ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΠΡΑΓΜΑΤΩΝ:

ΕΥΦΥΗ ΠΕΡΙΒΑΛΛΟΝΤΑ ΣΕ ΔΙΚΤΥΑ ΝΕΑΣ ΓΕΝΙΑΣ

Ανίχνευση ήπιας γνωστικής εξασθένησης με χρήση παιχνιδιών σοβαρού σκοπού και αλγορίθμων μηχανικής μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Χρήστου Καράπαπα

Επιβλέπων : Χρήστος Γκουμόπουλος

Μέλη εξεταστικής επιτροπής:

Σάμος, Φεβρουάριος 2021

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον καθηγητή μου και επιβλέποντα, κ. Χρήστο Γκουμόπουλο για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου την εκπόνηση της συγκεκριμένης διπλωματικής, καθώς και για τη πολύτιμη καθοδήγηση που μου προσέφερε σε όλα τα στάδια της εκπόνησης της εργασίας.

Θα ήθελα επίσης να ευχαριστήσω τον υποψήφιο διδάκτορα Γεώργιο Σκίκο για την εποικοδομητική συνεργασία που είχαμε και τις πολύτιμες συμβουλές του που συνέβαλαν ουσιαστικά στην ολοκλήρωση αυτής της εργασίας.

Τέλος, θέλω να ευχαριστήσω θερμά την οικογένεια μου για την υπομονή, τη βοήθεια και τη συμπαράσταση τους όλο αυτό το διάστημα.

του

Χρήστου Καράπαπα

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Πίνακας περιεχομένων

Λίστα Σχημάτων

Λίστα Πινάκων

Ακρωνύμια

Ακρωνύμιο	Ονομασία
MCI	Mild Cognitive Impairment
AD	Alzheimer Disease
NC	Normal Cognition
MOCA	Montreal Cognitive Assessment
MMSE	Mini Mental State Examination
CRISP-DM	Cross-industry standard process for data mining
TDSP	Team Data Science Process
EDA	Exploratory Data Analysis
ETL	Extract Transform Load
PCA	Principal Component Analysis
PC	Principal Component
PMC	Production Model Creation
SMOTE	Synthetic Minority Oversampling Technique
MDI	Mean Decrease in Impurity
MDA	Mean Decrease in Accuracy
ROC	Receiver Operating Characteristics
AUC	Area Under Curve
API	Application Programming Interface
CSAPI	Classification Service API
REST	Representational State Transfer

Περίληψη

Στη παρούσα διπλωματική εργασία εξετάζουμε το θέμα της ήπιας γνωστικής εξασθένησης, σε συνδυασμό με τα παιχνίδια σοβαρού σκοπού και τους αλγόριθμους μηχανικής μάθησης.

Ο κύριος σκοπός της εργασίας είναι να απαντήσει στο ερώτημα του κατά πόσο είναι εφικτό να δημιουργηθεί ένα μοντέλο μηχανικής μάθησης το οποίο θα είναι σε θέση να κατηγοριοποιήσει έναν χρήστη μιας εφαρμογής παιχνιδιών σοβαρού σκοπού σε επιμέρους κατηγορίες γνωστικής επάρκειας.

Το θέμα προσεγγίστηκε με μια προσαρμοσμένη μεθοδολογία, η οποία περιλαμβάνει μια σειρά διαδικασιών μέσω των οποίων προσπαθούμε να φτάσουμε στο σημείο όπου θα δημιουργήσουμε ένα μοντέλο το οποίο θα παρέχει έναν ικανοποιητικό βαθμό ακρίβειας για τη τρέχουσα συλλογή δεδομένων και έναν βαθμό αξιοπιστίας για δεδομένα μελλοντικών ερευνών. Τα δεδομένα που αναλύονται αποτελούν τη καταγραφή των επιδόσεων, μιας συγκεκριμένης ομάδας εστίασης, στην εφαρμογή παιχνιδιών σοβαρού σκοπού MCI Rehab.

Στα πρώτα βήματα της μεθοδολογίας, επικεντρωνόμαστε σε διαδικασίες που έχουν να κάνουν με τη μεταφορά και το μετασχηματισμό των δεδομένων, έως ότου αυτά έρθουν στη μορφή που επιθυμούμε.

Στη συνέχεια, ο κύριος όγκος της εργασίας αφορά τη διερευνητική ανάλυση των δεδομένων, με σκοπό να προκύψουν συμπεράσματα για το ποια θα ήταν τα ιδανικά Features, ο πιο αποδοτικός αλγόριθμος αλλά και οι κατάλληλες μέθοδοι βελτιστοποίησης για τη δημιουργία του επιθυμητού μοντέλου.

Έπειτα θα μας απασχολήσουν διαδικασίες που έχουν να κάνουν με τη δημιουργία μιας παραγωγικής έκδοσης του μοντέλου που έχει κριθεί ως ιδανικό από τα προηγούμενα βήματα, αλλά και της ενσωμάτωσής του σε μια υπηρεσία η οποία θα μπορούσε να λειτουργήσει ως ένα REST API.

Κλείνοντας πριν καταλήξουμε στα συμπεράσματα, γίνεται αναλυτική αναφορά στα αποτελέσματα του κάθε μοντέλου σε συγκεκριμένα Metrics, τόσο για τα πειραματικά μοντέλα που δημιουργήθηκαν κατά τη διερευνητική ανάλυση όσο και για αυτό που ονομάζουμε παραγωγικό μοντέλο.

Κάτω από ιδανικές συνθήκες, μια τέτοια υπηρεσία θα μπορούσε να κάνει χρήση ενός ή και περισσοτέρων από τα μοντέλα που έχουν δημιουργηθεί από τη προσαρμοσμένη μεθοδολογία με σκοπό να αποτελέσουν ένα συμπληρωματικό εργαλείο ανίχνευσης της ήπιας γνωστικής εξασθένησης.

Λέξεις Κλειδιά: *Ήπια γνωστική εξασθένηση, παιχνίδια σοβαρού σκοπού, χαρακτηριστικά (ανεξάρτητες μεταβλητές δεδομένων), εξαρτημένη μεταβλητή δεδομένων, μετασχηματισμοί δεδομένων, scaling, κατηγοριοποίηση, δείκτες απόδοσης, βελτιστοποίηση*

Abstract

In this thesis we examine the topic of Mild Cognitive Impairment, in conjunction with serious games and machine learning algorithms.

The main purpose of this work is to give an answer to the question of whether it is possible to develop a machine learning model that will be able to classify a user of a serious-games application to distinct classes of cognitive competence.

The way we approach this issue, is by creating a custom methodology, which consists of a series of processes that we are going to use to create a machine learning model that will be able to classify a user with sufficient confidence and that will be able to retain sufficient accuracy for datasets that will be generated in future studies. The dataset used for this purpose is the record of in-game performance for a focus group in the application of serious games MCI Rehab.

During the first steps of this custom methodology, we focus on processes relative to data manipulation until the dataset reaches a certain state in terms of storage and format.

In the following steps, the bulk of the work focuses on the exploratory analysis of the data, through which we try to draw conclusions for the ideal feature set to use, the algorithm that creates the most accurate model and the processes to follow for the optimization of the model.

Later we deal with processes that are related to creating a production level version of the model and its integration to a service that will play the role of a REST API.

Before we draw the final conclusions related to this work, we analyze the results of each model created during the exploratory analysis process and of the production model.

In ideal circumstances, such a service could make use of one or more of the models created in order to play the role of a complementary process to the existing process of Mild Cognitive Impairment detection.

Keywords: *mci, serious-games, features, target-class, transformation, scaling, classification, metric, optimization*

1

Εισαγωγή

1.1 Αντικείμενο

Το αντικείμενο της παρούσας διπλωματικής είναι η ανίχνευση ήπιας γνωστικής εξασθένησης (Mild Cognitive Impairment, MCI) σε ενήλικες της τρίτης ηλικίας, με χρήση παιχνιδιών σοβαρού σκοπού και μοντέλα μηχανικής μάθησης.

Όπως αναφέρεται και πιο αναλυτικά στην υποενότητα των θεματικών πεδίων, η ήπια γνωστική εξασθένηση χαρακτηρίζεται ως η, μικρής κλίμακας, έκπτωση μέρους των γνωστικών λειτουργιών ενός ανθρώπου, ιδιαίτερα όταν αυτή η εξασθένηση δε συνάδει με το προφίλ του υποκειμένου.

Όσον αφορά τα παιχνίδια σοβαρού σκοπού, στα οποία θα αναφερθούμε πιο αναλυτικά στα επόμενα κεφάλαια, θα μας απασχολήσει μια υπάρχουσα εφαρμογή με τίτλο MCI Rehab [1]. Η εφαρμογή αποτελείται από ένα σύνολο παιχνιδιών τα οποία δημιουργήθηκαν με γνώμονα την εξάσκηση ατόμων με διαγνωσμένη MCI και στόχο τη βελτίωση των επιμέρους γνωστικών λειτουργιών όπως για παράδειγμα η μνήμη, ο προσανατολισμός, η προσοχή και η αντίληψη.

Το πεδίο στο οποίο επικεντρώνεται η παρούσα διπλωματική αφορά τη δημιουργία μοντέλων για την αναγνώριση περιπτώσεων με MCI. Στα κεφάλαια που ακολουθούν θα δούμε τη διαδικασία στο σύνολό της, από την εισαγωγή και την επεξεργασία των δεδομένων, την δημιουργία βασικών μοντέλων και τη βελτιστοποίησή τους, ενώ τέλος θα δούμε τη δημιουργία μιας υπηρεσίας η οποία θα κάνει χρήση των μοντέλων με σκοπό τη κατηγοριοποίηση ενός υποκειμένου βάση των νέων δεδομένων που συλλέγει η εφαρμογή MCI Rehab.

Ο λόγος για τον οποίο παρουσιάζει ενδιαφέρον το συγκεκριμένο αντικείμενο είναι διότι σύμφωνα με μελέτη του Παγκόσμιου Οργανισμού Υγείας που δημοσιεύτηκε το 2012, πάνω από 35 εκατομμύρια άνθρωποι παγκοσμίως πάσχουν από κάποιας μορφής άνοια, ενώ αυτός ο αριθμός αναμένεται να φτάσει τα 115 εκατομμύρια μέχρι το 2050 [17].

1.2 Θεματικά Πεδία

1.2.1 Ήπια γνωστική διαταραχή

Αναφορικά με την ήπια γνωστική διαταραχή δεν θα μπορούσαμε να τη ταυτίσουμε με ένα μοναδικό κλάδο της ιατρικής διότι απασχολεί ένα ευρύ φάσμα ιατρικών και μη ιατρικών επιστημών που εξετάζουν το ίδιο ζήτημα κάτω από διαφορετικό πρίσμα. Για παράδειγμα, έχουμε τη νευρολογία η οποία ασχολείται με τη κλινική διάγνωση της διαταραχής και τη θεραπεία των ασθενών [12]. Επιπλέον, υπάρχει ο κλάδος τη γηριατρικής που περικλείει οτιδήποτε σχετίζεται με τη φροντίδα των ηλικιωμένων και τη βελτίωση του βιοτικού επιπέδου τους [14]. Ενώ παράλληλα έχουμε το κλάδο της φαρμακολογίας που βρίσκεται σε διαρκή αναζήτηση για τη βελτίωση των φαρμάκων που έχουν να κάνουν με νευροεκφυλιστικές διαταραχές [13].

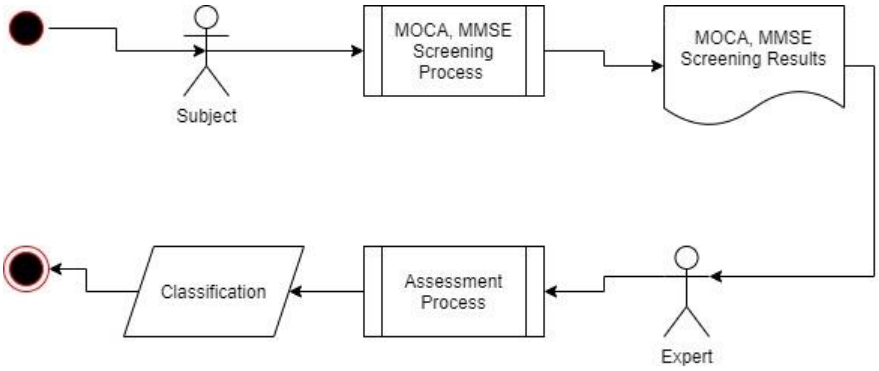
Υπάρχουν πολλές ενδιαφέρουσες πληροφορίες που σχετίζονται με τους τρεις παραπάνω κλάδους, τόσο σχετικά με την ιστορία όσο και με τις τρέχουσες εξελίξεις στο ζήτημα. Παρόλα αυτά, αυτό που μας ενδιαφέρει να εξετάσουμε ως προς την MCI είναι τα χαρακτηριστικά που παρουσιάζει ως διαταραχή και το πως γίνεται μέχρι σήμερα η διάγνωση, έτσι ώστε να μπορέσουμε να αντιπαραθέσουμε σε αυτή μια πρόταση για μια συμπληρωματική διαδικασία έγκαιρης διάγνωσης.

Αρχικά εξετάζοντας το τι ακριβώς είναι η MCI, θα λέγαμε ότι πρόκειται για την έκπτωση μιας ή και περισσοτέρων γνωστικών λειτουργιών ενός υποκειμένου, οι οποίες δε δικαιολογούνται με βάση το ηλικιακό, το μαθησιακό και το ιατρικό ιστορικό του. Πολλές φορές στη βιβλιογραφία αναφέρεται ως πρόδρομος σοβαρότερων νευροεκφυλιστικών παθήσεων όπως είναι το Alzheimer [2].

Η τυπική διαδικασία που εφαρμόζεται στις μέρες μας για την διάγνωση της MCI είναι μια δια ζώσης κλινική εξέταση από νευρολόγο. Η εξέταση χωρίζεται στα εξής επιμέρους στάδια:

- Μια συνέντευξη με το ίδιο το υποκείμενο για τη διαπίστωση της κατάστασης των γνωστικών λειτουργιών.
- Συγκέντρωση οικογενειακού και ατομικού ιατρικού ιστορικού. Στα πλαίσια της διαδικασίας είναι τυπικό να καλούνται συγγενείς του υποκειμένου να συμπληρώσουν κάποιο ερωτηματολόγιο, διότι πολλές φορές ένα συγγενικό πρόσωπο μπορεί να έχει καλύτερη εικόνα της μεταβολής των γνωστικών λειτουργιών του υποκειμένου.
- Μια σειρά από νευρολογικές εξετάσεις που έχουν ως στόχο να δοκιμάσουν την κινητικότητα, την ισορροπία, τα αντανακλαστικά, τις εγκεφαλικές συζυγίες, και άλλες δοκιμασίες που μπορούν να αναδείξουν κάποιο πρόβλημα στη λειτουργικότητα του νευρικού συστήματος.
- Τέλος, την αξιολόγηση των γνωστικών λειτουργιών μέσω νευροψυχολογικών δοκιμασιών όπως είναι το Montreal Cognitive Assessment (MOCA) [18] και το Mini Mental State Examination (MMSE) [20].

Στην Εικόνα 1 παρατηρούμε το διάγραμμα των βημάτων της διαδικασίας με επίκεντρο την αξιολόγηση των γνωστικών λειτουργιών, όπως αυτή πραγματοποιείται με την υπάρχουσα διαδικασία.



Εικόνα 1 Υπάρχουσα διαδικασία εξέτασης και αξιολόγησης της γνωστικής επάρκειας ενός υποκειμένου.

Ο λόγος για τον οποίο υπάρχει μια επιπλέον διαδικασία αξιολόγησης από τον ειδικό είναι το γεγονός ότι τα αποτελέσματα της δοκιμασίας δε μπορούν να μας δώσουν πάντοτε από μόνα τους ένα ασφαλές συμπέρασμα, οπότε ο εκάστοτε ειδικός πρέπει να λαμβάνει υπόψη του και άλλους παράγοντες όπως είναι για παράδειγμα η ηλικία και το επίπεδο της μόρφωσης του υποκειμένου [5]. Αυτό συμβαίνει διότι για παράδειγμα κάποια αποτελέσματα μπορούν να δικαιολογηθούν βάση του προφίλ του υποκειμένου ενώ άλλα να μην δικαιολογούνται πράγμα που θα ήταν ένδειξη ύπαρξης κάποιας μορφής MCI.

