Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

ΑΝΑΛΥΣΗ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΧΩΡΟΧΡΟΝΙΚΩΝ ΔΕΔΟΜΕΝΩΝ

ΥΔΑ

Project FrailSafe

Πετράκης Κωνσταντίνος

Α.Μ. 1041589

2021

Όλα τα ζητούμενα υλοποιήθηκαν στη γλώσσα προγραμματισμού Python και με χρήση του Jupyter Notebook. Το notebook με τον κώδικα βρίσκεται στον παρακάτω σύνδεσμο:

https://drive.google.com/drive/folders/1ADzbuqYrpO2Tb7Htsco75NdVZ-DgB8oX?usp=sharing

Part A

1. Preprocessing

Convert nominal to numerical

Για την μετατροπή των nominal χαρακτηριστικών σε αριθμητικά χρησιμοποίησα όσες τιμές ήταν αναγκαίες ξεκινώντας πάντα από το 0, με την λογική το 0 να αντιστοιχεί στην 'χειρότερη' τιμή του χαρακτηριστικού και οι αυξανόμενες τιμές σε όλο και καλύτερες τιμές για το εκάστοτε χαρακτηριστικό.

Remove erroneous values

Αντικαθιστώ με nan όλες τιμές οι οποίες είναι πάνω από την τιμή 900. Αυτό γιατί στην στήλη bmi_score, για παράδειγμα, υπάρχει μια τιμή 921. Επίσης καθώς δεν είχα πληροφορία για τη μονάδα χρόνου στην οποία αντιστοιχούν οι τιμές της στήλης social_phone θεωρώ και εδώ τις τιμές >900 σαν εσφαλμένες. Σημειώνω πως το συγκεκριμένο χαρακτηριστικό έχει μέση τιμή 213 και τυπική απόκλιση 279, είναι επομένως δύσκολο να είμαστε σίγουροι πως αυτές οι τιμές είναι όντως εσφαλμένες, εδώ ενεργήσαμε με τη θεώρηση πως είναι.

Handle Missing values

Έπειτα και από το προηγούμενο βήμα αντικαθιστώ όλες τις ελλιπείς τιμές με το μέσο όρο των χαρακτηριστικών. Θεωρώ αυτήν την καλύτερη επιλογή διότι, πλην ελάχιστων στηλών (π.χ. bmi_body_fat και lean_body_mass), το πλήθος των ελλιπών τιμών στις υπόλοιπες είναι σχετικά μικρό. Δοκιμάστηκε και η χρήση του IterativeImputer που παρέχει το sklearn για την συμπλήρωση ελλιπών τιμών αλλά δεν είχε κάποια ιδιαίτερη επίπτωση στους παρακάτω αλγόριθμους κατηγοριοποίησης και δεν εξετάστηκε η χρήση του περαιτέρω.

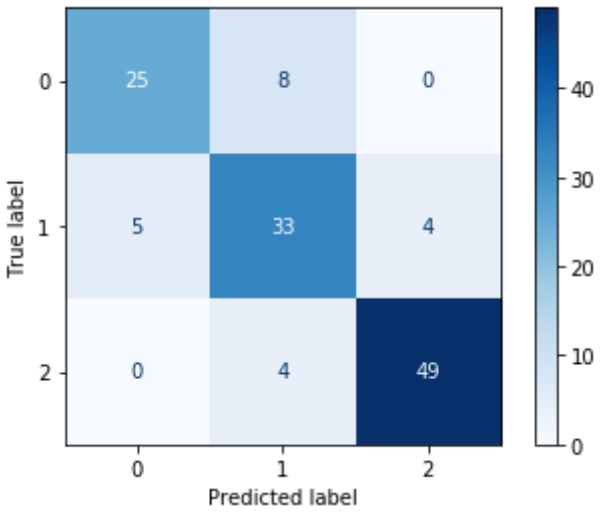
2. Classification

Αφού αφαιρέσω από το dataframe τις 5 παραμέτρους που χρησιμοποιήθηκαν για την παραγωγή της μεταβλητής fried, αφαιρώ και την στήλη της μεταβλητής fried καθώς και την στήλη part_id. Πρώτα από όλα παρατηρώ πως το πλήθος των δειγμάτων για κάθε κλάση δεν είναι ισορροπημένο. Έχουμε σχεδόν το ίδιο πλήθος δειγμάτων, 227 και 213 για τις κλάσεις 'Pre-frail' και 'Non-frail' αντίστοιχα, ενώ το πλήθος των δειγμάτων για την κλάση 'Frail' είναι μόλις 100. Για να διορθωθεί αυτό δημιουργώ αντίγραφα των δειγμάτων της κλάση 'Frail' κλάσης και τα προσθέτω στο dataframe. Με αυτόν τον τρόπο έχω πλέον 200 (διπλότυπα) δείγματα από την κλάση 'Frail'.

Αρχικά διαχωρίζω το σύνολο δεδομένων σε υποσύνολα 80-20 για εκπαίδευση και έλεγχο αντίστοιχα. Στην συνέχεια, σαν ένα βήμα προ-επεξεργασίας, τυποποιώ τα χαρακτηριστικά αφαιρώντας από το κάθε ένα την μέση τιμή και διαιρώντας με την τυπική του απόκλιση με την χρήση του StandardScaler. Στους παρακάτω αλγόριθμους χρησιμοποιούνται αυτά τα τυποποιημένα δεδομένα. Για την κατηγοριοποίηση χρησιμοποιώ RandomForest

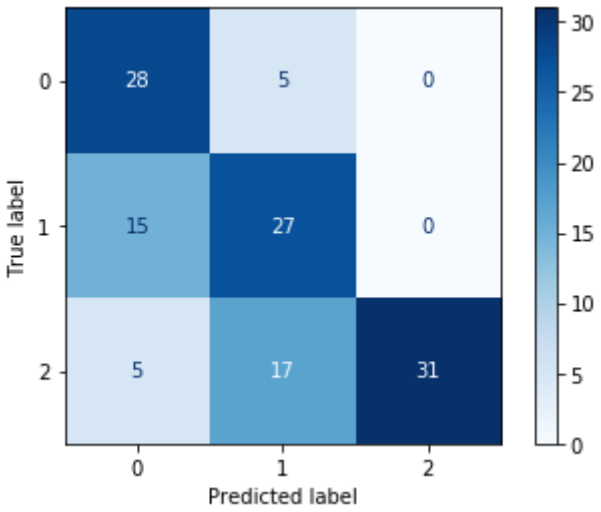
κατηγοριοποιητές, τις μηχανές διανυσμάτων υποστήριξης (SVM) και κατηγοριοποίηση K-κοντινότερων γειτόνων (K-NN). Γνωρίζω πως η μετρική της ακρίβειας δεν αποτελεί την πιο ενδεικτική μετρική σε προβλήματα όπου παρατηρείται ανισοκατανομή μεταξύ των κλάσεων γι' αυτό και μαζί με την ακρίβεια παραθέτω και μια μετρική ακρίβειας ζυγισμένη από το αντίστοιχο πλήθος δειγμάτων κάθε κλάσης, υπό το όνομα balanced_accuracy, ώστε να έχω μία ένδειξη και για το κατά πόσο η ανισοκατανομή επηρεάζει τα αποτελέσματα. Εκτός αυτού σε όλες τις περιπτώσεις παραθέτω τις μετρικές Precision και Recall καθώς και τον πίνακα σύγχυσης για την καλύτερη ερμηνεία των αποτελεσμάτων. Για τα παρακάτω η αντιστοίχιση των κλάσεων είναι 0: No-Frail ,1: Pre-Frail ,2: Frail. Τέλος και για τους 3 παρακάτω κατηγοριοποιητές το σύνολο ελέγχου στο οποίο δοκιμάστηκαν αποτελείται από 33 δείγματα της κλάσης Non-frail, 42 δείγματα της κλάσης Pre-frail και 53 δείγματα της κλάσης Frail.

Όσον αφορά τα RandomForests η καλύτερη απόδοση παρατηρήθηκε για πλήθος DecisionTrees ίσο με 12. Σημειώνω πως επειδή έχω θέσει την παράμετρο bootstrap=False αυτό σημαίνει πως χρησιμοποιούνται όλα τα δείγματα του συνόλου εκπαίδευσης για την κατασκευή κάθε DecisionTree. Σε κάθε εκτέλεση η απόδοση του αλγορίθμου διαφέρει και οι τιμές που παρουσιάζονται αντιστοιχούν σε μία από τις καλύτερες εκτελέσεις των RandomForests. Στο παραπάνω σύνολο ελέγχου έλαβα Accuracy=83.59%, Precision= 82.26% και Recall =82.26%.



