1 εώς και 7 που είναι κινήσεις όπου έχουμε αρκετή κίνηση τόσο από τους αισθητήρες των μηρών όσο και της πλάτης. Στον δεύτερο cluster είναι οι κινήσεις 8 που είναι η κλάση από τους ξαπλωμένους συμμετέχοντες καθώς και οι κλάσεις από 9 εώς και 12 που είναι στο ποδήλατο. Οι κλάσεις από 9 εώς 12 είναι μοιρασμένες στους 2 clusters, συμπεράναμε λοιπόν ότι αυτό οφείλεται στην ακινησία του κορμού στις μετρήσεις αυτές και ενδεχομένως στις μετρήσεις όπου ο ποδηλάτης δεν κάνει πετάλι. Να σημειωθεί ότι έγινε απόπειρα να συσταδοποιήσουμε τα δεδομένα σε 12 clusters,όσες και οι κλάσεις, που παρόλα αυτά δεν έδινε ικανοποιητικά αποτελέσματα ούτε και στην silhouette.