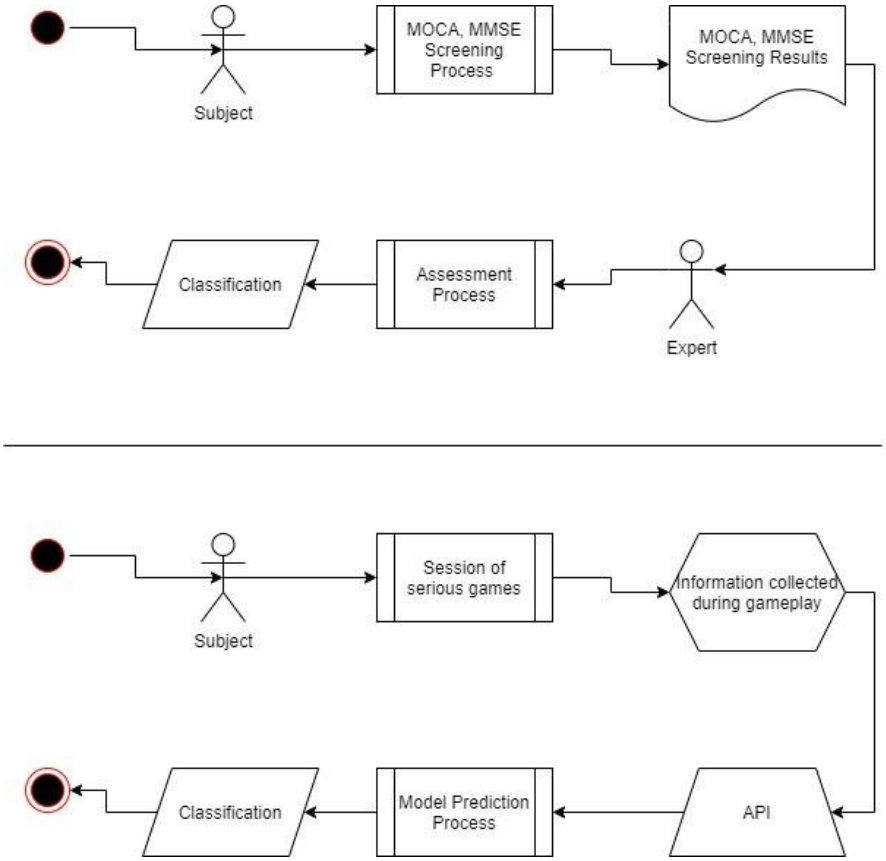
Η παραπάνω διαδικασία παρουσιάζει κάποια χαρακτηριστικά τα οποία μπορούν να θεωρηθούν ως μειονεκτήματα, όπως για παράδειγμα το ότι η διαδικασία γίνεται υπό τη μορφή κλινικής εξέτασης και το υποκείμενο μπορεί να παρουσιάσει χαμηλότερες από τις αναμενόμενες επιδόσεις στις δοκιμασίες. Η κύρια αιτία συνήθως είναι το άγχος που μπορεί να παρουσιάσει το υποκείμενο γνωρίζοντας ότι εκείνη τη στιγμή αξιολογείται.

Επιπλέον οι νευροψυχολογικές δοκιμασίες τύπου MOCA και MMSE, επειδή ακριβώς γίνονται στα πλαίσια κλινικής εξέτασης, έχουν χαμηλή επαναληψιμότητα [42]. Πράγμα που σημαίνει ότι πρώτον, επειδή δεν γίνεται συχνός επανέλεγχος των γνωστικών λειτουργιών πιθανώς να μην υπάρχει καλή εικόνα για την εξέλιξη της MCI του υποκειμένου. Δεύτερον, για τον ίδιο λόγο της χαμηλής επαναληψιμότητας, μια αξιολόγηση στην οποία για τον οποιοδήποτε λόγο το υποκείμενο έχει παρουσιάσει κακή εικόνα, μπορεί να οδηγήσει σε λάθος συμπεράσματα.

Η βασική ιδέα που αποτέλεσε έναυσμα αυτής της διπλωματικής είναι μια διαδικασία η οποία θα μπορεί να λειτουργήσει συμπληρωματικά στην υπάρχουσα και να βελτιώσει τον τρόπο με τον οποίο καταγράφεται το επίπεδο των γνωστικών λειτουργιών ενός υποκειμένου. Αυτό θα είχε ως αποτέλεσμα τη γενικότερη βελτίωση της διαδικασίας της έγκαιρης διάγνωσης της MCI.

Με την προϋπόθεση ότι η απάντηση στο βασικό ερευνητικό ερώτημα είναι θετική, κάτι το οποίο φαίνεται να ισχύει με βάση τα συμπεράσματα μετά την αξιολόγηση των μοντέλων, μπορούμε να πούμε ότι η χρήση παιχνιδιών σοβαρού σκοπού και η αξιολόγηση των καταγεγραμμένων δεδομένων με αλγόριθμους μηχανικής μάθησης θα μπορούσε να συνδράμει στην έγκαιρη διάγνωση με αρκετά καλή ακρίβεια.

Αντιπαραθέτοντας τη διαδικασία αξιολόγησης με χρήση παιχνιδιών σοβαρού σκοπού στο πρότυπο της υπάρχουσας διαδικασίας προκύπτει το διάγραμμα στην Εικόνα 2.



Εικόνα 2 Αντιπαραβολή της υπάρχουσας διαδικασίας εξέτασης και αξιολόγησης της γνωστική επάρκειας ενός υποκειμένου, με την προτεινόμενη αντίστοιχη με χρήση παιχνιδιών σοβαρού σκοπού και μοντέλων μηχανικής μάθησης.

Στο πλαίσιο της διπλωματικής αυτό που μας ενδιαφέρει αναφορικά με τη διαδικασία της εξέτασης είναι η συνέντευξη με το υποκείμενο αλλά κυρίως η νευροψυχολογική δοκιμασία MOCA.

Ο λόγος για τον οποίο είναι σημαντική η έγκαιρη διάγνωση της MCI είναι το ότι τα υποκείμενα που έχουν ήδη MCI με βάση τη δοκιμασία MOCA, τείνουν στο να βλέπουν μια μείωση της γνωστικής τους επάρκειας σε αντίθεση με τα υποκείμενα που παρουσιάζουν φυσιολογική γνωστική επάρκεια, όπου εκεί παρατηρούμε σταθερά αποτελέσματα σε διαδοχικές εξετάσεις. Για παράδειγμα, σε έρευνα που δημοσιεύτηκε το 2016, με 139 συνολικά συμμετέχοντες και δύο διαδοχικές δοκιμασίες MOCA με 3.5 χρόνια απόσταση, όσοι είχαν χαρακτηριστεί ήδη από τη πρώτη δοκιμασία με MCI, στη δεύτερη παρουσίασαν μια μείωση κατά 1.7 μονάδες, σε σύγκριση με όσους κατά τη πρώτη δοκιμασία είχαν φυσιολογική γνωστική επάρκεια των οποίων τα αποτελέσματα παρέμειναν σταθερά και στη δεύτερη [16].

1.2.2 Παιχνίδια σοβαρού σκοπού

Σε αντίθεση με τη πλειονότητα των παιχνιδιών τα οποία έχουν ως στόχο αποκλειστικά τη διασκέδαση και τη ψυχαγωγία του χρήστη, τα παιχνίδια σοβαρού σκοπού έχουν ως στόχο να πετύχουν ένα συγκεκριμένο αποτέλεσμα αναφορικά με τον χρήστη.

Τα παιχνίδια αυτά μπορούν να χωριστούν περαιτέρω σε δύο κατηγορίες, αυτά που έχουν ως στόχο να εξάγουν μετρήσεις σχετικά τις επιδόσεις του χρήστη σε συγκεκριμένες εργασίες και αυτά που έχουν δημιουργηθεί με γνώμονα να βελτιώσουν κάποιες από τις δεξιότητες του χρήστη. Στη περίπτωση που εξετάζουμε, οι δεξιότητες στις οποίες δοκιμάζεται ο χρήστης, είναι δεξιότητες που επηρεάζονται από

παράγοντες που προκαλούν άνοια, οπότε το επιθυμητό αποτέλεσμα των παιχνιδιών σοβαρού σκοπού είναι η βελτίωση αυτών των δεξιοτήτων, με απώτερο στόχο τη βελτίωση του βιοτικού επιπέδου του χρήστη.

Συγκεκριμένα για την εφαρμογή MCI Rehab, μέσω της οποίας συλλέχθηκαν τα δεδομένα που αναλύονται στη παρούσα διπλωματική, εντάσσεται στο είδος παιχνιδιών σοβαρού σκοπού και μάλιστα στη δεύτερη κατηγορία, όπου στόχος είναι η βελτίωση της γνωστικής κατάστασης του χρήστη.

Επιπλέον ως παράδειγμα παιχνιδιού σοβαρού σκοπού που προσανατολίζονται στη βελτίωση των γνωστικών λειτουργιών είναι το WarCAT το οποίο αναλύεται στο κεφάλαιο της υπάρχουσας βιβλιογραφίας.

Τέλος στην κατηγορία παιχνιδιών σοβαρού σκοπού, οι οποίες σχετίζονται με την MCI, θα μπορούσαμε να θεωρήσουμε πως εντάσσονται όλες οι εφαρμογές όπου ενώ στη πραγματικότητα αποτελούν νευροψυχολογικές δοκιμασίες έχουν σχεδιαστεί με γνώμονα να προσομοιώνουν κάποιο παιχνίδι ακολουθώντας τις αρχές του Gamification με σκοπό είτε να καταγράψουν είτε να βελτιώσουν το επίπεδο της γνωστικής επάρκειας του χρήστη [43].

1.2.3 Μηχανική Μάθηση

Τα τελευταία χρόνια λόγω της μεγάλης αύξησης των ιατρικών δεδομένων που βρίσκονται σε ηλεκτρονική μορφή παρατηρείται όλο και μεγαλύτερη εμπλοκή του πεδίου της μηχανικής μάθησης στο τομέα της ιατρικής.

Τόσο σε θέματα που έχουν να κάνουν με τη πρόγνωση, τη διάγνωση, τη θεραπεία αλλά και την αποκατάσταση, η μηχανική μάθηση καθώς και συναφή αντικείμενα όπως η υπολογιστική όραση, η στατιστική και τα μεγάλα δεδομένα, χρησιμοποιούνται για τη βελτίωση των υπαρχόντων διαδικασιών και για την εξεύρεση νέων λύσεων.

Από αυτή την αύξηση των διαθέσιμων πληροφοριών προκύπτουν ερωτήματα, όπως για παράδειγμα ποιος είναι ο κατάλληλος τρόπος επεξεργασίας των δεδομένων και ποια κριτήρια θα πρέπει να εφαρμόζονται για να κριθεί μια μεταβλητή ως σημαντική ή όχι, όταν εξετάζεται ένα ερευνητικό ερώτημα. Σε τέτοιου είδους ερωτήματα, μειώνοντας τον παράγοντα της ανθρώπινης κρίσης και αφήνοντας τους αλγόριθμους μηχανικής μάθησης να ερμηνεύσουν τις συσχετίσεις μεταξύ των δεδομένων, προκύπτουν μοντέλα τα οποία μπορούμε να χαρακτηρίσουμε ως μοντέλα μηχανικής μάθησης [35].

Όσον αφορά τα δεδομένα που σχετίζονται με την υγεία, αυτά μπορούν να προέρχονται από ποικίλες πηγές διαφορετικού είδους. Για παράδειγμα, πέρα από τα δεδομένα που αφορούν διαγνώσεις οι οποίες καταγράφονται σε αρχεία τύπου Electronic Health Records (EHR), ιατρικά δεδομένα μπορούν να προέλθουν και από άλλες πηγές όπως Fitness Trackers, γενετικές εξετάσεις, πληροφορίες σχετικά με τον τρόπο ζωής, κοινωνικές συνήθειες, οικογενειακό ιστορικό ακόμη και το περιβάλλον στο οποίο ζει ένα άτομο [35], [36].

Με βάση τα παραπάνω λοιπόν, είναι πολύ πιθανό να καταφέρουμε να δημιουργήσουμε ένα μοντέλο το οποίο να περιγράφει, σε ικανοποιητικό βαθμό, τη συσχέτιση του επιπέδου των γνωστικών λειτουργιών ενός ατόμου, με τα δεδομένα που παράγει χρησιμοποιώντας μια εφαρμογή σχεδιασμένη ειδικά για την εξάσκηση των γνωστικών λειτουργιών.

1.3 Στόχοι

Ως βασικούς στόχους θα μπορούσαμε να αναγνωρίσουμε τρεις στο πλαίσιο τη διπλωματικής. Αυτοί είναι, πρώτον η απάντηση στο βασικό ερευνητικό ερώτημα του εάν είναι εφικτό να διακρίνουμε εάν ένας

χρήστης της εφαρμογής MCI Rehab, μέσω των δεδομένων επίδοσης που έχουν συλλεχθεί, μπορεί να χαρακτηριστεί ως άτομο με MCI. Δεύτερον, η δημιουργία μιας σειράς μοντέλων, χρησιμοποιώντας διάφορους αλγόριθμους μηχανικής μάθησης, που θα έχουν τη δυνατότητα να κάνουν αυτή τη διάκριση. Τρίτον, πέραν της διερευνητικής ανάλυσης, την ολοκλήρωση μέσω ενός Application Programming Interface (API) για τη χρήση του προτεινόμενου μοντέλου όπως ακριβώς θα συνέβαινε σε ένα παραγωγικό σύστημα, ώστε να θεωρήσουμε πως έχει κλείσει ο βρόγχος της δημιουργίας και της χρήσης των μοντέλων μηχανικής μάθησης για τη συγκεκριμένη έρευνα.

1.3.1 Βασικό ερευνητικό ερώτημα

Η παρούσα διπλωματική έχει ως στόχο να συνεισφέρει στην έρευνα που αφορά την έγκαιρη ανίχνευση της ήπιας γνωστικής εξασθένησης.

Αυτό που αποτελεί κίνητρο για αυτή τη διπλωματική είναι δημιουργία ενός συστήματος το οποίο θα λειτουργεί παράλληλα και συμπληρωματικά με την ήδη υπάρχουσα διαδικασία, συνεισφέροντας έτσι στην έγκαιρη ανίχνευση της MCI.

Ο τρόπος με τον οποίο προτάθηκε να γίνει αυτή η συνεισφορά είναι με τη δημιουργία ενός μοντέλου μηχανικής μάθησης το οποίο θα μπορεί να διαχωρίσει εάν ένα άτομο βρίσκεται στη κατηγορία MCI αναφορικά με τη κλίμακα MOCA.

Κατ' επέκταση, το βασικό ερευνητικό ερώτημα αυτής της διπλωματικής είναι το κατά πόσο είναι εφικτό να δημιουργηθεί ένα τέτοιο μοντέλο με τα δεδομένα που έχουν συλλεχθεί, σε τι επίπεδα κυμαίνεται η απόδοση του και πόσο αξιόπιστο μπορεί να θεωρηθεί για την αξιολόγηση νέων δεδομένων.

1.3.2 Στόχοι Υλοποίησης

Για να μπορέσουμε να θέσουμε συγκεκριμένους στόχους όσον αφορά την υλοποίηση, θα πρέπει να δούμε ποια είναι τα στάδια στα οποία αυτή διαχωρίζεται και ποιο είναι το επιθυμητό αποτέλεσμα των επιμέρους σταδίων.

Το σύνολο της υλοποίησης, περικλείεται σε ένα μεθοδολογικό πλαίσιο, μέσω του οποίου θα εισάγουμε και θα αναλύσουμε τα δεδομένα. Στη συνέχεια θα δημιουργήσουμε μια σειρά δοκιμαστικών μοντέλων, θα εκτελέσουμε βήματα βελτιστοποίησης ενός επιλεγμένου και θα εξάγουμε ένα παραγωγικό μοντέλο. Τέλος, θα δημιουργήσουμε μια υπηρεσία η οποία θα χρησιμοποιεί το παραγωγικό μοντέλο σε νέα δεδομένα.

1.3.2.1 Στόχοι που αφορούν τα δεδομένα

Η πρώτη ομάδα στόχων αφορούν τη διαδικασία Extract, Transform, Load η οποία περιγράφεται στην §3.1 και περιλαμβάνει τα εξής.

- Την κατανόηση των δεδομένων.
- Την εισαγωγή τους από την αρχική τους μορφή σε μια νέα δομή δεδομένων.
- Την εύρεση ενός τρόπου για εύκολη ανάκτηση τους κάθε φορά που θα εκτελούμε τη διαδικασία της διερευνητικής ανάλυσης.

1.3.2.2 Στόχοι που αφορούν τα δοκιμαστικά μοντέλα

Η δεύτερη ομάδα στόχων αφορά το κομμάτι της υλοποίησης που ονομάζεται διερευνητική ανάλυση των δεδομένων, η οποία περιγράφεται στην ενότητα §4.2.3 στο κεφάλαιο της μεθοδολογίας και περιλαμβάνει τη δημιουργία επιμέρους διαδικασιών με στόχο τα παρακάτω.

- Την προ-επεξεργασία των δεδομένων κάθε φορά που θέλουμε να ελέγξουμε νέα δοκιμαστικά μοντέλα.
- Τη διερεύνηση και διαλογή των Features που θα πληρούν τις κατάλληλες προϋποθέσεις όπως περιγράφονται στην ενότητα §4.2.3.3 Feature Selection στο κεφάλαιο της μεθοδολογίας.
- Τη δημιουργία δοκιμαστικών μοντέλων, τη σύγκριση των αποτελεσμάτων τους και την επιλογή εκείνου που θα πληροί τις προϋποθέσεις που τίθενται στην ενότητα §4.2.3.4 Classifier Selection στο κεφάλαιο της μεθοδολογίας.
- Την εύρεση των κατάλληλων παραμέτρων που θα βελτιστοποιούν την απόδοση του επιλεγμένου μοντέλου.