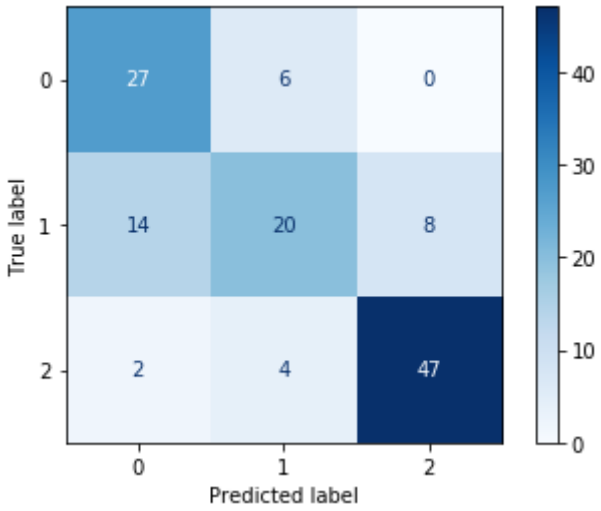
Από τον confusion matrix μπορούμε να δούμε πως τα RandomForests τα πάνε αρκετά καλά κατηγοριοποιώντας μόλις 8 δείγματα της κλάσης 'No-frail' λανθασμένα στην κλάση Pre-frail, 4 δείγματα της κλάσης Pre-frail κατηγοριοποιούνται λανθασμένα στην κλάση Frail και 5 στην κλάση No-frail, ενώ για την κλάση Frail μόλις 4 δείγματα κατηγοριοποιούνται λανθασμένα στην Pre-frail. Επίσης δεν φαίνεται να επηρεάζει την απόδοση του μοντέλου το γεγονός πως στο σύνολο ελέγχου υπάρχει κάποια ανισοκατανομή μεταξύ των 3 κλάσεων, όπως φαίνεται και από την μετρική balanced_accuracy, η οποία προσμετρά την ανισοκατανομή μεταξύ των κλάσεων, και εδώ έχει τιμή 82.26%, μόλις 1% μικρότερη από την κλασική μετρική της ακρίβειας. Θεωρείται θεμιτό το γεγονός πως για τις κλάσεις Frail και No-frail όσα δείγματα κατηγοριοποιούνται λάθος καταλήγουν στην 'ενδιάμεση' κλάση Pre-frail. Αυτό παρέχει κάποια ένδειξη πως η κατηγοριοποίηση γίνεται μάλλον με σωστά κριτήρια καθώς ακόμα και στην περίπτωση του λάθους αυτό δεν είναι μεγάλο, υπό την έννοια ότι έστω και με διαισθητικό τρόπο δεν απέχουμε πολύ από την επιθυμητή κλάση.

Τα SVM χρησιμοποιήθηκαν με πολυωνυμικό kernel βαθμού 3. Στο ίδιο σύνολο ελέγχου λάβαμε Accuracy: 67.18%, Precision: 71.14% και Recall: 69.20%.



Από τον confusion matrix παρατηρούμε πως τα αποτελέσματα είναι σαφώς χειρότερα σε σχέση με την χρήση των RandomForests καθώς τώρα ναι μεν έχουμε μόλις 5 δείγματα της κλάσης No-Frail να κατηγοριοποιούνται λανθασμένα στην Pre-frail αλλά για τα δείγματα της κλάσης Pre-frail έχουμε 15 που κατηγοριοποιούνται στην κλάση Non-frail και για την κλάση Frail 5 δείγματα κατηγοριοποιούνται στην No-frail και 17 στην Pre-frail. Και πάλι, σε μικρότερο βαθμό ωστόσο εξαιτίας του μεγάλου πλήθους των λάθος ταξινομημένων δειγμάτων, θα μπορούσαμε να πούμε πως μάλλον βρισκόμαστε σε σωστό δρόμο αφού λόγου χάρη για την κλάση Frail από τα λάθος ταξινομημένα δείγματα τα περισσότερα ταξινομούνται στην ‘γειτονική’ κλάση Pre-frail από ότι στην No-frail (το αντίθετο θα ήταν περισσότερο ανησυχητικό).

Για τον K-NN κατηγοριοποιητή χρησιμοποίησα παράμετρο K=10, δηλαδή λαμβάνεται απόφαση για την κλάση ενός αντικειμένου με βάση την κλάση πλειοψηφίας των 10 κοντινότερων γειτόνων του, ενώ η συνεισφορά του κάθε δείγματος στη γειτονιά ενός αντικειμένου ζυγίζεται αντιστρόφως ανάλογα από την απόσταση του από το αντικείμενο. Και πάλι στο ίδιο σύνολο ελέγχου λαμβάνουμε Accuracy:73.43%, Precision:71.63% Recall:72.70%. Και εδώ η ανισοκατανομή των 3 κλάσεων δεν φαίνεται να επηρεάζει παρά ελάχιστα την ακρίβεια. Πιο συγκεκριμένα φαίνεται να υπερεκτιμάται μόλις 1% αφού η κατάλληλα ζυγισμένη ακρίβεια είναι ίση με 72.7%.



Από τον confusion matrix τώρα 6 δείγματα της κλάσης No-frail κατηγοριοποιούνται λανθασμένα στην Pre-frail, 14 δείγματα της κλάσης Pre-frail κατηγοριοποιούνται λανθασμένα στην κλάση No-frail και άλλα 8 στην κλάση Frail, ενώ 4 δείγματα της κλάσης Frail κατηγοριοποιούνται λανθασμένα στην κλάση Pre-frail και 2 στην κλάση No-frail.

Στην συνέχεια ακολουθεί ένας πίνακας με τα συγκεντρωτικά αποτελέσματα.

Κατηγοριοποιητής	SVM	K-NN	RandomForests
Accuracy	67.18%	73.43%	**83.59%**
Balanced_Accuracy	69.20%	72.70%	**82.26%**
Precision	71.14%	71.63%	**83.03%**
Recall	69.20%	72.70%	**82.26%**

Συγκεντρωτικός Πίνακας Αποτελεσμάτων

Όσον αφορά την σύγκριση μεταξύ των 3 αλγορίθμων τα SVM είχαν τα χειρότερα αποτελέσματα, με τους K-NN κατηγοριοποιητές να τα πηγαίνουν σαφώς καλύτερα σε σχέση με τα SVM, περίπου 6% όσον αφορά την ακρίβεια, αλλά να έχουν παρόμοια απόδοση όσον αφορά Precision και Recall. Τα RandomForests είχαν την καλύτερη επίδοση στο σύνολο των μετρικών οδηγώντας σε βελτιωμένα αποτελέσματα της τάξης του 10% όσον αφορά την ακρίβεια σε σχέση με τους K-NN, και 16% σε σχέση με τα SVM. Η βελτίωση σε σχέση και με τις υπόλοιπες μετρικές είναι ανάλογη του 10%. Το γεγονός πως τα RandomForests έχουν την καλύτερη απόδοση ίσως οφείλεται σε αυτό που γνωρίζουμε ήδη από την θεωρία, και επιβεβαιώνεται εμπειρικά για άλλη μία φορά, πως ο συνδυασμός κατηγοριοποιητών (ensembling) μπορεί να επιφέρει βελτιωμένα αποτελέσματα.

<u>Part B</u>

1. Preprocessing of the beacons dataset

Correct room labels

Επιλέγω μία ενδεικτική λέξη για κάθε δωμάτιο και αντικαθιστώ κάθε τιμή πού αναφέρεται σε αυτό το δωμάτιο με κάποιο ορθογραφικό λάθος με την επιλεγμένη λέξη. Μετά από αυτή την διαδικασία απομένει περίπου 5% του συνόλου beacons με τιμές nan.