Έχοντας καλύψει τους παραπάνω στόχους βρισκόμαστε σε θέση να εξάγουμε χρήσιμα συμπεράσματα και να απαντήσουμε στο βασικό ερευνητικό ερώτημα. Ωστόσο, ως επιπλέον στόχους που αφορούν την υλοποίηση μπορούμε να θέσουμε τους παρακάτω.

1.3.2.3 Στόχοι που αφορούν τη δημιουργία του παραγωγικού μοντέλου

Όσον αφορά το παραγωγικό μοντέλο, όπως αυτό περιγράφεται αναλυτικά στην ενότητα §4.2.4, ένα μοντέλο δηλαδή το οποίο θα καλούνταν να επεξεργαστεί δεδομένα τα οποία θα εξέταζε για πρώτη φορά, οι στόχοι, είναι οι εξής.

- Θα πρέπει να έχουμε πετύχει όλους τους στόχους των προηγούμενων ενοτήτων.
- Θα πρέπει κατά την εκπαίδευσή του να διασφαλίζεται πως ότι μετασχηματισμός που έχει γίνει στα δεδομένα κατά τη διαδικασία του Preprocessing, έχει συμπεριληφθεί στο παραγωγικό μοντέλο, έτσι ώστε να εφαρμόζεται και στα νέα δεδομένα.
- Θα πρέπει να υπάρχει ένας τρόπος για την εύκολη αποθήκευσή του έτσι ώστε να μπορεί να ανακτηθεί από το Service το οποίο θα κάνει την κατηγοριοποίηση των νέων δεδομένων.
- Ενώ επίσης το παραγωγικό μοντέλο θα πρέπει να διαχειρίζεται το ζήτημα του Data Leakage, με άλλα λόγια να διασφαλίζει πως καμία πληροφορία όσον αφορά το Testing Dataset δε θα περνάει στους Transformers που θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου.

Με τους όρους Training και Testing Dataset αναφερόμαστε στα αντίστοιχα υποσύνολα που προκύπτουν μετά από τον διαχωρισμό του αρχικού Dataset σε δύο επιμέρους υποσύνολα, ώστε να χρησιμοποιήσουμε ένα υποσύνολο για την εκπαίδευση ενός μοντέλου και ένα για την αξιολόγηση της απόδοσης του μοντέλου [37].

Με τον όρο Transformers, αναφερόμαστε σε όλες τις μεθόδους που χρησιμοποιούμε για να μετασχηματίσουμε τα δεδομένα μας. Όλες οι μέθοδοι μετασχηματισμού δεδομένων που χρησιμοποιήθηκαν στο πλαίσιο της διπλωματικής αναλύονται στις επιμέρους υποενότητες της ενότητας §3.3. Λεπτομερέστερη περιγραφή αυτών, αλλά και όλων των υπόλοιπων διαθέσιμων μεθόδων μετασχηματισμού της βιβλιοθήκης Scikit-learn, υπάρχει διαθέσιμη στο Documentation του Scikit-learn [23].

1.3.2.4 Στόχοι που αφορούν την υπηρεσία κατηγοριοποίησης

Τέλος, μια επιπλέον κατηγορία στόχων που μπορούμε να θέσουμε αφορά τη δημιουργία μιας υπηρεσίας, η οποία θα λειτουργεί ως ένα API, μέσω της οποίας θα μπορούμε να κάνουμε χρήση του παραγωγικού μοντέλου για την κατηγοριοποίηση νέων δεδομένων.

Έχοντας κατά νου τον τρόπο με τον οποίο θα θέλαμε να λειτουργεί αυτή η υπηρεσία, οι στόχοι που μπορούμε να θέσουμε είναι οι εξής.

- Η υπηρεσία να έχει τη δυνατότητα ανάκτησης του παραγωγικού μοντέλου.
- Να υπάρχει η δυνατότητα λήψης και επεξεργασίας ενός POST Request.
- Να εκτελείται μια μέθοδος κατηγοριοποίησης και τα αποτελέσματα αυτής να επιστρέφονται στο Response κομμάτι του POST Request.
- Να διερευνηθούν τυχόν προκλήσεις που θα συναντούσαμε στο σενάριο όπου μια τέτοια υπηρεσία θα καλούνταν να λειτουργήσει σε πραγματικές συνθήκες.

1.4 Δομή

Όσον αφορά τη δομή της διπλωματικής, στο κεφάλαιο 1 έχουμε την εισαγωγή όπου περιγράφονται συνοπτικά τα βασικά θεματικά πεδία στα οποία άπτεται η έρευνα που πραγματοποιήθηκε, ενώ επίσης αναλύονται και οι στόχοι που έχουν τεθεί.

Στο κεφάλαιο 2 πραγματοποιείται μια βιβλιογραφική επισκόπηση, όπου μπορεί κανείς να βρει μια σειρά από συνοπτικές αναλύσεις που αφορούν δημοσιεύσεις με συναφές αντικείμενο.

Στη συνέχεια ακολουθεί το κεφάλαιο 3 όπου περιέχεται η ορολογία με χρήσιμες πληροφορίες και επεξηγήσεις σχετικά με την έρευνα, που πιθανώς να βοηθήσουν τον αναγνώστη να αποκτήσει μια καλύτερη εικόνα για κάποιες από τις διαδικασίες.

Το κεφάλαιο 4 αφορά τη μεθοδολογία, όπου αρχικά παρουσιάζονται οι κύριες μεθοδολογίες που υπάρχουν διαθέσιμες για επίλυση προβλημάτων ανάλυσης δεδομένων και χρήση αλγορίθμων μηχανικής μάθησης. Ενώ στη συνέχεια περιγράφεται αναλυτικά η μεθοδολογία που σχεδιάστηκε και ακολουθήθηκε στο πλαίσιο της παρούσας διπλωματικής.

Έπειτα, στο κεφάλαιο 5 της αξιολόγησης των μοντέλων ανίχνευσης MCI που δημιουργήθηκαν κατά τη διάρκεια της έρευνας, μπορεί κανείς να βρει στατιστικά για την απόδοση του εκάστοτε μοντέλου αναλόγως με τον αλγόριθμο που έγινε η εκπαίδευση, τα επιλεγμένα Features καθώς και την μέθοδο βελτιστοποίησης. Ενώ στο αμέσως επόμενο κεφάλαιο 6 αναφέρονται οι προκλήσεις που παρουσιάστηκαν κατά τη διάρκεια της υλοποίησης.

Τέλος στο κεφάλαιο 7 παρατίθενται τα συμπεράσματα από όλη την έρευνα που προηγήθηκε ενώ επισημαίνονται και κάποιες προτάσεις που πιθανώς να παρουσιάζουν ενδιαφέρον.

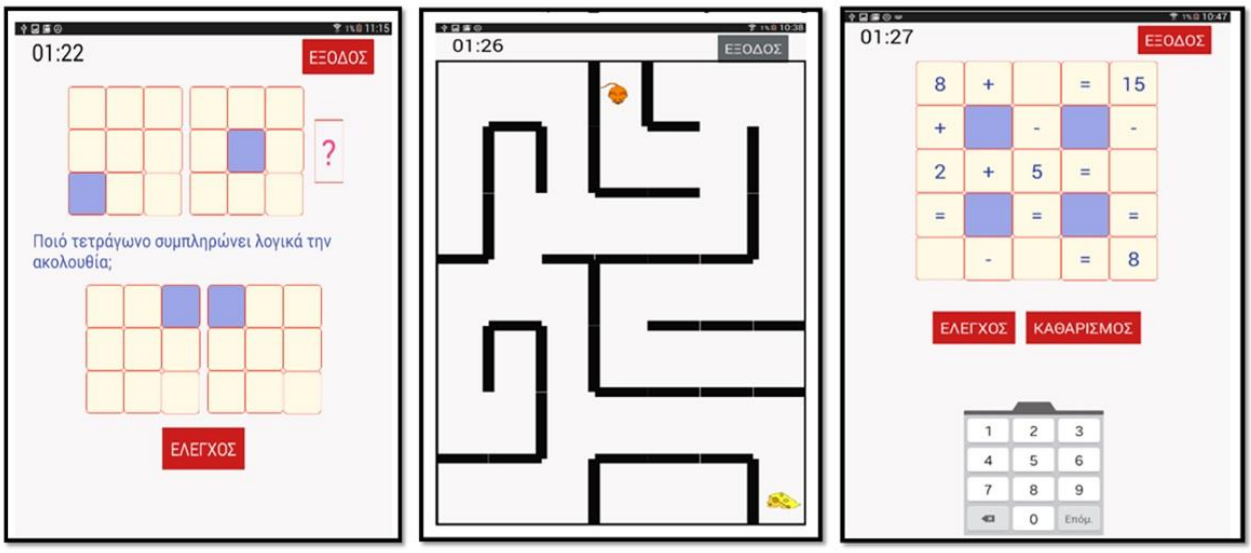
2

Βιβλιογραφική Επισκόπηση

Σε αυτό το κεφάλαιο αρχικά γίνεται αναφορά στην εφαρμογή MCI Rehab, τα δεδομένα της οποίας αποτελούν το αρχικό Dataset της παρούσας διπλωματικής. Στη συνέχεια γίνεται αναφορά σε δημοσιεύσεις από την βιβλιογραφία οι οποίες αναφέρονται σε τουλάχιστον δύο από τα τρία θεματικά πεδία αυτής της διπλωματικής και κατ' επέκταση έχει αξία να δούμε πως ακριβώς προσεγγίζουν το πρόβλημα και τα συμπεράσματα στα οποία καταλήγουν. Τέλος γίνεται παρουσίαση των αποτελεσμάτων αξιολόγησης της νευροψυχολογικής δοκιμασίας MOCA, σε μια σειρά δημοσιεύσεων, για τα Metrics Sensitivity και Specificity.

2.1 MCI Rehab

Το MCI Rehab πρόκειται για μια εφαρμογή Android η οποία περιλαμβάνει 10 παιχνίδια σοβαρού σκοπού, η οποία αναπτύχθηκε στο πλαίσιο διπλωματικής εργασίας και απευθύνεται σε άτομα με συμπτώματα ήπιας γνωστικής εξασθένησης (MCI) [1].



Εικόνα 3 Ενδεικτικά η αντίστοιχη οθόνη που έχει στη διάθεση του ο χρήστης για τα παιχνίδια «Λογική Σειρά», «Λαβύρινθος» και «Υπολογισμός»

Ο σκοπός της εφαρμογής είναι η βελτίωση των γνωστικών λειτουργιών μέσω της χρήσης παιχνιδιών σοβαρού σκοπού. Η εφαρμογή περιλαμβάνει τα παιχνίδια που αναφέρονται στον Πίνακα 1, με τους αντίστοιχους γνωστικούς τομείς στους οποίους εστιάζουν.

Παιχνίδι	**Γνωστικός τομέας**
Παζλ	Προσοχή
Λαβύρινθος	Οπτικοκινητικός
Ανάκληση	Οπτική μνήμη – Ανάκληση
Υπολογισμός	Συγκέντρωση – Μαθηματικός Τομέας
Παρατήρηση	Παρατήρηση
Ταίριασμα Ήχων	Ακουστική μνήμη
Χρονική Σειρά	Χωροχρονικός προσανατολισμός
Γλώσσα	Γλώσσα
Λογική Σειρά	Αντίληψη
Κάρτες Μνήμης	Μνήμης

Πίνακας 1 Λίστα παιχνιδιών της εφαρμογής MCI Rehab με τους αντίστοιχους τομείς γνωστικής επάρκειας

Η έρευνα που πραγματοποιήθηκε είχε ως στόχο να διερευνήσει το κατά πόσο μπορούν τα παιχνίδια σοβαρού σκοπού να συμβάλουν στην βελτίωση των γνωστικών λειτουργιών σε άτομα με συμπτώματα MCI.

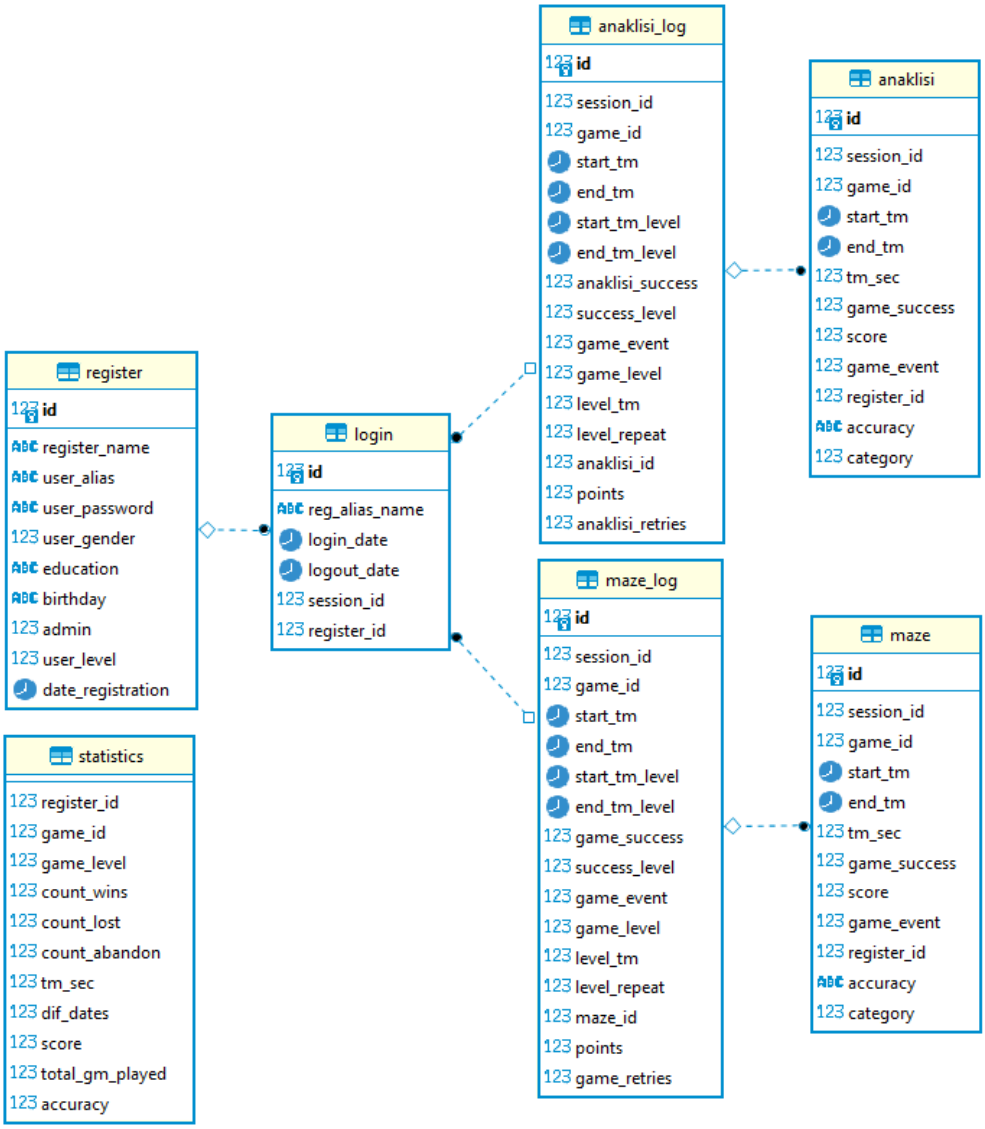
Η έρευνα περιελάμβανε τη χρήση της εφαρμογής από μια ομάδα εστίασης 9 ατόμων με συμπτώματα MCI. Η μέθοδος που ακολουθήθηκε ήταν η αξιολόγηση των χρηστών μέσω των νευροψυχολογικών δοκιμασιών MMSE και MOCA σε δύο στάδια, μια φορά πριν την έναρξη χρήσης της εφαρμογής, και μια φορά στο τέλος της έρευνας. Η διάρκεια της έρευνας ήταν συνολικά 3 μήνες και κάθε ένας από τους 9 συμμετέχοντες είχε στη διάθεση του την εφαρμογή για 10 συνεδρίες των 30 λεπτών.

Τα αρχικά Features που προκύπτουν από τα δεδομένα που έχουν συλλεχθεί και τα οποία έχουμε στη διάθεση μας πριν από την οποιαδήποτε περαιτέρω επεξεργασία των δεδομένων, αναφέρονται στην ενότητα §4.2.1. Ενώ τα Features που προκύπτουν έπειτα από την περαιτέρω επεξεργασία των δεδομένων αναλύονται στην ενότητα §4.2.3.3.1.

Οι τρεις βασικές οντότητες που παρατηρούμε στη διαδικασία που αφορά την εφαρμογή MCI Rehab, είναι ο χρήστης της εφαρμογής (User), μια συνεδρία παιχνιδιών (Game Session) και ένας μεμονωμένος γύρος ενός παιχνιδιού (Game Round) που πραγματοποιείται στο πλαίσιο μιας συνεδρίας. Οι συσχετίσεις μεταξύ αυτών των οντοτήτων είναι, πολλά Game Round προς ένα Game Session και πολλά Game Sessions προς έναν User.

Βάση αυτής της λογικής παρατηρούμε πως είναι δομημένη η βάση δεδομένων, όπου πιο συγκεκριμένα έχουμε τους Users στον πίνακα “register”, τα Game Sessions στον πίνακα “login”, ενώ στη συνέχεια για κάθε Game Session υπάρχει μια δυάδα πινάκων για το κάθε παιχνίδι. Η ονοματολογία αυτών των πινάκων ορίζεται πάντα ως το όνομα του παιχνιδιού για τον πίνακα που αφορά ένα Game Round, ενώ ο δεύτερος πίνακας ορίζεται πάντα ως το όνομα του παιχνιδιού ακολουθούμενο από τη λέξη “_log”, όπου υπάρχουν συγκεντρωτικά στοιχεία για έναν τύπο παιχνιδιού εντός ενός Game Session. Επιπλέον στο σχήμα

υπάρχει και ο πίνακας "statistics" όπου καταγράφονται στατιστικά ομαδοποιημένα ανά χρήστη, είδος παιχνιδιού και επιπέδου δυσκολίας.



Εικόνα 4 Σχήμα της βάσης δεδομένων της εφαρμογής MCI Rehab, όσον αφορά τα παιχνίδια, στην εικόνα βλέπουμε τους πίνακες για την «Ανάκληση» και τον «Λαβύρινθο», ενώ παραλείπονται οι πίνακες των άλλων παιχνιδιών για συντομία καθώς ακολουθούν το ίδιο μοτίβο

Η διαδικασία της μετάπτωσης των δεδομένων από τη συγκεκριμένη βάση δεδομένων σε αυτή που υλοποιήθηκε για τη παρούσα διπλωματική περιγράφεται στην ενότητα §4.2.2.

Η αξιολόγηση της έρευνας πραγματοποιήθηκε σε δύο επίπεδα. Πρώτον, μέσω ενός ερωτηματολογίου ευχρηστίας, που αφορούσε την εμπειρία του χρήστη σχετικά με την εφαρμογή, όπου το συμπέρασμα ήταν πως η υλοποίηση καλύπτει τα επιθυμητά χαρακτηριστικά. Δεύτερον, μέσω της αξιολόγησης του επιπέδου των γνωστικών λειτουργιών των χρηστών, η οποία έγινε με την ανάλυση των στατιστικών στοιχείων που συλλέχθηκαν κατά τη διάρκεια χρήσης της εφαρμογής. Το συμπέρασμα που προέκυψε είναι πως τα παιχνίδια αντικατοπτρίζουν τις αντίστοιχες νευροψυχολογικές δοκιμασίες καθώς οι χρήστες παρουσίασαν, στην εφαρμογή και τις δοκιμασίες, προβλήματα σε αντίστοιχους γνωστικούς τομείς.

Τέλος, αναφορικά με την αξιολόγηση της έρευνας για την εφαρμογή MCI Rehab, αναφέρονται ενδείξεις για το ότι η χρήση μιας εφαρμογής παιχνιδιών σοβαρού σκοπού μπορεί να αποτελέσει μια εναλλακτική των νευροψυχολογικών δοκιμασιών, για τη διάγνωση της MCI.

2.2 The game is the assessment

Στη δημοσίευση με τίτλο «Serious games to assess mild cognitive impairment: 'The game is the assessment'» [8], ο στόχος είναι η διερεύνηση του κατά πόσο θα μπορούσαν τα Serious Games να αποτελέσουν ένα εργαλείο αξιολόγησης γνωστικών λειτουργιών που αφορούν την προσοχή, την αναγνώριση, τη μνήμη και την ανάκληση.

Στο πλαίσιο της συγκεκριμένης δημοσίευσης αναπτύχθηκαν δύο Serious Games με τίτλο WarCAT και Lock Picking αντίστοιχα. Και τα δύο παιχνίδια έχουν ως στόχο να συλλέξουν δεδομένα από την ανάδραση του χρήστη στη προσπάθεια του να επιλύσει το εκάστοτε πρόβλημα.

Το ενδιαφέρον που παρουσιάζει η συγκεκριμένη δημοσίευση είναι η αναφορά στο ότι η έκπτωση σε μια συγκεκριμένη γνωστική λειτουργία μπορεί να έχει επίπτωση σε μια συγκεκριμένη συμπεριφορά ενός ατόμου. Για παράδειγμα, αναφέρεται πως εάν ένα άτομο παρουσιάζει έκπτωση των λειτουργιών της μνήμης και της ανάκλησης τότε αυτό μπορεί να του προκαλέσει αυξημένο άγχος κατά τη διάρκεια μιας αξιολόγησης.

Η πρόταση η οποία γίνεται στη δημοσίευση για την αντιμετώπιση αυτού του φαινομένου, είναι η αναγνώριση από την εφαρμογή της έκπτωσης της συγκεκριμένης γνωστικής λειτουργίας και η αλλαγή στρατηγικής στο παιχνίδι με στόχο η εφαρμογή να προσαρμόζεται δυναμικά στο χρήστη. Επιτυγχάνοντας με αυτό το τρόπο το Gamification της εφαρμογής η οποία ουσιαστικά παραμένει ένα μέσο αξιολόγησης.

2.3 WarCAT

Στη δημοσίευση με τίτλο «A Framework for Utilizing Serious Games and Machine Learning to Classifying Game Play Towards Detecting Cognitive Impairments» [7] παρουσιάζονται τα αποτελέσματα μιας υλοποίησης ενός framework το οποίο έχει ως στόχο να μπορέσει να εκπαιδεύσει ένα μοντέλο για την αναγνώριση τυχόν ήπιας γνωστικής εξασθένησης μέσω της επίδοσης του χρήστη σε ένα παιχνίδι σοβαρού σκοπού. Με άλλα λόγια έχει τον ίδιο στόχο με αυτόν της εφαρμογής MCI Rehab, ωστόσο η προσέγγιση είναι αρκετά διαφορετική.

Η εφαρμογή, η οποία ονομάζεται War CAT, περιλαμβάνει μόλις ένα παιχνίδι με τράπουλα, συγκεκριμένα το War (πόλεμος). Το War είναι ένα σχετικά εύκολο παιχνίδι, στο οποίο στόχος είναι ο παίκτης να συγκεντρώσει όλες τις κάρτες της τράπουλας. Η κάθε κάρτα της τράπουλας αντιστοιχεί σε μια αξία από το 0 έως το 12. Σε κάθε γύρο, οι παίκτες ανοίγουν από ένα φύλλο και ο παίκτης που έχει τη κάρτα με τη μεγαλύτερη αξία κερδίζει το γύρο και παίρνει τη κάρτα του αντιπάλου. Σε περίπτωση ισοβαθμίας ο γύρος επαναλαμβάνεται με τη διαφορά ότι πέρα από το φύλλο το οποίο θα ανοίξει ένας παίκτης επιλέγει και άλλα 3 κλειστά τα οποία θα περιέλθουν στο νικητή του γύρου. Το παιχνίδι ολοκληρώνεται όταν ένας παίκτης συγκεντρώσει και τα 52 φύλλα της τράπουλας.

Η γενική ιδέα της εφαρμογής είναι η δημιουργία μιας μεγάλης βάσης δεδομένων από μοτίβα στρατηγικής παιξίματος του συγκεκριμένου παιχνιδιού με σκοπό το μοντέλο να είναι σε θέση να κατατάξει κάποιον με μεγαλύτερη ακρίβεια μεταξύ των διαφόρων σταδίων της ήπιας γνωστικής εξασθένησης.

Επιπλέον στόχος της υλοποίησης ήταν η δημιουργία αυτής της βάσης δεδομένων από συνθετικά δεδομένα, στη προκειμένη περίπτωση ουσιαστικά από παρτίδες που έχουν προσομοιωθεί. Για την επίτευξη αυτού του στόχου, η ομάδα της έρευνας χρησιμοποίησε αλγορίθμους ενισχυμένης μάθησης Reinforcement Learning καθώς αυτοί οι αλγόριθμοι προσεγγίζουν πολύ τον τρόπο με τον οποίο μαθαίνει ένας άνθρωπος μια νέα εργασία. Με αυτό το σκεπτικό οι ερευνητές δημιούργησαν δύο ομάδες από αυτοματισμούς bots, μια στο ρόλο του ανθρώπου (player bot) για τον οποίο θέλουμε να κάνουμε τη διάγνωση και μια στο ρόλο του υπολογιστή (game bot). Η ουσιαστική διαφορά μεταξύ των δύο bot είναι ότι το game bot εκτελεί επιλογές με βάση συγκεκριμένες και προκαθορισμένες στρατηγικές, ενώ το player bot εκτελεί αρχικά τυχαίες επιλογές λαμβάνοντας μια επιβράβευση κάθε φορά που η επιλογή του οδηγεί σε επιτυχία.

Όπως και στην εφαρμογή MCI Rehab υπάρχει η έννοια της διαφοροποίησης της επιτυχίας, όπου οι πόντοι που κερδίζει ένας παίκτης διαφοροποιούνται από μια προκαθορισμένη εξίσωση η οποία λαμβάνει υπόψη τον χρόνο ολοκλήρωσης ενός γύρου και επιβραβεύει με περισσότερους πόντους τους πιο σύντομους γύρους. Έτσι αντίστοιχα και στην εφαρμογή War CAT υπάρχει η έννοια της διαφοροποίησης της επιβράβευσης όπου ένας χρήστης κερδίζει περισσότερους πόντους σε έναν επιτυχημένο γύρο όταν η διαφορά μεταξύ των δύο φύλλων της τράπουλας είναι μεγάλη.

Όσον αφορά την εκπαίδευση των μοντέλων αναγνώρισης των μοτίβων στρατηγικής η ομάδα χρησιμοποίησε μια σειρά από αλγορίθμους νευρωνικών δικτύων συνέλιξης (convolutional neural network).

Τελικός στόχος του War CAT είναι η δημιουργία μιας βάσης με δεδομένα πραγματικών χρηστών χωρίς προβλήματα γνωστικής επάρκειας, τα οποία θα αποτελούν το σημείο αναφοράς ως οι βέλτιστες πιθανές επιδόσεις. Ενώ παράλληλα θα εμπλουτίζεται η βάση με δεδομένα από τα αποτελέσματα των προσομοιωμένων παρτίδων, τα οποία ουσιαστικά θα ξεκινούν από τις πολύ χαμηλές επιδόσεις που θα έχουν οι Reinforcement Learning αλγόριθμοι στα πρώτα στάδια μέχρι, να φτάσουν τις επιδόσεις των πραγματικών χρηστών, οι οποίες αποτελούν το σημείο αναφοράς.

Θα μπορούσαμε να χαρακτηρίζουμε τη παραπάνω τεχνική ως ένα εμπλουτισμό δεδομένων ή αλλιώς ως Data Augmentation όπως είναι στην ορολογία της μηχανικής μάθησης. Το αποτέλεσμα αυτής της τεχνικής θα είναι μια βάση δεδομένων η οποία θα περιέχει τόσο πραγματικά όσο και πλασματικά δεδομένα, τα οποία από την άποψη των επιδόσεων, θα αντικατοπτρίζονται σε όλες τις στάθμες της γνωστικής επάρκειας.

2.4 Metrics to Monitor Performance

Η συγκεκριμένη δημοσίευση «Metrics to Monitor Performance of Patients with Mild Cognitive Impairment using Computer Based Games» [9] εξετάζει μια σειρά από Metrics, που μπορούν να χρησιμοποιηθούν για τη καταγραφή επιδόσεων των χρηστών με MCI που δοκιμάζονται σε ηλεκτρονικά παιχνίδια, για το προσδιορισμό του επιπέδου της έκπτωσης της γνωστικής τους επάρκειας.

Τα δύο ηλεκτρονικά παιχνίδια, τα οποία δεν αναφέρονται ως παιχνίδια σοβαρού σκοπού, είναι πρώτον το Carleton Sudoku Game (CSG) και δεύτερον το Carleton Word Search Game (CWG), όπου και τα δύο αποτελούν ηλεκτρονικές παραλλαγές των γνωστών παιχνιδιών Sudoku και Σταυρόλεξο αντίστοιχα.

Οι συγγραφείς της δημοσίευσης αναφέρονται αρχικώς στα υπάρχουσες νευροψυχολογικές δοκιμασίες όπως είναι το MOCA και το Repeatable Battery for the Assessment of Neuropsychological Status

(RBANS) και στο γεγονός ότι μπορούν να μετρήσουν το επίπεδο γνωστικής επάρκειας ενός χρήστη μόνο σε συγκεκριμένες χρονικές στιγμές. Αναφέροντας στη συνέχεια τα μειονεκτήματα τους όπως είναι το ότι δεν μπορούν να καταγράψουν τυχόν διακυμάνσεις που μπορεί να έχει το υποκείμενο. Επίσης, πως ακόμη και τις λιγοστές φορές που το υποκείμενο θα συμμετάσχει σε μια δοκιμασία, είναι πολύ πιθανό να μην γίνει σωστή εκτίμηση της κατάστασης λόγο του ότι οι επιδόσεις μπορούν να επηρεαστούν από διάφορους παράγοντες.

Στη συνέχεια γίνεται αντιπαράθεση με τα προτερήματα από τη χρήση ηλεκτρονικών παιχνιδιών σε υπολογιστές και smartphones, ως διαγνωστικά της γνωστικής επάρκειας, τα οποία έχουν τα πλεονεκτήματα της απομακρυσμένης χρήσης καθώς επίσης και της τακτικής πραγματοποίησης των κατάλληλων μετρήσεων.

Ενώ στη συνέχεια αναφέρονται στα Metrics που χρησιμοποιήθηκαν, για να καταγράψουν τις επιδόσεις δύο χρηστών, οι οποίοι έλαβαν μέρος σε έρευνα που πραγματοποίησαν με τη χρήση αυτών των δύο παιχνιδιών, με σκοπό να γίνει μια εκτίμηση της γνωστικής τους επάρκειας.

Πιο συγκεκριμένα, για το Carleton Sudoku Game, τα Metrics είναι τα εξής.

- Ο χρόνος συμπλήρωσης ενός γύρου, σε λεπτά.
- Ο ρυθμός των λαθών (Error Rate) ανά λεπτό, που κάνει ένας χρήστης κατά τη διάρκεια ενός ολοκληρωμένου γύρου.

Ενώ για το Carleton Word Search Game (CWG) τα Metrics που χρησιμοποιήθηκαν είναι τα εξής.

- Ο χρόνος παιχνιδιού (Playtime) για ένα γύρο σε λεπτά.
- Ο αριθμός των λέξεων που βρήκε ο χρήστης ανά παιχνίδι.
- Ο αριθμός των λέξεων που βρήκε ο χρήστης ανά λεπτό (Word Rate)

Και οι δύο συμμετέχοντες στην έρευνα, αξιολογήθηκαν στις δοκιμασίες MOCA και RBANS πριν από την αξιολόγηση τους με τη χρήση των παιχνιδιών, ώστε να υπάρχει ένα σημείο αναφοράς της γνωστικής τους επάρκειας. Ως συμπέρασμα της έρευνας, αναφέρεται ότι τα παραπάνω Metrics βρέθηκαν ικανά να διαχωρίσουν τους συμμετέχοντες μεταξύ τους, ενώ παράλληλα αναφέρεται πως οι βαθμολογίες που καταγράφηκαν στα Metrics συμφωνούν με τη βαθμολογία στις δοκιμασίες MOCA και RBANS.

2.5 Digital Clock Drawing Test

Η δημοσίευση με τίτλο «Machine Learning Analysis of Digital Clock Drawing Test Performance for Differential Classification of Mild Cognitive Impairment Subtypes Versus Alzheimer's Disease» [10] έχει ως στόχο να καθορίσει το εάν και κατά πόσο, αλγόριθμοι μηχανικής μάθησης, είναι σε θέση να εκτελέσουν Classification σε ένα Dataset το οποίο περιέχει τα αποτελέσματα 163 υποκειμένων στη νευροψυχολογική δοκιμασία Digital Clock Drawing Test (dCDT).

Για την δημιουργία των μοντέλων χρησιμοποιήθηκαν Features από τα αποτελέσματα της δοκιμασίας dCDT. Τα μοντέλα όπου εκπαιδεύτηκαν στα πλαίσια της συγκεκριμένης έρευνας ήταν τύπου NN (Neural Networks). Οι πιθανές τιμές του Target Class ορίστηκαν ως οι εξής κατηγορίες.

- Άτομα χωρίς MCI
- Άτομα με συνδυαστικής μορφής MCI μεταξύ των υποκατηγοριών του MCI (aMCI, mxMCI)
- Άτομα με MCI αποκλειστικά της μορφής Amnestic MCI (aMCI)
- Άτομα με AD.

Στη συνέχεια γίνεται αναφορά σε μια σειρά από πειράματα που εκτελέστηκαν, για το καθένα από τα οποία, το ερώτημα ήταν η απόδοση του μοντέλου στο Classification μεταξύ των παραπάνω κατηγοριών.

Η διαδικασία περιελάβανε τα στάδια της προ-επεξεργασίας των δεδομένων, της επιλογής των κατάλληλων Features, της παραγωγής επιπλέον συνθετικών δεδομένων για την εξισορρόπηση των κατηγοριών του Target Class με τον αλγόριθμο SMOTE και τέλος την εκπαίδευση των μοντέλων, τα οποία και εκτελέστηκαν με αυτή τη σειρά.