Remove erroneous users

Βρίσκω όλες τις τιμές της στήλης part_id που δεν αντιστοιχούν σε 4-ψήφιο αριθμό και τις αφαιρώ. Μένουν 46782 δείγματα από το αρχικό σύνολο δεδομένων beacons

Generate features

Αρχικά κατασκευάζω το user_dict το οποίο για κάθε χρήστη κρατά τις ξεχωριστές ημερομηνίες που δίνονται. Στην συνέχεια τροποποιώ αυτές τις ημερομηνίες ώστε να είναι σε μορφή YY-MM-DD για να είμαι σε θέση να εντοπίσω τις μετακινήσεις του χρήστη μέσα σε μια μέρα. Στην συνέχεια διατρέχω αυτό το νέο date_formated_user_dict και για κάθε χρήστη και κάθε ξεχωριστή ημερομηνία γι' αυτόν τον χρήστη κατασκευάζω ένα νέο προσωρινό dataframe με όλες τις μετρήσεις που αντιστοιχούν σε αυτόν τον χρήστη για την συγκεκριμένη ημερομηνία. Για να βρω χρονικές διαφορές μετακίνησης από δωμάτιο σε δωμάτιο αρχικά κατασκευάζω datetime αντικείμενα για την επιλεγμένη μέρα και για κάθε ώρα που υπάρχει μέτρηση γι' αυτόν τον χρήστη μέσα στην ημέρα. Θεωρώ ότι η μετακίνηση από δωμάτιο σε δωμάτιο διαρκεί 5 δευτερόλεπτα και σαν το χρόνο που πέρασε το άτομο στο δωμάτιο ορίζω την επόμενη χρονική στιγμή που έχω μέτρηση πλην την τρέχουσα χρονική στιγμή πλην 5 που χρειάστηκε για την μετακίνηση. (end_time-begin_time-5 ο χρόνος που πέρασε το άτομο στο δωμάτιο με timestamp begin_time). Δηλαδή ο χρόνος που πέρασε ένα άτομο σε ένα δωμάτιο είναι ίσος με την χρονική διαφορά των χρονικών στιγμών μείον 5 δευτερόλεπτα, που θεωρείται ότι διαρκεί η μετακίνηση. Έπειτα αθροίζοντας τους χρόνους για τα δωμάτια που μας ενδιαφέρουν προκύπτει ο συνολικός χρόνος που πέρασε ο εκάστοτε χρήστης στο κάθε δωμάτιο. Στην συνέχεια υπολογίζω τον συνολικό χρόνο για τον οποίο έχω δεδομένα για ένα άτομο μέσα σε μία ημέρα αφαιρώντας το αρχικό timestamp της ημέρας από το τελευταίο timestamp της ημέρας. Αθροίζοντας αυτούς τους χρόνους για κάθε μέρα παίρνω τον συνολικό χρόνο που είχαμε διαθέσιμο για κάθε χρήστη. Τέλος αποθηκεύω στο person_percentage_per_room_dict το ποσοστό που πέρασε ο εκάστοτε χρήστης σε κάθε δωμάτιο διαιρώντας τον συνολικό χρόνο που πέρασε ο χρήστης στο δωμάτιο όπως υπολογίστηκε παραπάνω, με τον συνολικό χρόνο κάθε χρήστη. Τελικά το person_percentage_per_room_dict αντιστοιχεί σε ένα dictionary όπου για κάθε χρήστη έχω ένα άλλο dictionary το οποίο περιέχει το ποσοστό του χρόνου του χρήστη σε κάθε ένα από τα 4 δωμάτια.

2. Merging the two preprocessed datasets

Ορίζω ένα νέο beacons σύνολο δεδομένων το οποίο αποτελείται από μία στήλη part_id και 4 στήλες μία για κάθε δωμάτιο που μας ενδιαφέρει [Bedroom, Bathroom, Livingroom, Kitchen], οι οποίες περιέχουν το ποσοστό χρόνου του χρήστη στο αντίστοιχο δωμάτιο. Εν συνεχεία μετατρέπω το part_id σε τύπο int64 και εκτελώ την συνένωση με το clinical σύνολο δεδομένων ως προς τη στήλη part_id. Σημειώνω πως σε αυτό το συγχωνευμένο σύνολο δεδομένων δεν υπάρχουν οι 5 στήλες που αφαιρέθηκαν στην κατηγοριοποίηση και η παράμετρος fried.

3. Clustering

Σαν βήμα προ-επεξεργασίας πριν την ομαδοποίηση εκτελώ ομαλοποίηση των δεδομένων με την χρήση του Normalizer() που παρέχει η βιβλιοθήκη sklearn. Η χρήση του Normalizer() έχει σαν αποτέλεσμα κάθε δείγμα να κανονικοποιείται ώστε να έχει νόρμα ίση με 1 (δηλ. κάθε σειρά του συνόλου δεδομένων θα έχει νόρμα 1). Η εκτέλεση αυτού του βήματος προ-επεξεργασίας είναι απαραίτητη όταν υπάρχει ανάγκη να ποσοτικοποιηθεί η ομοιότητα μεταξύ

δειγμάτων. Η ομαδοποίηση γίνεται με βάση κάποιο μέτρο ομοιότητας μεταξύ των αντικείμενων επομένως και εδώ είναι απαραίτητη η ομαλοποίηση. Χρησιμοποιώ τον αλγόριθμο kmeans, έναν bottom-up ιεραρχικό αλγόριθμο (AgglomerativeClustering) συνενώνοντας κάθε φορά δυο ομάδες με βάση την μικρότερη απόσταση μεταξύ των δειγμάτων τους (single link) και δυο παραλλαγές φασματικής ομαδοποίησης (SpectralClustering). Η ομοιότητα μεταξύ των αντικειμένων στην φασματική ομαδοποίηση υπολογίζεται με βάση το μητρώο γειτνίασης ενός γράφου που κατασκευάζεται από τα δεδομένα, ή γενικότερα με βάση ένα μητρώο που περιέχει τις αποστάσεις μεταξύ των δειγμάτων. Εδώ χρησιμοποιώ 2 τρόπους εφαρμογής της μεθόδου. Στον πρώτο κατασκευάζω ένα μητρώο αποστάσεων μεταξύ των δειγμάτων με την χρήση Laplacian_kernel, οι οποίοι αποτελούν μια παραλλαγή των rbf kernels όπου αντί για την ευκλείδεια χρησιμοποιείται η Manhattan απόσταση μεταξύ των διανυσμάτων των δειγμάτων. Πιο συγκεκριμένα η απόσταση μεταξύ δυο δειγμάτων x, y με τη χρήση Laplacian_kernel υπολογίζεται ως $\mathrm{k}(x, y) = \exp(-\gamma \|x - y\|_1)$ όπου η παράμετρος γ=1/πλήθος_των_χαρακτηριστικών. Οι ομάδες δημιουργούνται με βάση αυτές τις αποστάσεις μεταξύ των δειγμάτων. Στην δεύτερη παραλλαγή χρησιμοποιείται το μητρώο γειτνίασης του γράφου κοντινότερων γειτόνων που προκύπτει από τα δεδομένα και η ομαδοποίηση γίνεται με βάση αυτό το μητρώο γειτνίασης. Και για τους 3 αλγορίθμους παραθέτω στον κώδικα το silhouette_score για πλήθος ομάδων από 2 μέχρι 30. Σε όλες τις περιπτώσεις το silhouette_score μειώνεται όσο αυξάνεται το πλήθος των ομάδων με την καλύτερη τιμή να επιτυγχάνεται για 2 ομάδες. Εδώ εξαιτίας της φύσης του προβλήματος χρησιμοποίησα 3 ομάδες σε όλους του αλγορίθμους, καθώς θα αναμέναμε με βάση το πρόβλημα κάθε ομάδα να αντιστοιχεί σε κάποια κατηγορία ασθενών ως προς την μεταβλητή fried. Για 3 ομάδες λοιπόν το silhouette_score για τον kmeans είναι 0.3218 για την ιεραρχική ομαδοποίηση είναι 0.3792 ενώ για την φασματική ομαδοποίηση με Laplacian_kernel 0.3433 και με τη χρήση nearest_neighbors 0.3473.

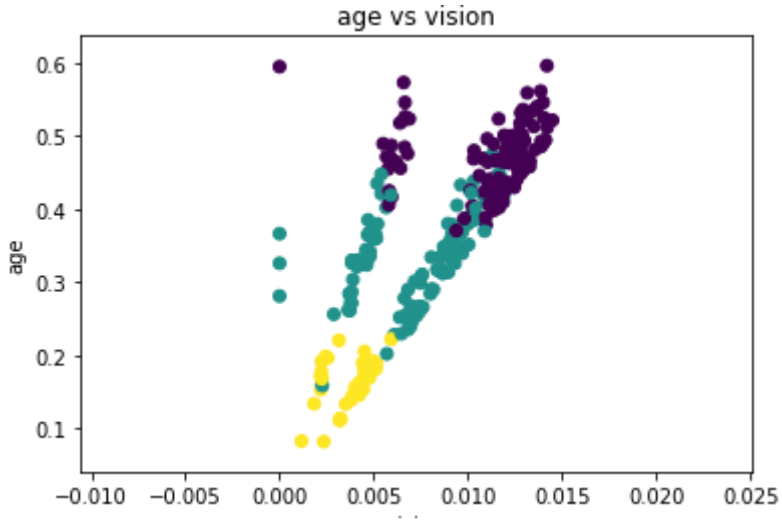
Εκτελώ την μέθοδο PCA για μείωση της διαστατικότητας, όπως προτείνεται, στοχεύοντας στη διατήρηση του 90% της διασποράς του αρχικού συνόλου δεδομένων. Η μέθοδος ανακτά 6 κυρίαρχες συνιστώσες (principal components) οι οποίες εκφράζουν το 91.11% της διασποράς των αρχικών δεδομένων. Αυτό σημαίνει πως από τις 52 διαστάσεις (αριθμός στηλών) στις οποίες επιχειρούσαμε να ομαδοποιήσουμε τα δεδομένα πριν τώρα βρισκόμαστε στις 6. Εκτελώ και πάλι όλους τους αλγορίθμους για 3 ομάδες σε αυτό το νέο 6-διάστατο χώρο. Τα αποτελέσματα στον δείκτη silhouette_score έχουν ως εξής: για τον kmeans 0.3719, για τον Agglomerative 0.2213 για τη φασματική ομαδοποίηση με laplacian_kernel 0.3396 ενώ με χρήση nearest_neighbors 0.4018. Για τον Agglomerative αλγόριθμο η απόδοση μειώνεται σημαντικά ενώ για την φασματική ομαδοποίηση με Laplacian_kernel η μείωση είναι ελάχιστη. Αντίθετα για τον kmeans και για την φασματική ομαδοποίηση με nearest_neighbors η μείωση διαστατικότητας επιφέρει σημαντική βελτίωση στο δείκτη silhouette_score.

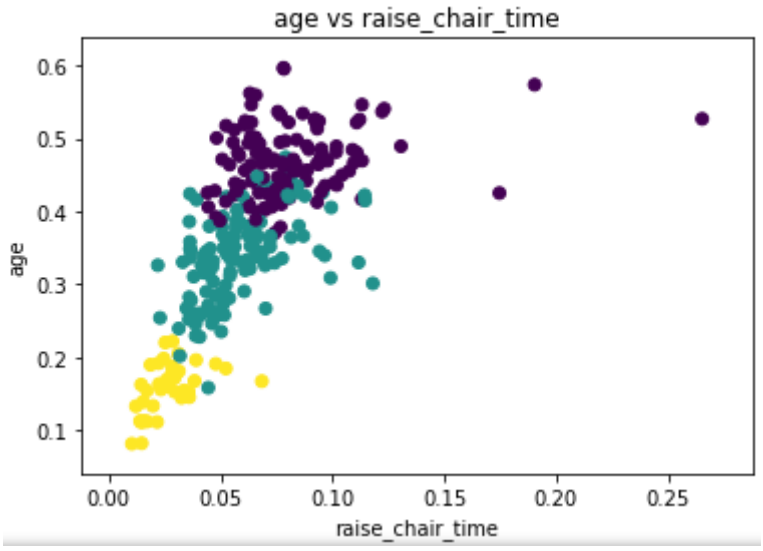
Part C

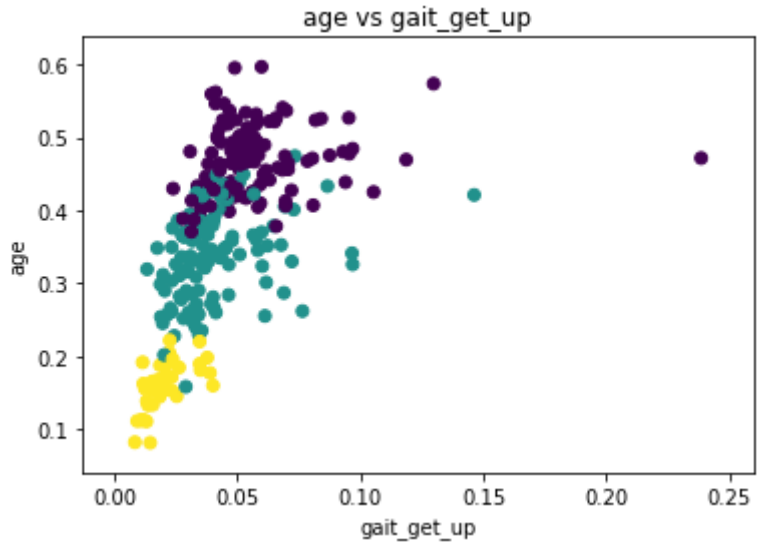
Εν συνεχεία η οπτικοποίηση των αποτελεσμάτων της ομαδοποίησης γίνεται παραθέτοντας διαγράμματα των ομαδοποιημένων δειγμάτων για κάθε ζεύγος χαρακτηριστικών. Με αυτόν τον τρόπο μπορούμε να δούμε με βάση ποια χαρακτηριστικά έγινε ο διαχωρισμός των ατόμων σε ομάδες και τι χαρακτηριστικά έχουν τα άτομα κάθε ομάδας. Όλα τα διαγράμματα που παραθέτουμε εδώ αντιστοιχούν στον αλγόριθμο φασματικής ομαδοποίησης με nearest_neighbors που είχε την καλύτερη απόδοση. Η αντιστοίχιση μεταξύ χρωμάτων και ‘ετικετών’ (labels) που προέκυψαν από την ομαδοποίηση είναι η εξής: το μωβ χρώμα

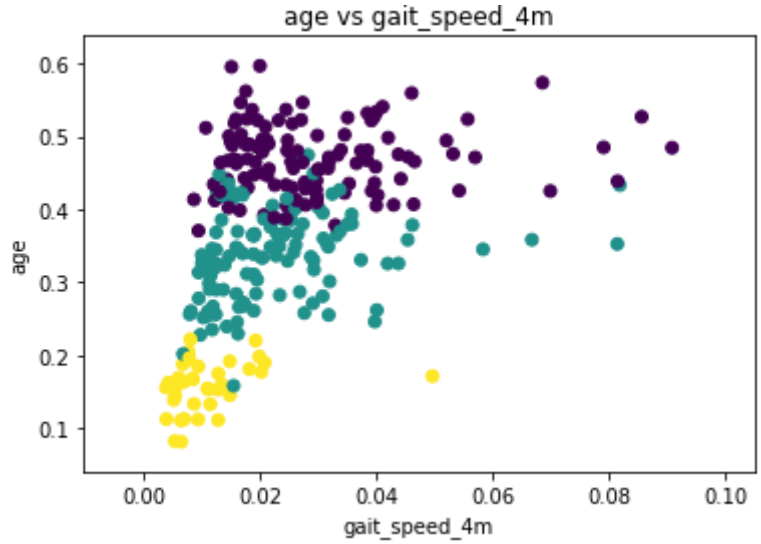
αντιστοιχεί στην ομάδα 0, το πράσινο χρώμα στην ομάδα 1 και το κίτρινο χρώμα στην ομάδα 2. Επισημαίνεται πως αυτές οι ετικέτες δεν έχουν καμία σχέση με τις ετικέτες που αφορούν την παράμετρο fried είναι απλά ο τρόπος που χρησιμοποιείται από τους αλγορίθμους για να αντιστοιχίζουν το κάθε δείγμα σε μια ομάδα.