Ενδιαφέρον παρουσιάζει η μέθοδος που χρησιμοποιήθηκε για την εξισορρόπηση των κατηγοριών του Target Class η οποία είναι η μέθοδος Synthetic Minority Oversampling Technique (SMOTE) [10] την οποία εξετάζουμε και στη παρούσα διπλωματική για τον ίδιο σκοπό.

Στα συμπεράσματα της έρευνας αναφέρεται πως η χρήση μοντέλων μηχανικής μάθησης μπορεί να χρησιμοποιηθεί για να διακρίνει εάν το γνωστικό επίπεδο ενός ατόμου βρίσκεται στο φάσμα της ήπιας έως μέτριας γνωστικής διαταραχής. Ενώ παράλληλα αναγνωρίζονται προκλήσεις που έχουν να κάνουν με τη πιθανότητα να υπάρχει κάποιο Bias στα αποτελέσματα, πρώτον λόγω της μη διαφοροποίησης όσον αφορά την εθνικότητα και δεύτερον, λόγω της ύπαρξης περισσότερων ατόμων με AD μεταξύ όσων συμμετείχαν στην έρευνα.

2.6 Καταγεγραμμένη ακρίβεια αναγνώρισης MCI

Στη συνέχεια παρουσιάζεται μια σύνοψη όσον αφορά σε μια σειρά από δημοσιεύσεις οι οποίες παρουσιάζουν συγκεκριμένες τιμές για τα Metrics Sensitivity και Specificity, ανάλογα τη τιμή Cutoff που χρησιμοποιείται στη διάκριση της κατηγορίας MCI, για τη νευροψυχολογική δοκιμασία MOCA.

Από έρευνα, που παρουσιάστηκε στο 37^{ο} ετήσιο επιστημονικό συνέδριο “Integrating Care, Making an Impact”, παρατηρούμε ότι για τιμή Cutoff: 26, η οποία είναι η επίσημα δημοσιευμένη τιμή Cutoff [18], τα ποσοστά αναγνώρισης γνωστικής εξασθένησης είναι, για το Metric Sensitivity: 95% και για το Metric Specificity: 54%, ενώ η ιδανική τιμή Cutoff: 24 βάση της έρευνας που παρουσιάστηκε ήταν 24, για την οποία τα ποσοστά των Metric είχαν ως εξής, Sensitivity: 93% και Specificity: 78% [38].

Επίσης από έρευνα η οποία είχε ως στόχο να αξιολογήσει την εγκυρότητα της δοκιμασίας MOCA, σε πληθυσμό συγκεκριμένης εθνικότητας και ηλικιακής ομάδας, παρατηρούμε τα εξής ποσοστά. Για κατηγοριοποίηση σε 3 διαφορετικές κατηγορίες, NC, MCI, AD, με τιμές Cutoff για MCI από 23 μέχρι και 24, για το Metric Sensitivity: 95% ενώ για το Metric Specificity: 63% [39].

Σε ακόμη μια έρευνα σύγκρισης νευροψυχολογικών δοκιμασιών, όσον αφορά στη δοκιμασία MOCA, για τιμή Cutoff: 26, αναφέρονται τα εξής ποσοστά, για το Metric Sensitivity: 96%, ενώ για το Metric Specificity: 58% [40].

Μια ακόμη έρευνα για την εγκυρότητα της δοκιμασίας MOCA και συγκεκριμένα για τη Κορεάτικη εκδοχή της δοκιμασίας, για τιμή Cutoff: 22/23 αναφέρονται τα εξής ποσοστά, για το Metric Sensitivity: 89% ενώ για το Metric Specificity: 84% [41].

3

Ορολογία

Σε αυτό το κεφάλαιο παρουσιάζονται κάποιοι βασικοί όροι που συναντώνται στο πλαίσιο της διπλωματικής και είναι σημαντική η αναφορά και η επεξήγηση τους. Μια σημαντική διευκρίνηση σε αυτό το σημείο είναι ότι πιθανόν κάποιοι από τους παρακάτω όρους να απαντώνται και σε άλλα επιστημονικά πεδία πέραν του πεδίου της μηχανικής μάθησης, ωστόσο οι επεξηγήσεις που παρουσιάζονται παρακάτω επικεντρώνονται στη σημασία που έχουν αναφορικά πάντα με το πεδίο της μηχανικής μάθησης. Επίσης, μια ακόμη σημαντική διευκρίνηση που αφορά την ορολογία είναι πως, σε όλη την έκταση της διπλωματικής, όποτε το επίκεντρο της αναφοράς είναι κάτι σχετικό με το ιατρικό κομμάτι τα άτομα αναφέρονται ως υποκείμενα κατά την ιατρική ορολογία, ενώ όποτε γίνεται αναφορά σε κάτι που σχετίζεται με τα παιχνίδια σοβαρού σκοπού ή το πεδίο της μηχανικής μάθησης τα άτομα αναφέρονται ως χρήστες.

3.1 Extract, Transform, Load

Με τον όρο Extract, Transform, Load (ETL), αναφερόμαστε σε όλα εκείνα τα απαραίτητα βήματα που πρέπει να γίνουν ώστε να συγκεντρώσουμε τα δεδομένα από τις εξωτερικές πηγές όπου βρίσκονται στην αρχική τους μορφή (Raw Data), σε μία δομή από την οποία θα μπορούμε να τα ανακτήσουμε για περαιτέρω ανάλυση. Πιο συγκεκριμένα, οι επιμέρους όροι σημαίνουν τα εξής.

- Extract: Εξαγωγή των δεδομένων από την αρχική πηγή.
- Transform: Προετοιμασία μετάπτωσης δεδομένων σε μια νέα δομή.
- Load: Ανάκτηση των δεδομένων για χρήση σε διαδικασίες ανάλυσης και μοντελοποίησης.

Σχετικά με την εξαγωγή, τα σημαντικά στοιχεία είναι να γνωρίζουμε τόσο το Format στο οποίο είναι αποθηκευμένα τα δεδομένα, όσο και τις συσχετίσεις μεταξύ τους. Για παράδειγμα, θα μπορούσαμε να είχαμε τα δεδομένα αποθηκευμένα σε αρχεία CSV σε κάποιο δικτυακό Repository, ή σε κάποια βάση δεδομένων. Αντίστοιχα η νέα δομή στην οποία θα θέλαμε να κάνουμε τη μετάπτωση θα μπορούσε να είναι είτε μια νέα βάση δεδομένων είτε κάποιας μορφής αρχεία όπως π.χ. MS Access, TXT ή ARFF, αναλόγως με το Software που θα θέλαμε να χρησιμοποιήσουμε για να αναλύσουμε τα δεδομένα.

Όσον αφορά το κομμάτι του Transformation στη διαδικασία του ETL, αυτό δεν έχει να κάνει με τα Transformations των τιμών των πεδίων που θα δούμε στη συνέχεια στη διαδικασία της EDA, αλλά έχει να κάνει με τη μεταβολή των εγγραφών μιας βάσης δεδομένων για τη μετάπτωση από το παλιό στο νέο Schema.

Στις βάσεις δεδομένων κάθε μεταβολή θεωρείται ως μια μετάπτωση δεδομένων σε ένα νέο σχήμα (Database Migration) ακόμη και όταν αυτή η μεταβολή περιλαμβάνει μια απλή προσθήκη ενός νέου πεδίου σε έναν πίνακα. Οπότε η μεταφορά των δεδομένων σε ένα νέο Schema, θεωρείται πάντα μια καλή ευκαιρία για βελτιώσεις που αφορούν το Database Normalization.

Επιπλέον, η μετάπτωση των δεδομένων από πολλά Schemas σε ένα ενιαίο προσφέρει τα πλεονεκτήματα του Consistency και του Integrity των δεδομένων.

Όσον αφορά το Consistency, το πρόβλημα είναι ότι έχοντας πολλαπλά Schemas η κάθε αλλαγή που θέλουμε να κάνουμε, για παράδειγμα η προσθήκη ενός πεδίου σε έναν πίνακα, ενέχει τον κίνδυνο να μην πραγματοποιηθεί επιτυχώς σε όλα τα Schemas. Οπότε διατηρώντας ένα ενιαίο Schema διασφαλίζουμε την Consistency της της βάσης δεδομένων μιας και πλέον έχουμε ένα συγκεκριμένο σημείο στο οποίο γίνονται όλες οι τροποποιήσεις.

Ενώ όσον αφορά το Integrity, το πρόβλημα που προκύπτει διατηρώντας πολλαπλά Schemas, είναι ότι το σύνολο της πληροφορίας για έναν πίνακα βρίσκεται αποθηκευμένο σε πολλαπλά σημεία, ο πίνακας είναι με άλλα λόγια Partitioned μεταξύ διαφόρων Schemas, συνεπώς τα όποια Foreign Keys χρησιμοποιούσαμε θα περιοριζόταν στο εκάστοτε Schema. Πραγματοποιώντας τη μετάπτωση σε ένα ενιαίο Schema το όφελος είναι ότι πλέον μπορούμε να χρησιμοποιήσουμε Foreign Keys και να είμαστε σίγουροι ότι καλύπτουμε το εύρος όλων των δεδομένων, διασφαλίζοντας έτσι το Integrity των δεδομένων.

Τέλος, όσον αφορά το Load, αυτό περιλαμβάνει τις διαδικασίες για την ανάκτηση των δεδομένων από τη νέα δομή, στο Software όπου θα πραγματοποιηθεί η ανάλυση των δεδομένων. Αυτό διαφέρει κατά περίπτωση, ωστόσο υπάρχουν δύο κύριοι τρόποι με τους οποίους γίνεται και είναι είτε με την επιλογή της διαδρομής όταν πρόκειται για αρχεία, είτε με την εγκατάσταση των απαραίτητων Drivers και τη δημιουργία μιας σύνδεσης όταν πρόκειται για βάση δεδομένων.

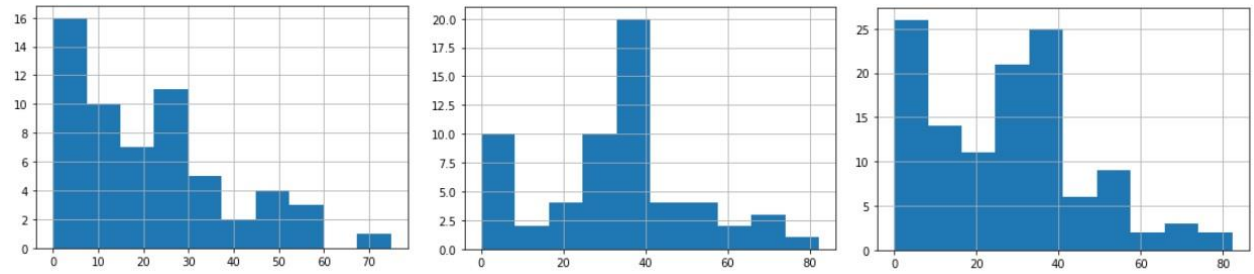
3.2 Concept Drift

Αρχικά ας αναφερθούμε στο τι ακριβώς είναι και πότε συμβαίνει. Όπως περιγράφεται στην ενότητα §4.2.4, δημιουργούμε ένα μοντέλο το οποίο πλέον θεωρούμε ότι μπορεί να χρησιμοποιηθεί σε ένα παραγωγικό σύστημα. Αυτό σημαίνει πως από τη πρώτη κιόλας νέα εγγραφή, μέρος ενός νέου Dataset, η κατανομή του κάθε Feature δυνητικά μπορεί να αρχίσει να μεταβάλλεται. Εάν πράγματι παρατηρηθεί αυτό το φαινόμενο, εάν δηλαδή αλλάξει δραστικά η κατανομή των τιμών ενός Feature τότε πλέον έχουμε να κάνουμε με Concept Drift, εναλλακτικά στη βιβλιογραφία αναφέρεται και ως Domain Shift ή Domain Drift [15].

Στο σενάριο λοιπόν, όπου με τη χρήση του μοντέλου σε νέα δεδομένα παρατηρήσουμε Concept Drift σε κάποια Features, εάν έχουμε χρησιμοποιήσει τα συγκεκριμένα Features που εμφανίζουν αυτή τη τάση για την εκπαίδευση του μοντέλου, τότε είναι φυσικό επακόλουθο το μοντέλο να παρουσιάσει μείωση της απόδοσης, κάνοντας λάθος εκτιμήσεις. Για αυτό το πρόβλημα υπάρχουν πολλές μέθοδοι που μπορούμε να ακολουθήσουμε ως λύση (Adaptation Methods) και η διαδικασία γενικότερα αναφέρεται ως Domain Adaptation. Αναλόγως τη μέθοδο που θα επιλέξουμε μπορούμε να έχουμε επανεκπαίδευση του μοντέλου ή απλή προσαρμογή αυτού.

Η διαδικασία για την ανίχνευση του Concept Drift κατά τη διάρκεια της χρήσης ενός μοντέλου σε ένα παραγωγικό σύστημα αναφέρεται στη βιβλιογραφία ως Sequential Analysis [15].

Ως παράδειγμα από τα δεδομένα μας, μπορούμε να δούμε το Feature του μέσου χρόνου ενός κερδισμένου γύρου σε ένα Session. Πιο συγκεκριμένα, βλέπουμε στο πρώτο ιστόγραμμα ότι για τις 59 πρώτες εγγραφές η κατανομή είναι Right Skewed, για τις 60 τελευταίες η κατανομή δείχνει κανονική, ενώ για όλες τις εγγραφές μαζί στο τρίτο ιστόγραμμα, δείχνει Bimodal. Θεωρητικά, αντίστοιχο Concept Drift, με αυτό που βλέπουμε μεταξύ των δύο πρώτων ιστογραμμάτων, θα μπορούσαμε να έχουμε μεταξύ του τρέχοντος Dataset και ενός μελλοντικού.



Εικόνα 5 Μέσος χρόνος κερδισμένου γύρου. Πρώτες 59, τελευταίες 60 και σύνολο εγγραφών.

3.3 Transformations

3.3.1 Encoding

Μια από τις πιο σημαντικές διαδικασίες στο πλαίσιο του Preprocessing των δεδομένων είναι το Encoding των τιμών του Dataset. Με τον όρο Encoding αναφερόμαστε στην μετατροπή των Categorical Features σε Numerical. Στην ουσία πρόκειται για μια διαδικασία αντιστοίχισης των Categorical Features που έχουν αλφαριθμητικό τύπο σε αριθμητικές τιμές. Ο λόγος για τον οποίο πρέπει τα δεδομένα να καταλήξουν σε Numerical μορφή είναι διότι οι περισσότεροι αλγόριθμοι της βιβλιοθήκης Scikit-learn απαιτούν τα δεδομένα σε Numerical μορφή. Επιπλέον με αυτό το τρόπο μπορούμε να εξοικονομήσουμε αποθηκευτικό χώρο στη βάση δεδομένων.

Κατά τη διαδικασία του Encoding, θα πρέπει να διασφαλίσουμε ότι με τον τρόπου που θα γίνει, θα έχουμε τη δυνατότητα να πράξουμε τα εξής:

Πρώτον, ανεξαρτήτως του πως θα εκτελέσουμε το Encoding, θα πρέπει να υπάρχει η δυνατότητα για Reverse Transformation, το οποίο σημαίνει να έχουμε φροντίσει για να υπάρχει η δυνατότητα μετάφρασης τις τιμής ενός Numerical πλέον Feature στο αντίστοιχο αρχικό αλφαριθμητικό Label.

Δεύτερον, θα πρέπει να έχουμε καλή γνώση σχετικά με το τι αντιπροσωπεύει το κάθε Feature που θα περάσει από τη διαδικασία του Encoding, καθώς αυτό θα καθορίσει με ποιο τρόπο θα γίνει. Για παράδειγμα εάν κάποιο Feature είναι τύπου είναι Ordinal Categorical, δηλαδή τα Labels του έχουν κάποιου είδους ιεραρχία, π.χ. {“Ποτέ”, “Σχεδόν ποτέ”, “Μερικές φορές”, “Συχνά”, “Συνέχεια”}, τότε μπορούμε να εφαρμόσουμε ένα βασικό Label Encoding ώστε το αρχικό Set από Labels να έρθει στη μορφή {0, 1, 2, 3, 4, 5}, χωρίς να μεταβληθεί με τον οποιοδήποτε τρόπο η σημαντικότητα του εκάστοτε Label. Ωστόσο, αυτό δεν θα μπορούσε να εφαρμοστεί και σε Non-Ordinal Categorical Features, διότι θα δημιουργούσε μια τεχνητή ιεραρχία εκεί που δεν υπάρχει. Για παράδειγμα, στο Feature που αφορά την οικογενειακή κατάσταση, όπου τα πιθανά Labels είναι {“Άγαμος”, “Έγγαμος”, “Διαζευγμένος”, “Χήρος”}, δεν θα θέλαμε ο αλγόριθμος εκπαίδευσης του μοντέλου να προσδώσει κάποιο βάρος σε μια από τις 4 οικογενειακές καταστάσεις. Η λύση με την οποία θα διαχειριστούμε τα Non-Ordinal Categorical Features είναι το One-Hot-Encoding, όπου

ουσιαστικά δημιουργούμε μια στήλη για κάθε Label και οι τιμές που μπορούν να πάρουν πλέον αυτά τα τεχνητά δημιουργημένα Features είναι Binary {0, 1}.