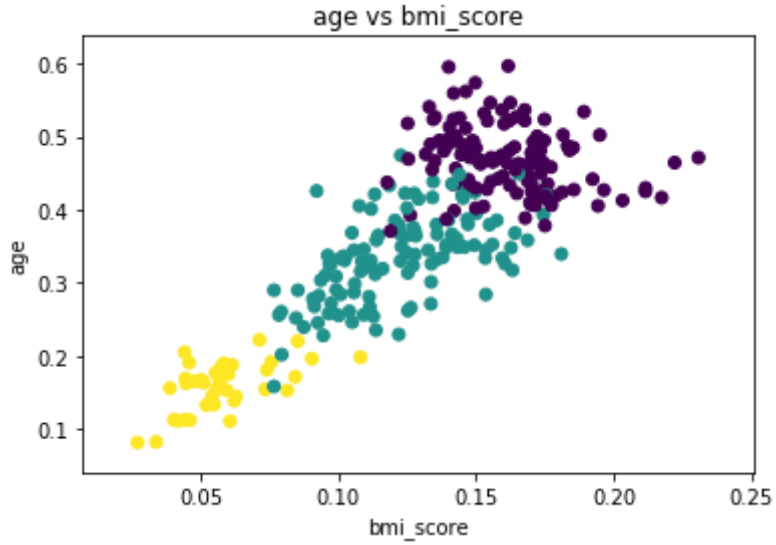
Στα διαγράμματα που ακολουθούν παραθέτω επιλεκτικά τα χαρακτηριστικά ως προς τα οποία φαίνεται να είναι σαφέστερος ο διαχωρισμός μεταξύ των ομάδων. Πιο συγκεκριμένα η ηλικία φαίνεται να είναι το κυρίαρχο χαρακτηριστικό ως προς το οποίο γίνεται ο διαχωρισμός μεταξύ των ομάδων και εδώ δείχνω τα διαγράμματα μεταξύ ηλικίας και κάποιων εκ των χαρακτηριστικών ως προς τα οποία οι ομάδες φαίνεται να διαφοροποιούνται επίσης. Η πλήρης λίστα όλων των διαγραμμάτων μεταξύ κάθε ζεύγους χαρακτηριστικών παρατίθεται στο notebook με τον κώδικα.

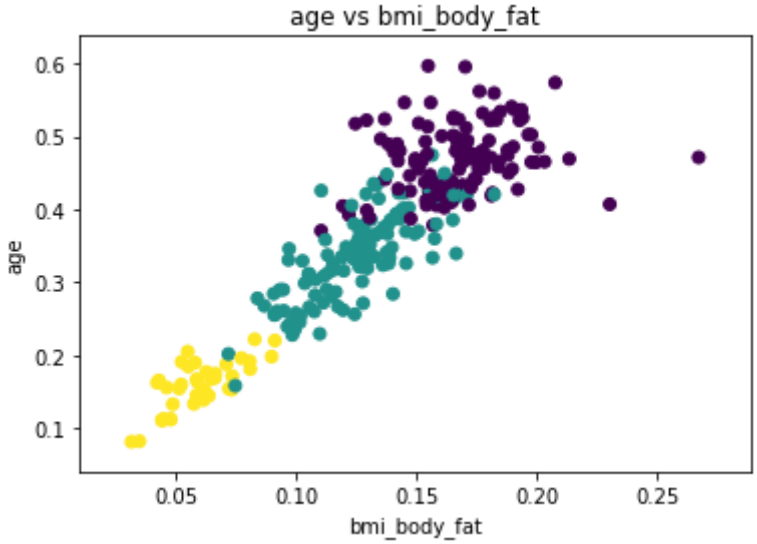


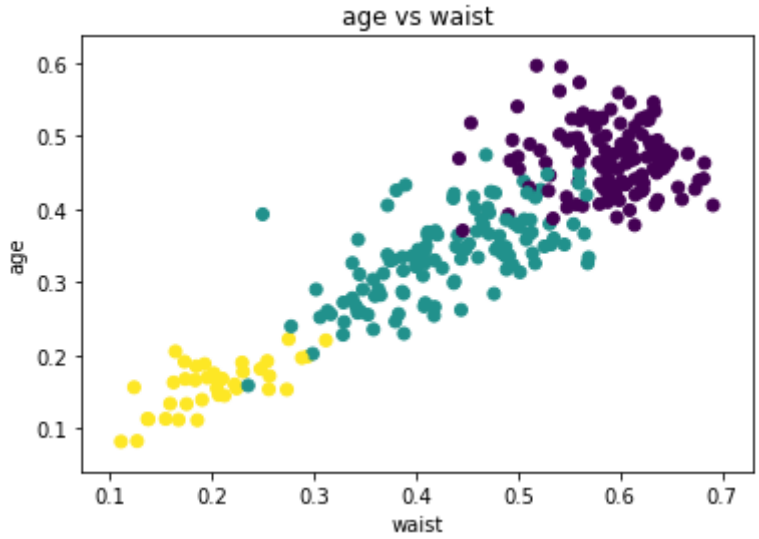


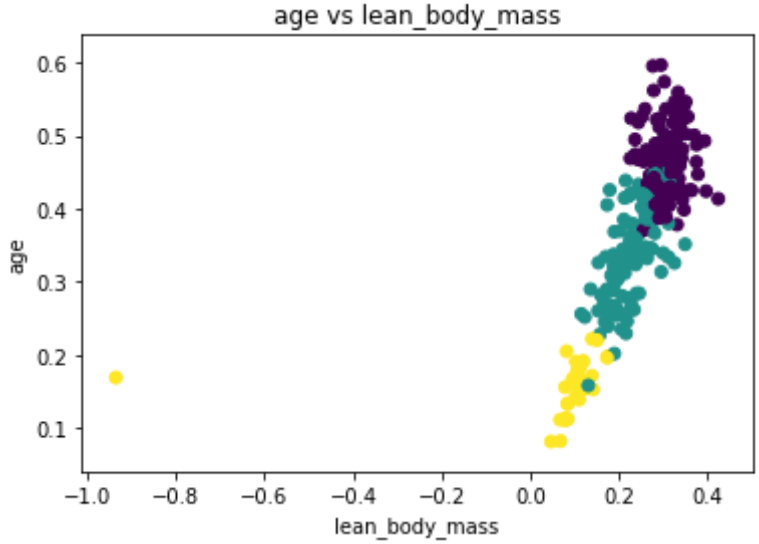

age vs gait_get_up
age
gait_get_up
0.1
0.2
0.3
0.4
0.5
0.6
0.00
0.05
0.10
0.15
0.20
0.25

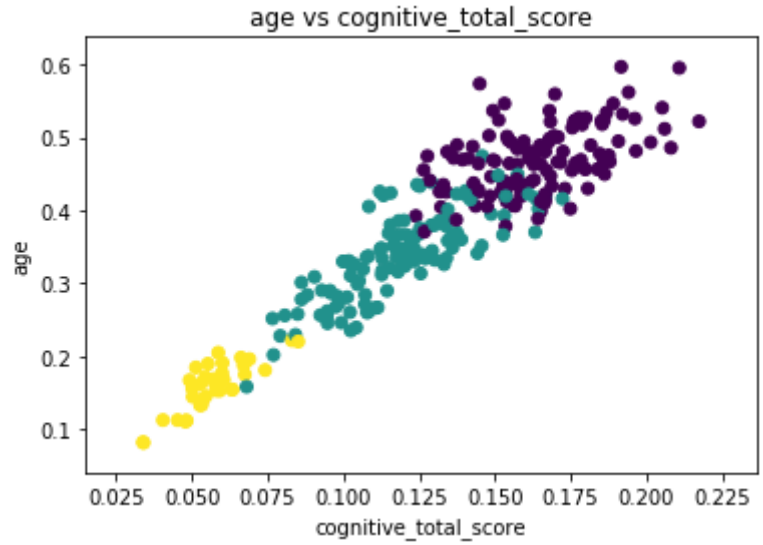

age vs gait_speed_4m
age
gait_speed_4m
0.1
0.2
0.3
0.4
0.5
0.6
0.00
0.02
0.04
0.06
0.08
0.10