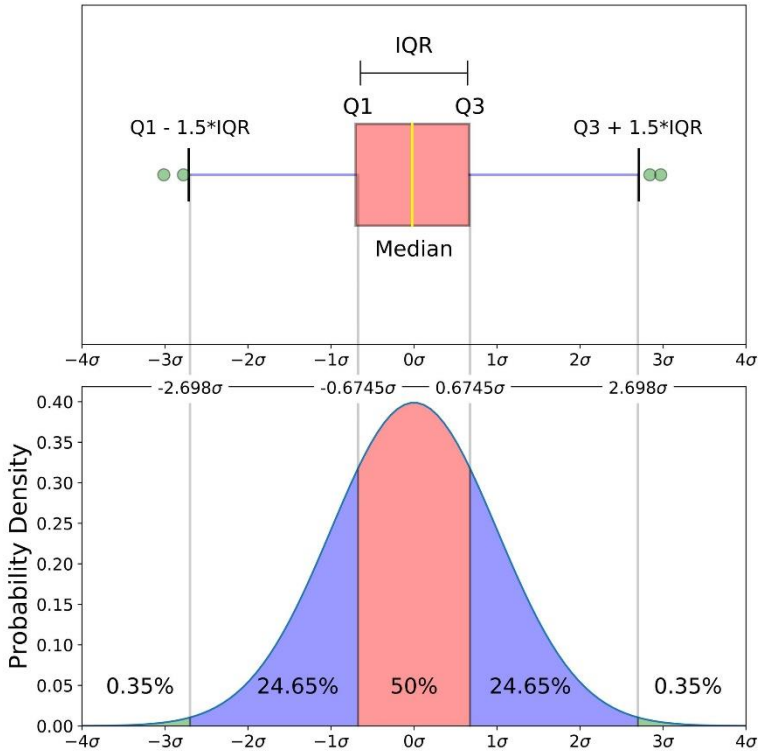
Τρίτον, θα πρέπει να διασφαλίσουμε πως τα νέα δεδομένα τα οποία θα φτάνουν στο Classification Service θα πρέπει είτε να είναι ήδη Encoded είτε να περνάνε από τις αντίστοιχες διαδικασίες Encoding που έχουν γίνει στα πλαίσια της εκπαίδευσης του τελικού μοντέλου.

3.3.2 Outliers

Με τον όρο Outliers περιγράφουμε τις ακραίες τιμές μιας κατανομής ενός Feature. Πιο συγκεκριμένα, μια τιμή θεωρείται Outlier όταν αυτή είναι είτε μικρότερη από το Minimum, είτε μεγαλύτερη από το Maximum της κατανομής του Feature. Όπου όπως φαίνεται και στο διάγραμμα Boxplot μιας κανονικής κατανομής στην Εικόνα 4, το Minimum και το Maximum υπολογίζονται από τις ακόλουθες εξισώσεις αντίστοιχα.

$$\text{Minimum} = Q1 - 1.5 * IQR$$

$$\text{Maximum} = Q3 + 1.5 * IQR$$



Εικόνα 6 Αποτύπωση των ποσοτικών (Quantile) στατιστικών σε μια Binomial κατανομή.

Σχετικά με τα Q1, Q2 και Q3, αυτά αποτελούν στοιχεία της περιγραφικής στατιστικής. Πιο συγκεκριμένα το Q1 είναι το πρώτο Quartile ή όπως αλλιώς αναφέρεται το 25ο Percentile, ενώ αντίστοιχα το Q3 είναι το τρίτο Quartile ή αλλιώς και 75ο Percentile. Ενώ το Q2 ή αλλιώς 50ο Percentile αντιπροσωπεύει τη διάμεσο (Median) τιμή της κατανομής. Όσον αφορά το IQR (Interquartile Range) αυτό αντιπροσωπεύει τη κατανομή που βρίσκεται μεταξύ των Q1 και Q3.

Αναφορικά με τα Percentiles, αποτελούν και αυτά μέρος της περιγραφικής στατιστικής και μας δείχνουν ουσιαστικά το που βρίσκεται μια τιμή συγκριτικά με τις υπόλοιπες τιμές μιας κατανομής. Για

παράδειγμα, το 75ο Percentile, το οποίο ισοδυναμεί με το Q3, μας δείχνει το 25% των δειγμάτων που έχουν τιμή μεγαλύτερη από την τιμή που βρίσκεται στο Q3.

Όλες οι παραπάνω πληροφορίες μας βοηθούν να κατανοήσουμε ποιες τιμές είναι Outliers για μια κατανομή και πως υπολογίζονται. Η διαχείριση των Outliers, είτε με την αντικατάσταση μιας τιμής, είτε με την αφαίρεση της εγγραφής που περιέχει Outlier για ένα ή παραπάνω Features, είναι μια προαιρετική διαδικασία και η χρησιμότητά της είναι η διασφάλιση ότι κατά το Scaling (βλέπε §3.3.4) δεν θα αλλοιωθεί η κατανομή. Η διαδικασία είναι προαιρετική διότι για την εφαρμογή του Scaling υπάρχουν αλγόριθμοι οι οποίοι λειτουργούν εξίσου καλά ακόμη και όταν στην κατανομή ενός Feature υπάρχουν Outliers.

3.3.3 Discretization

Με τον όρο Discretization αναφερόμαστε στη διαδικασία όπου για ένα Feature τύπου Numerical με συνεχείς τιμές εφαρμόζουμε ένα μετασχηματισμό ώστε οι τιμές να γίνουν διακριτές. Η διαδικασία πολλές φορές συναντάται και με τον όρο Binning.

Ανεξαρτήτως της υλοποίησης υπάρχουν τρεις βασικές επιλογές όσον αφορά τη μέθοδο που θα επιλέξουμε για να ορίσουμε τις στάθμες των διακριτών τιμών, αυτές είναι:

- Equal Width Discretization. Σε αυτή τη περίπτωση γνωρίζοντας το Range τιμών ενός Feature καθώς και το πλήθος των επιθυμητών διακριτών τιμών, μπορούμε να ορίσουμε ισαπέχουσες στάθμες.
- Equal Frequency Discretization. Σε αυτή τη περίπτωση, ορίζουμε τις στάθμες με τέτοιο τρόπο έτσι ώστε η κάθε στάθμη να έχει ίσο αριθμό δειγμάτων.
- Custom Discretization. Σε πολλές περιπτώσεις, όσον αφορά τις στάθμες, έχει νόημα να επιλέξουμε το πλήθος τους είτε αυθαίρετα για να κάνουμε δοκιμές, είτε με βάση το Business Logic του προβλήματος που προσπαθούμε να επιλύσουμε.

3.3.4 Scaling

Η διαδικασία του Scaling είναι η τελευταία ενέργεια τύπου του Preprocessing στο πλαίσιο της Exploratory Data Analysis βάση της μεθοδολογίας που έχουμε ορίσει για τη διπλωματική. Η διαδικασία πολλές φορές συναντάται και με την ονομασία Standardization ή Normalization. Αυτό έχει να κάνει με τον αλγόριθμο που χρησιμοποιούμε για το Transformation των τιμών. Σε αυτή την ενότητα θα δούμε ποιο είναι το κοινό χαρακτηριστικό αυτών των δύο μεθόδων και ποιο είναι το όφελος από αυτή τη διαδικασία.

Το Scaling είναι ουσιαστικά ένα Transformation που εφαρμόζουμε στα δεδομένα, εφαρμόζοντας έναν από τους πολλούς διαθέσιμους αλγόριθμους, με στόχο οι τιμές των ανεξάρτητων μεταβλητών να καταλήξουν να έχουν μια σειρά από ίδια χαρακτηριστικά, όπως για παράδειγμα να βρίσκονται εντός ενός συγκεκριμένου Range τιμών, να έχουν ως μέση τιμή το 0, να έχουν ίδιο Standard Deviation και ούτω καθεξής.

Ο βασικός λόγος για τον οποίο θέλουμε τις τιμές στην ίδια κλίμακα είναι έτσι ώστε το μοντέλο να μην μπορεί δώσει βαρύτητα σε κάποιο Feature μόνο και μόνο επειδή το Range των τιμών του κυμαίνεται σε υψηλές τιμές συγκριτικά με άλλα Features.

Εξίσου σημαντικός λόγος για εφαρμογή Scaling και πιο συγκεκριμένα της τεχνικής του Standardization, είναι η αποφυγή των επιπτώσεων του Concept Drift. Ουσιαστικά, όπως περιγράφεται και

στην ενότητα του Concept Drift, σε ένα παραγωγικό σύστημα, για κάποια Features, η κατανομή των τιμών μπορεί να αλλάζει σε μεγάλο βαθμό με κάθε νέο Dataset. Οπότε αυτό που μπορούμε να κάνουμε είναι, για κάθε νέο Dataset, κατά τη διάρκεια της EDA διαδικασίας, η οποία στο πλαίσιο της διπλωματικής περιγράφεται στην ενότητα §4.3.2, να εφαρμόζουμε Standardization ώστε να ελαχιστοποιούμε την αρνητική επίδραση στην απόδοση του μοντέλου που μπορεί να έχει μια μεταβολή της κατανομής τιμών ενός Feature [24].

Ένας ακόμη λόγος για τον οποίο θέλουμε τις τιμές των Feature στην ίδια κλίμακα τιμών είναι διότι στο πλαίσιο της EDA πολλές φορές βασιζόμαστε στο Visualization των δεδομένων για να εξάγουμε χρήσιμα συμπεράσματα για τα δεδομένα και τη μεταξύ τους συσχέτιση, κάτι το οποίο είναι αρκετά δύσκολο όταν οι τιμές βρίσκονται σε διαφορετικές κλίμακες.

Επίσης, το Scaling αποτελεί βασική προϋπόθεση για να έχουν καλή απόδοση οι αλγόριθμοι που βασίζονται σε Distance Metrics, είτε πρόκειται για Classification, είτε για Regression, είτε για Clustering. Όπου Distance Metrics, είναι ουσιαστικά οι μέθοδοι που χρησιμοποιεί ο εκάστοτε αλγόριθμος για τον υπολογισμό της απόστασης μεταξύ σημείων του Dataset, χαρακτηριστικότερο παράδειγμα είναι ο αλγόριθμος K-nearest Neighbors.

Βασική λεπτομέρεια που θα πρέπει επίσης να προσέχουμε στην επιλογή της μεθόδου που θα χρησιμοποιήσουμε για να εφαρμόσουμε Scaling είναι η ύπαρξη Outliers στο Dataset. Αυτό διότι εάν για παράδειγμα υπάρχουν και δεν έχουμε διαχειριστεί αυτές τις ακραίες τιμές, τότε εφαρμόζοντας για παράδειγμα Normalization, δηλαδή έναν αλγόριθμο ο οποίος μετατρέπει τις τιμές σε ένα συγκεκριμένο Range, το μεγαλύτερο μέρος των τιμών θα μετατοπιζόταν προς τη μέση τιμή αλλοιώνοντας έτσι την κατανομή.

3.3.5 Dimensionality Reduction

Με τον όρο Dimensionality Reduction, αναφερόμαστε σε μια ομάδα από μεθόδους τύπου Feature Extraction. Υπάρχουν πολλές μέθοδοι που μπορούν να χρησιμοποιηθούν για Dimensionality Reduction, ωστόσο οι πλέον γνωστές είναι οι εξής, Principal Components Analysis (PCA), LDA, t-SNE, UMap.

Η τεχνική PCA είναι μια από τις τεχνικές μετασχηματισμού των δεδομένων ενός Dataset σε ένα Dataset λιγότερων διαστάσεων. Όπου οι διαστάσεις ενός Dataset, είναι στην ουσία το πλήθος των μεταβλητών του.

Το αποτέλεσμα της τεχνικής PCA είναι ο μετασχηματισμός ενός συνόλου συσχετισμένων μεταβλητών σε ένα σύνολο από νέες μεταβλητές που ονομάζονται Principal Components (PC) και έχουν πλέον μεταξύ τους χαμηλό Correlation. Επίσης, όσον αφορά τις νέες αυτές μεταβλητές PC, έχουν το χαρακτηριστικό πως οι πρώτες από αυτές διατηρούν και το μεγαλύτερο μέρος του Variance του Dataset. Κάτι το οποίο μπορεί να μεταφραστεί και ως ότι οι πρώτες PC μεταβλητές είναι στην ουσία μια γραμμική συσχέτιση, των αρχικών μεταβλητών του Dataset, η οποία περιγράφει το μεγαλύτερο μέρος του Variance του Dataset. Η τεχνική PCA θεωρείται ως μια Unsupervised τεχνική, κάτι το οποίο την καθιστά χρήσιμη σε προβλήματα Clustering [34].

3.4 Feature Variance

Όταν αναφερόμαστε στο Variance μιας κατανομής ενός Feature, εννοούμε την απόκλιση που μπορεί να έχει μια τυχαία τιμή της κατανομής και ως γενικός κανόνας ισχύει ότι υπολογίζεται ως το τετράγωνο της απόκλισης της τιμής από τη μέση τιμή (Mean) της κατανομής. Ο βασικός τύπος υπολογισμού του Variance ενός Feature με κανονική κατανομή είναι ο παρακάτω, ωστόσο αυτός διαφέρει αναλόγως με το είδος της κατανομής [26].

$$var(x) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

3.5 Covariance & Correlation

Τόσο το Covariance όσο και το Correlation, είναι δείκτες που μας βοηθούν να καταλάβουμε τη συσχέτιση μεταξύ δύο μεταβλητών, στη προκειμένη περίπτωση μεταξύ δύο Features ή μεταξύ ενός Feature και του Target Class.

Όσον αφορά το Covariance, αυτό μας δείχνει ουσιαστικά το πρόσημο και το μέγεθος της συσχέτισης, εάν για παράδειγμα με την αύξηση της τιμής ενός Feature έχουμε αύξηση ή μείωση σε ένα δεύτερο Feature το οποίο εξετάζουμε και κατά πόσο. Επίσης, εάν υπολογίζαμε το Covariance ενός Feature ως προς τον εαυτό του τότε θα είχαμε ουσιαστικά το Variance του Feature. Το Covariance προκύπτει από τον παρακάτω τύπο.

$$cov(x,y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

Όσον αφορά το Correlation, πρόκειται για μια Normalized εκδοχή του Covariance, η οποία αναφέρεται και ως “Pearson's correlation coefficient”. Ο όρος Normalized εδώ χρησιμοποιείται γιατί ουσιαστικά πρόκειται για τον λόγο (Ratio) μεταξύ του Covariance προς το γινόμενο του Standard Deviation των δύο Features που εξετάζουμε. Το εύρος τιμών του Correlation είναι από -1 έως 1. Όπου το μέγεθος της απόλυτης τιμής μας δείχνει το πόσο ισχυρή είναι η συσχέτιση, ενώ το πρόσημο μας δείχνει τη κατεύθυνση της συσχέτισης όπως ακριβώς συμβαίνει και με το Covariance. Το Correlation εκφράζεται με τον παρακάτω τύπο.

$$r_{xy} = cor(x,y) = \frac{cov(x,y)}{\sigma_x \sigma_y}$$

Δύο σημαντικές λεπτομέρειες αναφορικά με το Correlation είναι οι εξής. Πρώτον, μια ακραία τιμή Correlation, είτε κοντά στο 1 είτε κοντά στο -1, δεν σημαίνει αυτομάτως ότι υπάρχει αιτιότητα μεταξύ των Features. Με άλλα λόγια δε σημαίνει απαραίτητα ότι μια διακύμανση στο ένα από τα δύο Features προκαλεί διακύμανση και στο άλλο. Ουσιαστικά δύο Features με υψηλό μεταξύ τους Correlation, μπορεί πράγματι να εξαρτώνται άμεσα μεταξύ τους. Εναλλακτικά, μπορεί να είναι ανεξάρτητα μεταξύ τους και να υπάρχει ένας τρίτος κοινός παράγοντας που τα επηρεάζει. Ενώ επίσης υπάρχει και η περίπτωση να πρόκειται για μια τυχαία τάση. Δεύτερον, με τον παραπάνω τύπο του Correlation μπορούμε να εξετάσουμε μόνο τη γραμμική συσχέτιση μεταξύ δύο Features [26].

3.6 Metrics

Με τον όρο Metrics περιγράφουμε τα στατιστικά εκείνα μεγέθη τα οποία χρησιμοποιούμε για να αξιολογήσουμε την απόδοση ενός μοντέλου μηχανικής μάθησης αναφορικά με το πόσο καλά μπορεί να προβλέψει ένα αποτέλεσμα y δεδομένης μιας εισόδου x. Όπου το x είναι ένα Set από δεδομένα.

$$x=\{x^{(1)}, \ldots, x^{(m)}\}$$

Όπως είναι για παράδειγμα ένα Set από εγγραφές δεδομένων, με κάθε στοιχείο του Set $x^{(i)}$ να αποτελείται από n αριθμό Features. Ενώ το y είναι αντίστοιχα ένα Set από αποτελέσματα, με το $y^{(i)}$ να αντιπροσωπεύει μια κατηγορία [32].

$$y=\{y^{(1)}, \ldots, y^{(m)}\}$$

Λόγω του ότι τα μοντέλα που εξετάζουμε στο πλαίσιο αυτής της διπλωματικής είναι τύπου Binary Classification, θα επικεντρωθούμε σε Metrics τα οποία μπορούν να ερμηνεύσουν την απόδοση ενός τέτοιου μοντέλου.

3.6.1 Confusion Matrix

Με τον όρο Confusion Matrix περιγράφουμε τη γραφική απεικόνιση των αποτελεσμάτων, ενός μοντέλου μηχανικής μάθησης, τα οποία κατανέμονται σε έναν πίνακα αναλόγως με το ποια είναι η πραγματική κατηγορία και ποια είναι αυτή που έχει προβλέψει το μοντέλο.

Όσον αφορά προβλήματα τύπου Binary Classification, όπως είναι το μοντέλο που προσπαθούμε να δημιουργήσουμε στο πλαίσιο της διπλωματικής, τα αποτελέσματα μιας πρόβλεψης κατανέμονται με βάση το παρακάτω πίνακα τύπου Confusion Matrix.

		Predicted Class		
		Positive MCI-AD	Negative NC	
Actual Class	Positive MCI-AD	True Positive	False Negative	Sensitivity TP/(TP+FN)
	Negative NC	False Positive	True Negative	Specificity TN/(FP+TN)
		Precision TP/(TP+FP)	NPV TN/(TN+FN)	Accuracy (TP+TN)/(TP+TN+FP+FN)

Πίνακας 2 Confusion Matrix βάση των κατηγοριών (MCI-AD, NC) του προβλήματος της διπλωματικής, όπως αυτές ορίζονται στην ενότητα επιλογής κατάλληλου Target Class §4.2.3.1

Ο συγκεκριμένος πίνακας ισχύει για όλα τα παραδείγματα που θα εξετάσουμε στο κεφάλαιο της αξιολόγησης. Πιο συγκεκριμένα οι καταστάσεις ενός δείγματος, με άλλα λόγια ενός Game Session προς κατηγοριοποίηση, με βάση το συγκεκριμένο Confusion Matrix, έχουν ως εξής.

- True Positive. Το Game Session αφορά χρήστη με MCI-AD και το μοντέλο προβλέπει επιτυχώς ότι ο χρήστης έχει MCI-AD.
- True Negative. Το Game Session αφορά άτομο με NC και το μοντέλο προβλέπει επιτυχώς ότι έχει NC.

- False Positive. Το Game Session στη πραγματικότητα αφορά χρήστη με NC αλλά το μοντέλο προβλέπει λανθασμένα ότι αφορά χρήστη με MCI-AD.
- False Negative. Το Game Session στη πραγματικότητα αφορά χρήστη με MCI-AD αλλά το μοντέλο προβλέπει λανθασμένα ότι αφορά χρήστη με NC.

3.6.2 Accuracy

Με το Metric Accuracy καταγράφουμε την ευστοχία στη πρόβλεψη ενός μοντέλου, μεγαλύτερο Accuracy συνεπάγεται καλύτερη απόδοση. Όσον αφορά τα Binary Classification μοντέλα, το Accuracy μπορεί να υπολογιστεί με τη παρακάτω συνάρτηση και αυτό που ουσιαστικά μας παρέχει ως πληροφορία είναι το κατά πόσο εύστοχα αναγνωρίζει το μοντέλο τις δύο κατηγορίες στο σύνολο των παρατηρήσεων.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

3.6.3 Precision

Το Metric Precision ή Positive Predictive Power όπως εναλλακτικά αναφέρεται, όσον αφορά μοντέλα Binary Classification, καταγράφει το ποσοστό των εύστοχων προβλέψεων μιας κατηγορίας ως προς το σύνολο των παρατηρήσεων. Μεγαλύτερο Precision συνεπάγεται καλύτερη απόδοση του μοντέλου. Η συνάρτηση με βάση την οποία υπολογίζεται είναι η ακόλουθη.

$$Precision = TP / (TP + FP)$$

3.6.4 Recall

Το Metric Recall ή Sensitivity όπως εναλλακτικά αναφέρεται, για προβλέψεις που αφορούν Binary Classification μοντέλα, μας δίνει το ποσοστό των εύστοχων παρατηρήσεων μιας κατηγορίας ως προς το πραγματικό σύνολο που θα έπρεπε να είχε προβλέψει το μοντέλο για αυτή τη συγκεκριμένη κατηγορία. Και σε αυτό το Metric μεγαλύτερο ποσοστό συνεπάγεται καλύτερη απόδοση του μοντέλου. Η συνάρτηση με βάση την οποία υπολογίζεται είναι η εξής.

$$Recall\ (Sensitivity) = TP / (TP + FN)$$

3.6.5 Specificity

Το Metric Specificity, όσον αφορά τις προβλέψεις ενός Binary Classifier μοντέλου, είναι ο λόγος (Ratio) του πραγματικού συνόλου της εκάστοτε κατηγορίας που θεωρούμε ως Negative (True Negative), προς το σύνολο των παρατηρήσεων που το μοντέλο χαρακτηρίζει ως Negative, δηλαδή ως προς τις True Negative συν τις False Positive παρατηρήσεις. Και στη περίπτωση του Specificity μια υψηλότερη τιμή συνεπάγεται καλύτερη απόδοση του μοντέλου. Η συνάρτηση με την οποία υπολογίζεται είναι η ακόλουθη.

$$Specificity = TN / (TN + FP)$$

3.6.6 Negative Predictive Value

Το Metric Negative predictive value (NPV), αντιπροσωπεύει την πιθανότητα μια τιμή να είναι στη πραγματικότητα Negative, δεδομένου του ότι το μοντέλο προέβλεψε τη συγκεκριμένη τιμή ως Negative. Η συνάρτηση υπολογισμού του NPV είναι η εξής.

Negative predictive value (NPV) = TN / (TN + FN)

Για παράδειγμα στη περίπτωση της διπλωματικής, όπου Positive θεωρούμε το ότι κάποιο άτομο έχει MCI-AD, το NPV μας δίνει την πιθανότητα, ένα άτομο που είναι NC να έχει κατηγοριοποιηθεί σωστά ως Negative.

3.6.7 F-Score

Το Metric F-Score ή F1 όπως αλλιώς συναντάται, είναι ένα ακόμη στατιστικό μέγεθος για τη μέτρηση της ακρίβειας ενός μοντέλου. Όσο αναφορά τα μοντέλα τύπου Binary Classification, το F-Score εκφράζεται ως ο μέσος όρος και συγκεκριμένα ο μέσος όρος τύπου Harmonic Mean, του λόγου (Ratio) του Precision προς το Recall. Το εύρος τιμών είναι πάντοτε μεταξύ 0 και 1, ενώ μια τιμή κοντά στο 1 συνεπάγεται καλύτερη απόδοση του μοντέλου. Το χαρακτηριστικό του F-Score είναι η ικανότητα του να αξιολογεί σωστά ένα Binary Classification μοντέλο ακόμη και όταν υπάρχει ανισοκατανομή μεταξύ των δύο κατηγοριών. Η συνάρτηση από την οποία προκύπτει είναι η ακόλουθη.

F-Score = 2 * (Precision * Recall) / (Precision + Recall)

3.6.8 Area Under Curve

Με τον όρο Area Under Curve (AUC) περιγράφουμε την επιφάνεια που βρίσκεται κάτω από τη γραφική αναπαράσταση της καμπύλης Receiver Operating Characteristics (ROC), όταν στον άξονα x έχουμε το False Positive Rate (FPR) και στον άξονα y έχουμε το True Positive Rate (TPR). Το FPR και το TPR περιγράφονται από τις παρακάτω συναρτήσεις.

FPR = Recall = TP / (TP + FN)

TPR = (1 - Specificity)

Αυτό που ουσιαστικά απεικονίζει η καμπύλη ROC και αντίστοιχα η επιφάνεια AUC, είναι η συσχέτιση του Sensitivity με το Specificity, κάτι το οποίο μεταφράζεται ως η ικανότητα ενός μοντέλου να αναγνωρίσει τη σωστή κατηγορία σε μια πρόβλεψη.

3.6.9 Precision Recall Curve

Με τον όρο Precision Recall Curve (PRC) περιγράφουμε την γραφική αναπαράσταση της καμπύλης που σχηματίζεται όταν στον άξονα x έχουμε το Recall (Sensitivity) και στον άξονα y το Precision.

Οπότε ουσιαστικά αυτή η καμπύλη μας δείχνει τη συσχέτιση μεταξύ αυτών των δύο. Πιο συγκεκριμένα η καμπύλη σχηματίζεται από μια σειρά διαδοχικών Ratios, μεταξύ του Precision και του Recall, στα οποία λαμβάνουμε υπόψη μας και ένα Threshold το οποίο αντιπροσωπεύει τα Probabilities του μοντέλου.

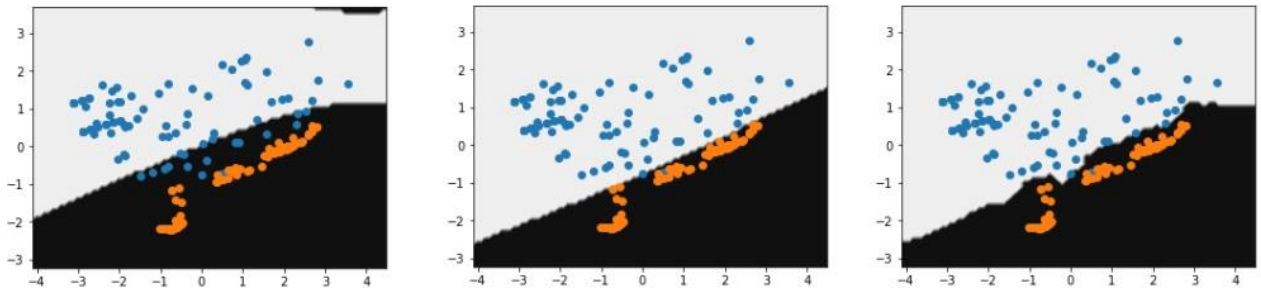
Η καμπύλη PRC προτιμάται σε σύγκριση με την καμπύλη ROC λόγω του ότι δεν επηρεάζεται όταν υπάρχει ανισοκατανομή μεταξύ των κατηγοριών του Target Class [33].

3.7 Overfitting, Underfitting & Solutions

Με τον όρο Overfitting αναφερόμαστε στη περίπτωση όπου ένα μοντέλο κατά τη διάρκεια της εκπαίδευσης έχει προσαρμοστεί απόλυτα στο Training Dataset, παρουσιάζοντας 100% Accuracy ή εναλλακτικά, όπως αναφέρεται στη βιβλιογραφία, παρουσιάζει υψηλό Variance [37].

Αντίστοιχα, ένα μοντέλο μπορεί να παρουσιάζει πολύ κακές επιδόσεις, σε εκείνη την περίπτωση έχουμε να κάνουμε με Underfitting του μοντέλου στο Training Dataset, κάτι που μεταφράζεται ως υψηλό Bias.

Αντιθέτως, ένα μοντέλο για να θεωρηθεί ιδανικό θα πρέπει να παρουσιάζει, μικρό Bias, μικρό Variance και οι επιδόσεις του στα Metrics, για παράδειγμα στο Accuracy, να είναι ελαφρώς καλύτερες στο Training Dataset, από ότι αυτές στο Testing Dataset.



Εικόνα 7 Παράδειγμα τριών διαφορετικών μοντέλων όπου για το ίδιο Dataset παρουσιάζουν, από τα αριστερά, Underfitting, Φυσιολογικό Bias/Variance Ratio και Overfitting αντίστοιχα.

Για τα μοντέλα που παρουσιάζουν υψηλό Variance, οι πιθανές λύσεις για την αποφυγή του Overfitting είναι οι εξής.

- Χρησιμοποίηση μεγαλύτερου Dataset, όπου αυτό είναι εφικτό.
- Εφαρμογή μεθόδων Regularization, για γραμμικά μοντέλα.
- Εφαρμογή μεθόδων Pruning, για μοντέλα τύπου Decision Trees και των παραλλαγών τους.
- Εκπαίδευση για λιγότερα Epochs (Iterations), για μοντέλα τύπου νευρωνικών δικτύων.
- Εφαρμογή Data Augmentation για εξισορρόπηση των κατηγοριών του Target Class.
- Χρήση πιο απλοϊκών μοντέλων.
- Ορισμός των προεπιλεγμένων τιμών για τις παραμέτρους του αλγορίθμου εκπαίδευσης.
- Εκτύπωση διαγράμματος Learning Curve για τη διαπίστωση του κατάλληλου λόγου μεταξύ του Training και του Testing Dataset, ως προς μια από τις παραμέτρους του αλγορίθμου εκπαίδευσης.

Για τα μοντέλα που παρουσιάζουν υψηλό Bias, οι πιθανές λύσεις για τη βελτίωση της απόδοσης και την αποφυγή του Underfitting είναι οι εξής.

- Χρήση ενός πιο σύνθετου αλγορίθμου εκπαίδευσης.
- Προσθήκη επιπλέον Feature στη διαδικασία της εκπαίδευσης.
- Εκπαίδευση για περισσότερα Epochs (Iterations) για μοντέλα τύπου Neural Networks.
- Εφαρμογή μεθόδων Optimization.

4

Μεθοδολογία

4.1 Διαθέσιμες μεθοδολογίες

Εάν ανατρέξει κανείς στη βιβλιογραφία με σκοπό να βρει ένα πρότυπο μεθοδολογίας για τη δημιουργία μοντέλων μηχανικής μάθησης, θα συναντήσει πληθώρα επιλογών. Για τις ανάγκες της διπλωματικής, έγινε μελέτη των εναλλακτικών μεθοδολογιών και στη συνέχεια λαμβάνοντας υπόψη τις απαιτήσεις της υλοποίησης, δημιουργήθηκε μια προσαρμοσμένη μεθοδολογία.

Η μεθοδολογία αυτή μοιράζεται κοινά χαρακτηριστικά με τις σημαντικότερες από τις ήδη υπάρχουσες ενώ εμβαθύνει σε σημεία που είναι σημαντικά για το συγκεκριμένο ερευνητικό ερώτημα. Στη συνέχεια ακολουθεί μια συνοπτική παρουσίαση των μεθοδολογιών που υπάρχουν διαθέσιμες, ενώ αμέσως μετά αναλύεται η προσαρμοσμένη μεθοδολογία που ακολουθήθηκε.

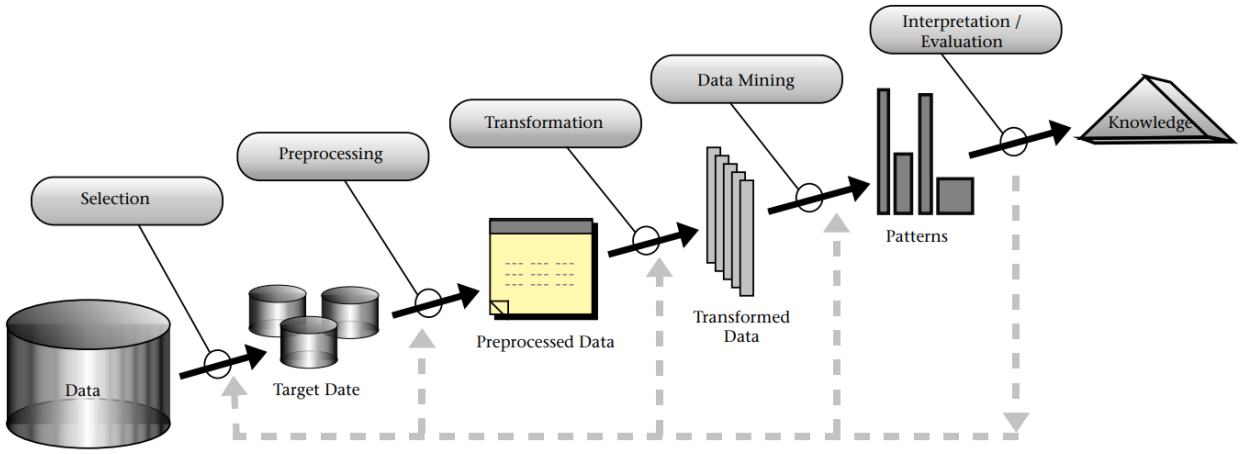
4.1.1 KDD

Η μεθοδολογία Knowledge Discovery in Databases (KDD), δημιουργήθηκε με σκοπό να αυτοματοποιήσει το μεγαλύτερο μέρος της διαδικασίας που ονομάζεται Online Analytical Processing (OLAP).

Η διαδικασία OLAP αποτελεί μια πολυδιάστατη προσέγγιση σε ότι αφορά τις διαδικασίες της ανάλυσης των δεδομένων μιας βάσης δεδομένων για εξαγωγή συμπερασμάτων, όπως για παράδειγμα συνόλων, υποσυνόλων και άλλων στατιστικών συμπερασμάτων.

Εμπνευστής της OLAP είναι ο Edgar Frank Codd, γνωστός επίσης και για τη δημιουργία του μοντέλου των σχεσιακών βάσεων δεδομένων (RDBMS) [3].