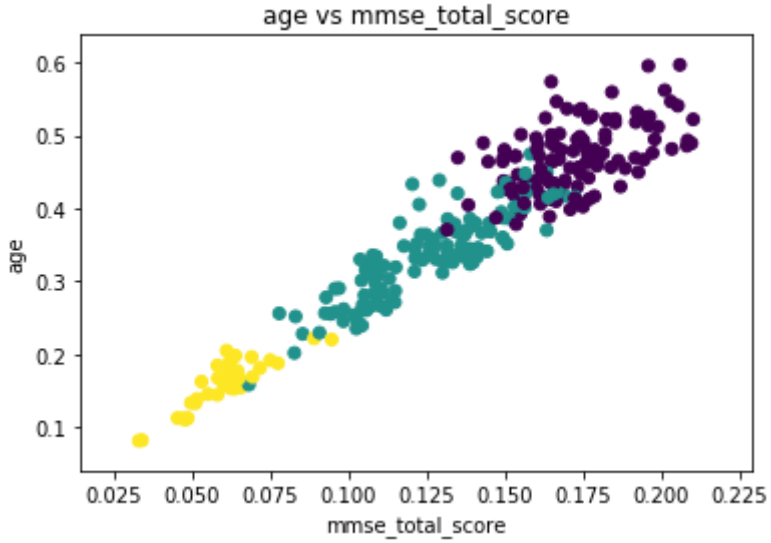

age vs bmi_score
age
bmi_score
0.1
0.2
0.3
0.4
0.5
0.6
0.05
0.10
0.15
0.20
0.25

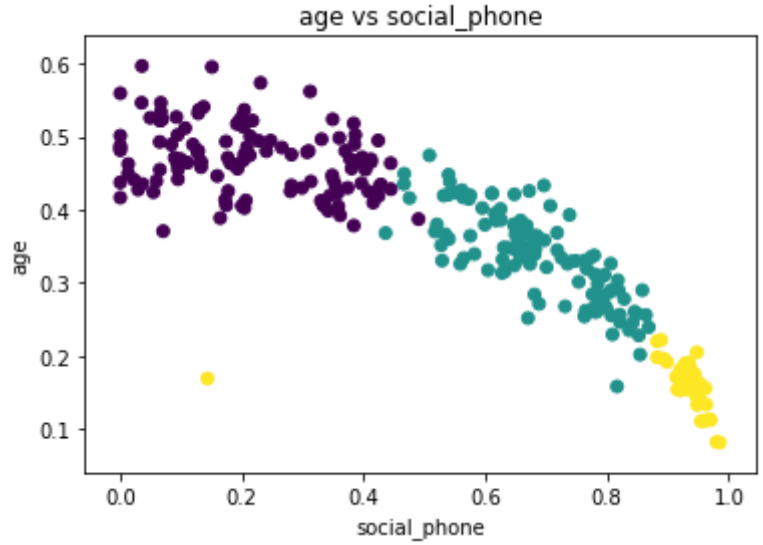

age vs bmi_body_fat
age
bmi_body_fat
0.6
0.5
0.4
0.3
0.2
0.1
0.05
0.10
0.15
0.20
0.25

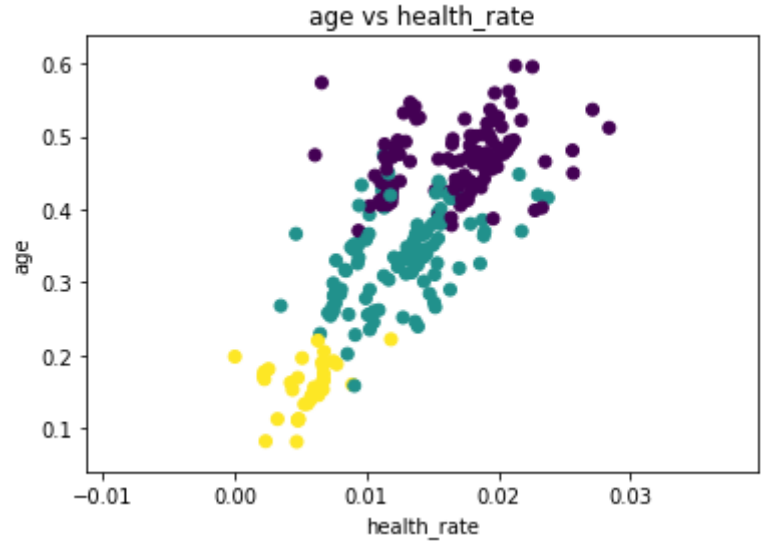

age vs waist
age
waist
0.6
0.5
0.4
0.3
0.2
0.1
0.1
0.2
0.3
0.4
0.5
0.6
0.7

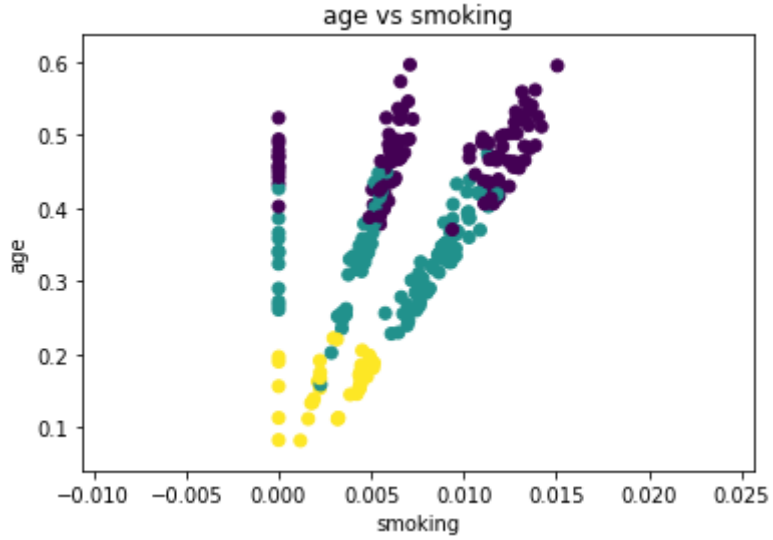

age vs lean_body_mass
age
lean_body_mass
0.6
0.5
0.4
0.3
0.2
0.1
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4

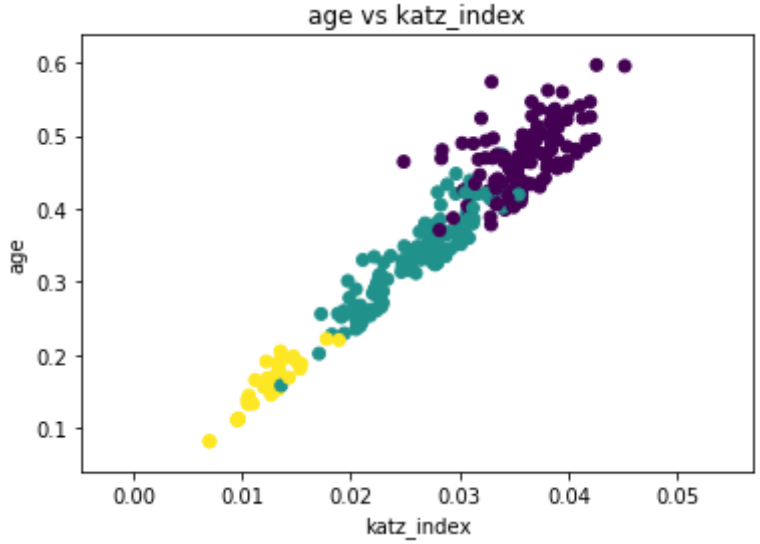

age vs cognitive_total_score
age
cognitive_total_score
0.1
0.2
0.3
0.4
0.5
0.6
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
0.225

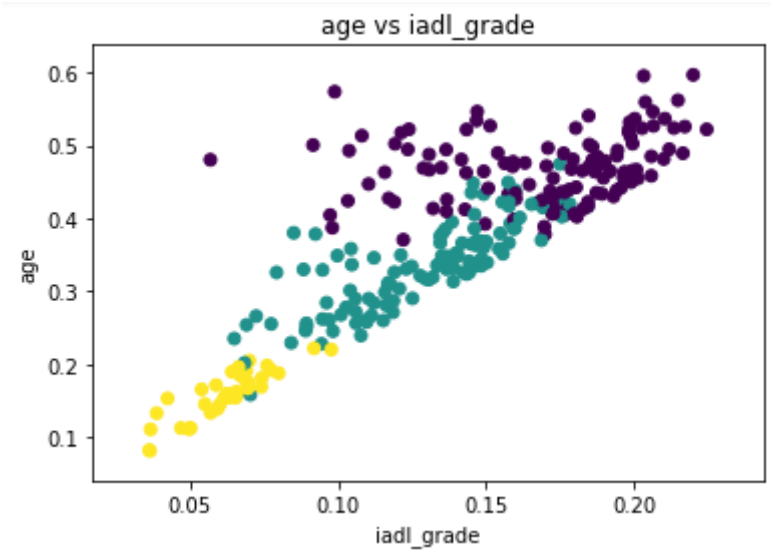

age vs mmse_total_score
age
mmse_total_score
0.1
0.2
0.3
0.4
0.5
0.6
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
0.225


age vs social_phone
age
social_phone
0.1
0.2
0.3
0.4
0.5
0.6
0.0
0.2
0.4
0.6
0.8
1.0


age vs health_rate
age
health_rate
0.6
0.5
0.4
0.3
0.2
0.1
−0.01
0.00
0.01
0.02
0.03


age vs smoking
age
smoking
0.6
0.5
0.4
0.3
0.2
0.1
−0.010
−0.005
0.000
0.005
0.010
0.015
0.020
0.025


age vs katz_index
age
katz_index
0.6
0.5
0.4
0.3
0.2
0.1
0.00
0.01
0.02
0.03
0.04
0.05

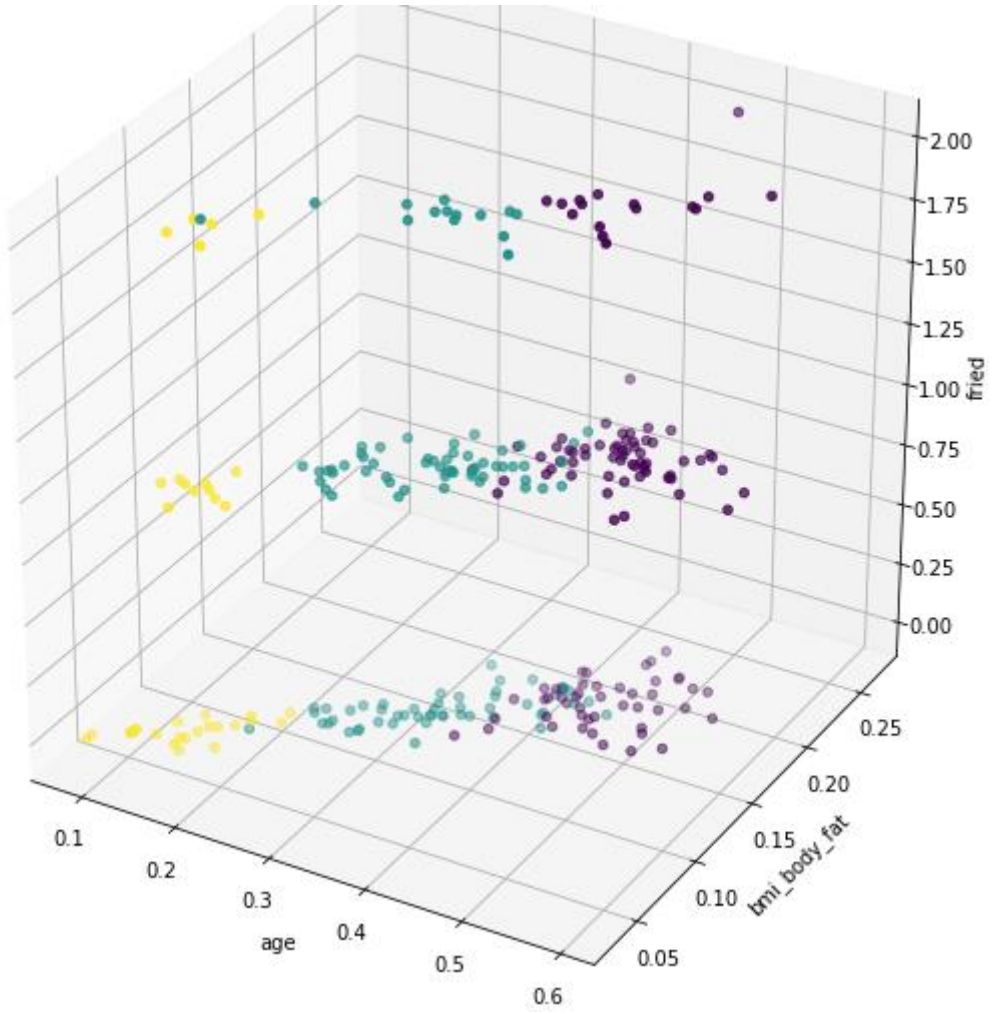


Παρατηρώ πως τα βασικότερα χαρακτηριστικά ως προς τα οποία παρουσιάζουν διαφοροποιήσεις οι ομάδες, πλην της ηλικίας, είναι ο δείκτης μάζας σώματος (bmi_score), το ποσοστό σωματικού λίπους (bmi_body_fat), το cognitive_total_score, το mmse_total_score, ο δείκτης katz, η άπαχη μάζα σώματος (lean_body_mass), η περίμετρος της μέσης (waist) και ο χρόνος που καταναλώνεται στο τηλέφωνο (social_phone). Σε μικρότερο βαθμό φαίνεται να παρουσιάζουν διαφοροποιήσεις τα δείγματα μεταξύ διαφορετικών ομάδων για τα χαρακτηριστικά κάπνισμα (smoking), το health_rate, το gait_get_up, ταχύτητα περπατήματος (gait_speed_4m) και όραση (vision).

Στην ομάδα 0 (μωβ) περιέχονται άτομα με μεγαλύτερη ηλικία που έχουν υψηλές τιμές δείκτη μάζας σώματος, ποσοστό σωματικού λίπους, άπαχη μάζα σώματος, μεγαλύτερη περίμετρο μέσης καθώς και υψηλές τιμές στα mmse_total_score, katz_index και cognitive_total_score ενώ καταναλώνουν πολύ λίγο χρόνο στο τηλέφωνο. Στην ομάδα 1 (πράσινο) περιέχονται άτομα μέσης ηλικίας, με χαμηλότερες τιμές σε όλα τα παραπάνω χαρακτηριστικά σε σχέση με τα δείγματα της ομάδας 1, εκτός του χρόνου που καταναλώνουν στον τηλέφωνο καθώς εκεί φαίνεται να ξοδεύουν περισσότερο χρόνο από τα άτομα της ομάδας 0. Τέλος τα άτομα της ομάδας 2 (κίτρινο) είναι τα αυτά με την μικρότερη ηλικία σε σχέση με τα υπόλοιπα, με τις χαμηλότερες τιμές δείκτη μάζας σώματος, ποσοστό σωματικού λίπους, άπαχη μάζα σώματος, μικρότερη περίμετρο μέσης καθώς και μικρές τιμές στα mmse_total_score, katz_index και cognitive_total_score ενώ καταναλώνουν πολύ περισσότερο χρόνο στο τηλέφωνο σε σχέση με τα υπόλοιπα άτομα.

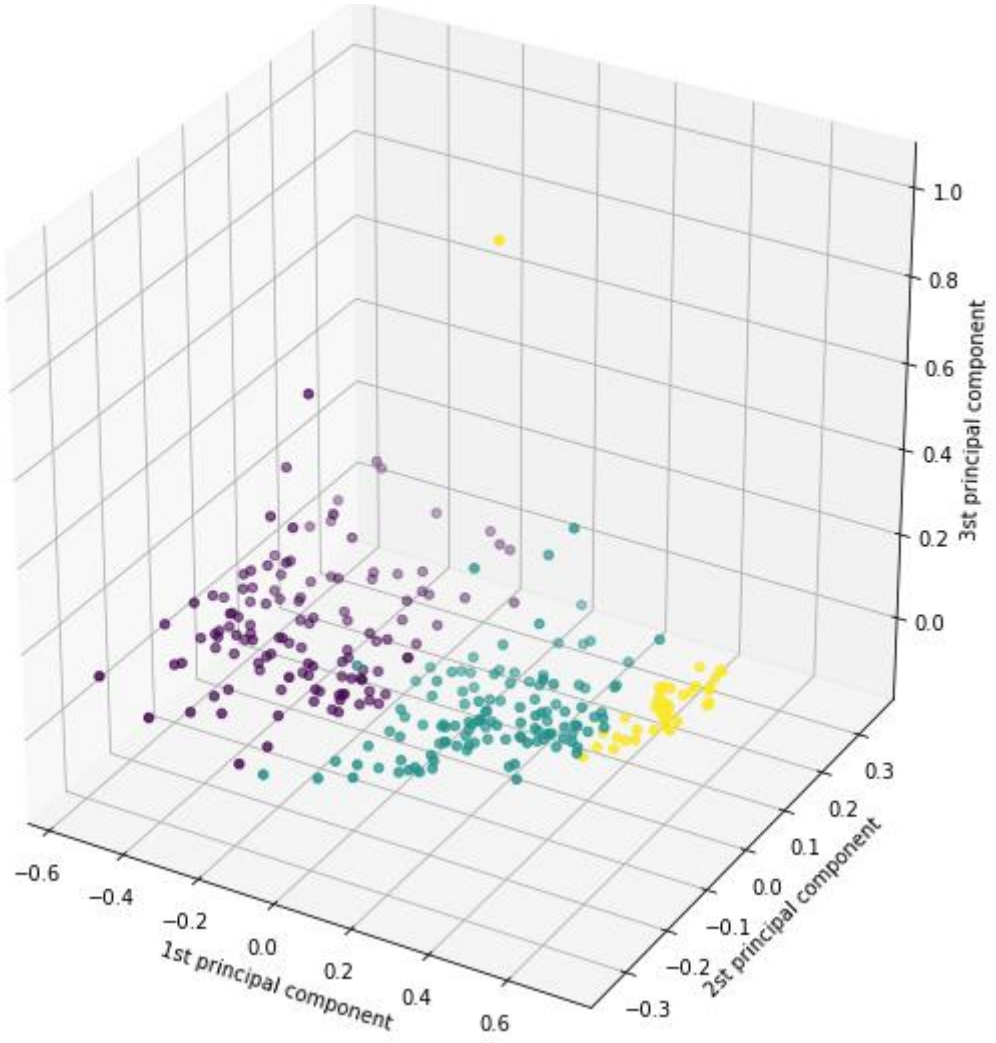
Με βάση τα παραπάνω στην ομάδα 2 (κίτρινο) φαίνεται να ανήκουν τα νεότερα, πιο υγιή και πιο κοινωνικά ενεργά άτομα, με βάση τα αποτελέσματα στις εξετάσεις και τους δείκτες ευεξίας που έχουμε στη διάθεση μας. Με το ίδιο σκεπτικό η ομάδα 1 (πράσινο) αποτελείται από τα λίγο μεγαλύτερης ηλικίας άτομα με μέση απόδοση στους δείκτες ευεξίας. Τέλος στην ομάδα 0 (μωβ) βρίσκονται τα άτομα μεγαλύτερης ηλικίας, με τα χειρότερα αποτελέσματα στις εξετάσεις και στους δείκτες ευεξίας και τη λιγότερο έντονη κοινωνική ζωή. Ως εκ τούτου θα μπορούσαμε σε κάποιο βαθμό, καθώς υπάρχουν αντικρουόμενες σχέσεις μεταξύ τιμών χαρακτηριστικών και αναμενόμενων ομάδων σε κάποιες περιπτώσεις, να πούμε πως η ομαδοποίηση κάνει αυτό το οποίο θα θέλαμε, βρίσκει δηλαδή σύνολα παρόμοιων ατόμων με βάση την κατάσταση της υγείας τους.

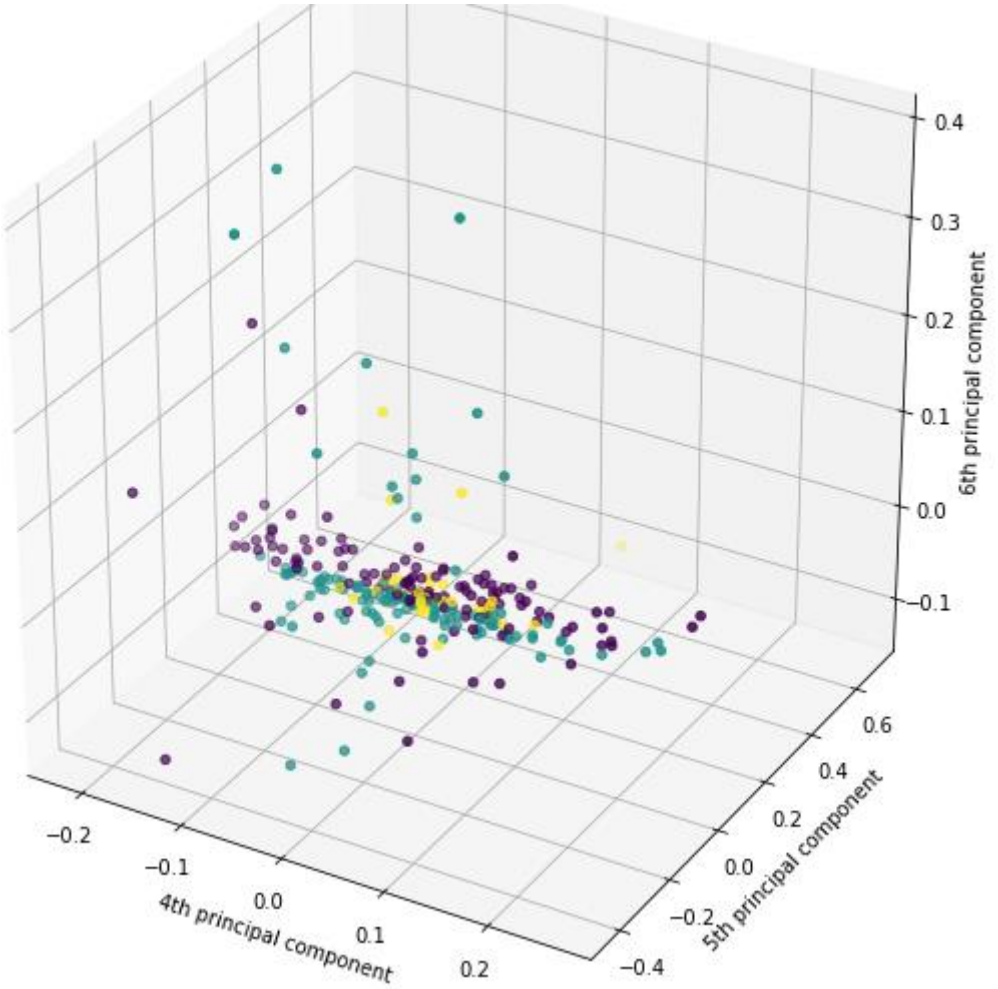
Το παρακάτω απεικονίζει τις ομάδες ως προς την ηλικία, το ποσοστό σωματικού λίπους και την παράμετρο fried.



Η ηλικία και το ποσοστό σωματικού λίπους είναι δυο εκ των χαρακτηριστικών ως προς τα οποία είναι πιο σαφής ο διαχωρισμός των ομάδων γι' αυτό και επιλέχθηκαν εδώ. Αυτό που θέλουμε να δούμε είναι αν υπάρχει κάποια συσχέτιση μεταξύ των ομάδων και της κατηγοριοποίησης των ατόμων σύμφωνα με την μεταβλητή fried. Αυτό που βλέπουμε είναι πως οι ομάδες 0 και 1 περιέχουν σχεδόν το ίδιο πλήθος ατόμων από κάθε κατηγορία (No-frail,Pre-frail,Frail). Αναλυτικότερα η ομάδα 0 περιέχει 47 άτομα της κατηγορίας No-frail, 59 άτομα της κατηγορίας Pre-frail και 16 της κατηγορίας Frail. Οι αντίστοιχοι αριθμοί για την ομάδα 1 είναι 51, 56 και 14. Η ομάδα 2 περιέχει μικρότερο πλήθος ατόμων και από τις 3 κατηγορίες, πιο συγκεκριμένα 23 άτομα της κατηγορίας Pre-frail, 10 Pre-frail και 5 Frail.

Τέλος παρατίθενται τα αποτελέσματα της ομαδοποίησης και ως προς τις κυρίαρχες συνιστώσες που προέκυψαν με την μέθοδο PCA.


1.0
0.8
0.6
0.4
0.2
0.0
3st principal component
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
2st principal component
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
1st principal component


0.4
0.3
0.2
0.1
0.0
−0.1
6th principal component
0.6
0.4
0.2
0.0
−0.2
−0.4
5th principal component
−0.2
−0.1
0.0
0.1
0.2
4th principal component