Η KDD μεθοδολογία αποτελείται από την είσοδο (Selection) των δεδομένων μιας βάσης δεδομένων και το μετασχηματισμό αυτών σε γνώση (Knowledge) μέσω των βημάτων (Preprocessing, Transformation, Data Mining) και τέλος στην εξαγωγή των συμπερασμάτων μέσω της ερμηνείας και της αξιολόγησης των αποτελεσμάτων (Interpretation, Evaluation) όπως φαίνεται και στο διάγραμμα της παρακάτω εικόνας.

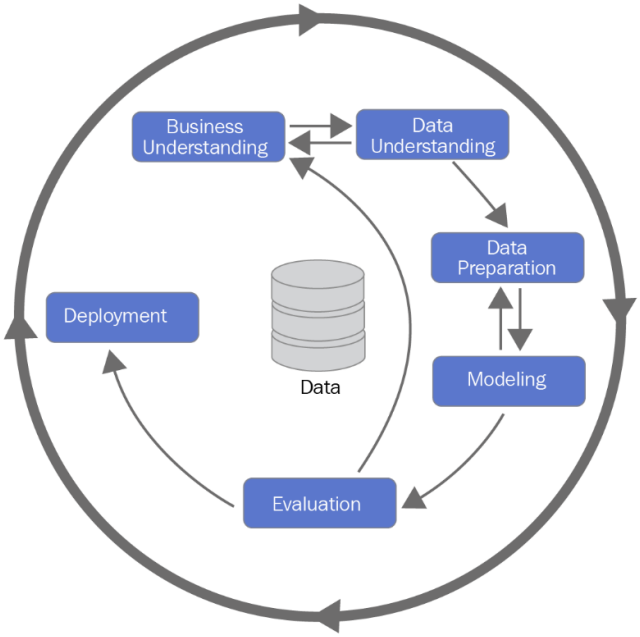


Εικόνα 8 Διάγραμμα μιας τυπικής διαδικασίας σχεδιασμένης με βάση το μοντέλο της KDD μεθοδολογίας [3]

Η διαδικασία του KDD, αποτελεί μια μη-τετριμμένη διαδικασία για την αναγνώριση σχεσιακών μοτίβων στα δεδομένα ενός Dataset. Πιο συγκεκριμένα στόχος της διαδικασίας του KDD είναι να αναγνωριστούν χρήσιμα, έγκυρα και ενδιαφέροντα μοτίβα και συσχετίσεις εντός του Dataset [6].

4.1.2 CRISP-DM

Η μεθοδολογία Cross-industry standard process for data mining (CRISP-DM), αποτελεί τη πλέον γνωστή μεθοδολογία, η οποία περιγράφει τα βήματα για να φτάσει κανείς από το σημείο να έχει ακατέργαστα δεδομένα στο να έχει δημιουργήσει ένα μοντέλο μηχανικής μάθησης έτοιμο είτε για να χρησιμοποιηθεί σε κάποιο παραγωγικό σύστημα. Η μεθοδολογία CRISP-DM δεν εμβαθύνει στα επιμέρους βήματα, αντιθέτως αφήνει στην ευχέρεια αυτού που θα υιοθετήσει τη μεθοδολογία να προσαρμόσει την υλοποίηση. Αυτό που προσφέρει είναι γενικές κατευθύνσεις για βήματα τα οποία είναι ουσιώδη και δεν πρέπει να παρακάμπτονται σε ένα project δημιουργίας μοντέλων μηχανικής μάθησης.



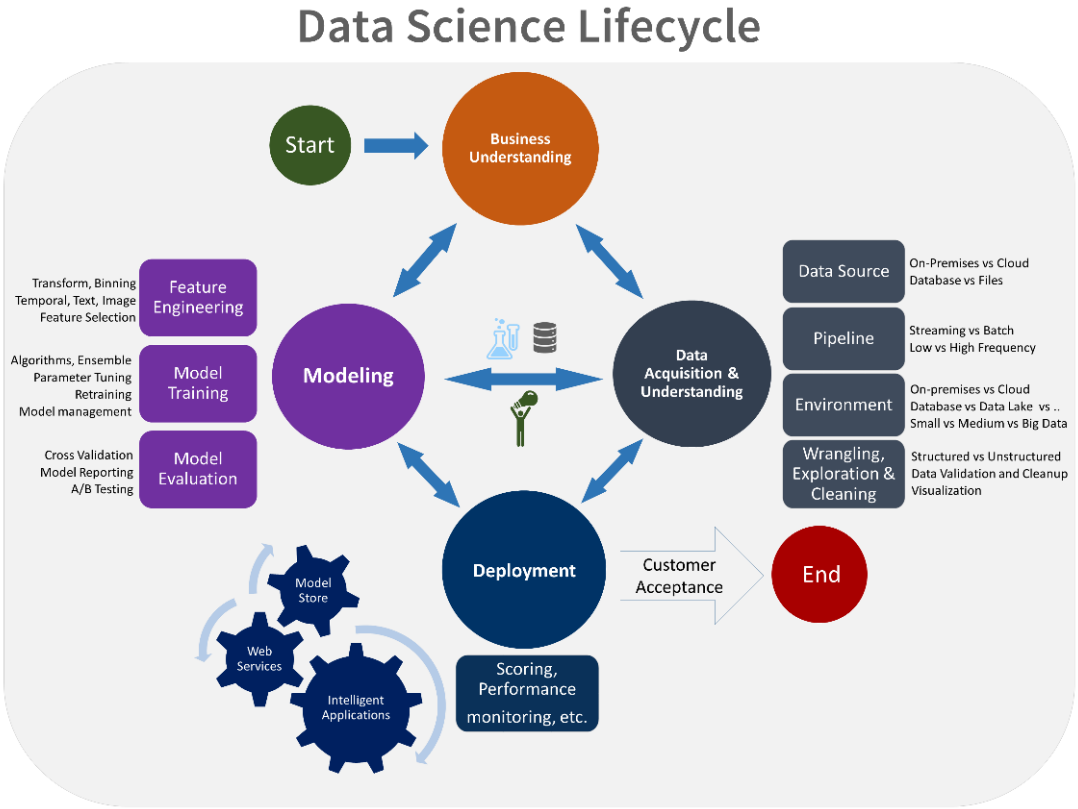
Εικόνα 9 Διάγραμμα μιας τυπικής διαδικασίας σχεδιασμένης με βάση το μοντέλο της CRISP-DM μεθοδολογίας [44]

Όσον αφορά τη διπλωματική, θα μπορούσαμε να πούμε ότι η μεθοδολογία που ακολουθήθηκε υιοθετεί το σύνολο των βημάτων της μεθοδολογίας CRISP-DM με εξαίρεση το τελευταίο στάδιο της παραγωγικής ανάπτυξης του συστήματος (Deployment).

Εάν θέλαμε να συγκρίνουμε το μοντέλο KDD με το μοντέλο CRISP-DM, θα μπορούσαμε να πούμε ότι το CRISP-DM αποτελεί ένα υπερσύνολο του KDD και πιο συγκεκριμένα ότι το KDD αντιπροσωπεύει τη διαδικασία που γίνεται μεταξύ των βημάτων Data Preparation, Modeling και Evaluation του μοντέλου CRISP-DM.

4.1.3 TDSP

Η μεθοδολογία Team Data Science Process (TDSP), δημιουργήθηκε το 2016 από τη Microsoft και βασίζεται στη φιλοσοφία της Agile αντιμετώπισης ενός Project, με άλλα λόγια δίνει έμφαση πρώτον στην επαναληψιμότητα των βημάτων της διαδικασίας και δεύτερον στη διευκόλυνση της συνεργασίας μεταξύ των μελών της ομάδας [4]. Αυτό που κάνει το TDSP να ξεχωρίζει ως προς τη μεθοδολογία CRISP-DM είναι η πληρότητα του μοντέλου καθώς πέρα από τα βασικά βήματα αναφέρονται και οι επιμέρους εργασίες σε κάθε ένα από αυτά.



Εικόνα 10 Διάγραμμα μιας τυπικής διαδικασίας σχεδιασμένης με βάση το μοντέλο της TDSP μεθοδολογίας [45]

Πέρα όμως από το αναλυτικό διάγραμμα που παρουσιάζει, γίνεται αναφορά στους διακριτούς ρόλους που υπάρχουν μέσα σε ένα Data Science Project, όπως είναι για παράδειγμα ο ρόλος του Project Lead, του Project Manager, του Application Developer, του Data Scientist, του Data Engineer και του Solution Architect. Επιπλέον, παρέχει ένα βασικό οργανόγραμμα με τις επιμέρους ενέργειες που αντιστοιχούν σε κάθε ρόλο, για κάθε στάδιο της διαδικασίας.

Τέλος, το TDSP παρέχει μια έτοιμη κενή δομή ενός Project, με άλλα λόγια ένα Project Scaffold, ώστε να προτυποποιήσει τη διαδικασία της αρχικοποίησης ενός Project, με σκοπό να γίνει πιο εύκολη η διαχείριση του από τα μέλη της ομάδας.

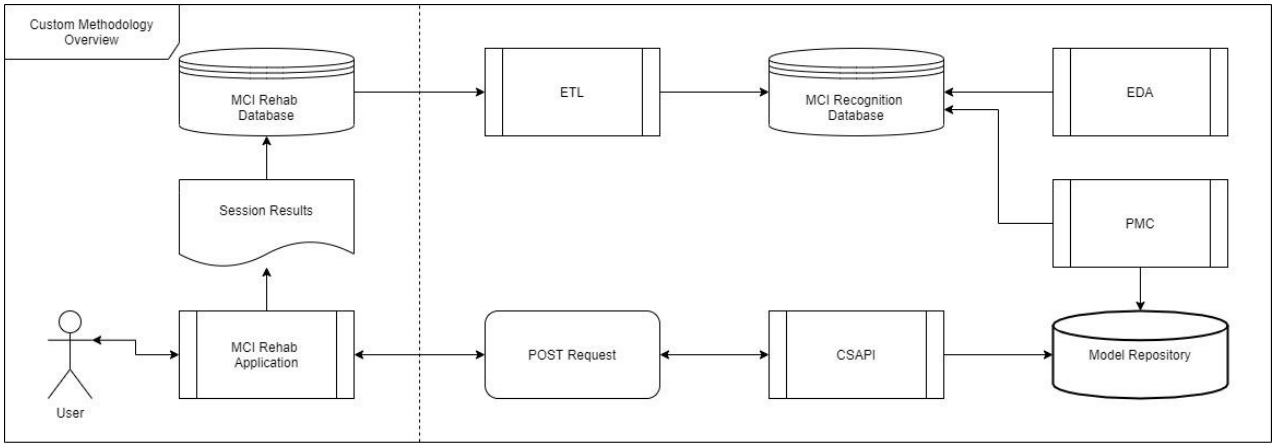
Στα πλαίσια της διπλωματικής δε θα είχε νόημα να υιοθετηθεί μια μεθοδολογία όπως το TDSP στο σύνολο της, καθώς πολλά από τα χαρακτηριστικά και τις διαδικασίες αφορούν τη συνεργασία μιας ομάδας, ωστόσο έχει υιοθετηθεί εν μέρη η δομή του Project και συγκεκριμένα το κομμάτι που αφορά το σαφή διαχωρισμό δεδομένων και κώδικα.

4.2 Προσαρμοσμένη μεθοδολογία

Όπως η συντριπτική πλειοψηφία των Data Scientist, 82% σύμφωνα με έρευνα που δημοσιεύτηκε το 2018 [4], έτσι και στη συγκεκριμένη διπλωματική δε θα υιοθετηθεί μια συγκεκριμένη μεθοδολογία αλλά θα δημιουργηθεί μια προσαρμοσμένη μεθοδολογία η οποίο θα εξυπηρετεί τις ανάγκες της υλοποίησης.

Εξετάζοντας μακροσκοπικά το σύνολο της υλοποίησης θα μπορούσε κανείς να διακρίνει τις διαδικασίες σε τέσσερα διακριτά μέρη, αυτές είναι οι εξής: Η διαδικασία διαχείρισης των δεδομένων των αρχικών δεδομένων Extract Transform Load (ETL), η διερευνητική ανάλυση Exploratory Data Analysis (EDA), η δημιουργία του παραγωγικού μοντέλου Production Model Creation (PMC) και τέλος η υπηρεσία για τη κατηγοριοποίηση των νέων δεδομένων Classification Service Application Programming Interface (CSAPI).

Στη συνέχεια θα δούμε τις διαδικασίες αναλυτικά, καταγράφοντας στοιχεία όπως για παράδειγμα, ποιο σκοπό εξυπηρετούν, ποια είναι η είσοδος, τα βήματα, το αναμενόμενο αποτέλεσμα, καθώς επίσης και ποιο ήταν το πραγματικό αποτέλεσμα που προέκυψε από τις δοκιμές για τη κάθε μια από αυτές τις διαδικασίες. Επίσης, στο παρακάτω διάγραμμα μπορούμε να δούμε μια γενική συγκεντρωτική επισκόπηση των επιμέρους διαδικασιών.



Εικόνα 11 Επισκόπηση της προσαρμοσμένης μεθοδολογίας που ακολουθήθηκε

4.2.1 Initial Dataset Fields

Σε αυτή την ενότητα, για να αποκτήσουμε μια εικόνα του Dataset, θα δούμε τα πεδία όπου υπήρχαν στην βάση δεδομένων της εφαρμογής MCI Rehab όσον αφορά τους πίνακες στους οποίους καταγράφονταν τα αντίστοιχα Game Sessions και Game Rounds.

Βάση της χαρτογράφησης των πηγών από όπου μπορούν να προέλθουν δεδομένα που σχετίζονται με την υγεία ενός ατόμου, όπως αυτή περιγράφεται στη δημοσίευση "Finding the Missing Link for Big Biomedical Data" [36], τα πεδία των δεδομένων που έχουμε στη διάθεση μας, θα μπορούσαν να ομαδοποιηθούν ως εξής.

- Δημογραφικά στοιχεία του χρήστη (HL7).
    - Ηλικία, φύλο, ανώτατο εκπαιδευτικό επίπεδο και οικογενειακή κατάσταση.
- Ιατρικό προφίλ του χρήστη (Diagnoses).
    - Οικογενειακό ιατρικό ιστορικό, κατάθλιψη, υπέρταση.
- Τρόπος ζωής του χρήστη (Life style).
    - Κάπνισμα, άσκηση, εξοικείωση με την τεχνολογία (χρήση Η/Υ, Smartphone).
- Καταγεγραμμένα στοιχεία που αφορούν τα παιχνίδια της εφαρμογής MCI Rehab. Τα στοιχεία αυτά θα μπορούσαν να θεωρηθούν ως δεδομένα εκτός του συστήματος υγείας, τα οποία θα εξετάσουμε για το κατά πόσο προσφέρουν ενδείξεις σχετικά με τη γνωστική κατάσταση ενός ατόμου.
    - Σε επίπεδο Session:
        - Το αναγνωριστικό του Session (session_id)
        - Το αναγνωριστικό του χρήστη (register_id)
    - Σε επίπεδο Game Round:
        - Το είδος του παιχνιδιού (game_id)
        - Το αποτέλεσμα του παιχνιδιού (game_success)
        - Ο λόγος τερματισμού ενός παιχνιδιού (game_event)
        - Η ώρα έναρξης και λήξης του παιχνιδιού (start_tm και end_tm αντίστοιχα)
        - Η διάρκεια του παιχνιδιού (tm_sec)
        - Οι πόντοι που συλλέχθηκαν σε ένα παιχνίδι (points)
        - Το επίπεδο δυσκολίας του παιχνιδιού (game_level)
        - Η επανάληψη του παιχνιδιού (game_repeat)
        - Η επανάληψη του επιπέδου δυσκολίας (level_repeat)

Όσον αφορά τα πεδία σε επίπεδο Game Round, λόγω του ότι για την εκπαίδευση του μοντέλου ανακτούμε τη πληροφορία σε επίπεδο Game Session, οι πληροφορίες που αφορούν τα Game Rounds ενός Session γίνονται Aggregate στη γραμμή του Session. Αυτός είναι και ο τρόπος με τον οποίο προχωρούμε στη δημιουργία των πρώτων Engineered Features όπως θα δούμε στην αντίστοιχη ενότητα της διαδικασίας EDA.

4.2.2 Extract, Transform, Load

Στο πλαίσιο της διπλωματικής, η βασική διαδικασία που έγινε στο στάδιο του ETL, ήταν η μετάπτωση της βάσης δεδομένων, οι επιμέρους ενέργειες αυτής της διαδικασία περιγράφονται παρακάτω.

Πρώτον, έγινε μελέτη των δεδομένων στην αρχική τους μορφή ώστε να διαπιστωθεί τι αναπαριστούν και ποιες είναι συσχετίσεις μεταξύ τους. Τα δεδομένα στην αρχική τους μορφή βρίσκονταν αποθηκευμένα σε μια βάση δεδομένων τύπου PostgreSQL και πιο συγκεκριμένα σε 3 πανομοιότυπα Schemas τα οποία προέρχονται ουσιαστικά από τα 3 διαφορετικά Tablets που είχαν στη διάθεση τους οι χρήστες της εφαρμογής MCI Rehab.

Δεύτερο βήμα ήταν η σχεδίαση ενός νέου Schema σε βάση δεδομένων τύπου MariaDb, λαμβάνοντας υπόψη τις βασικές αρχές της κανονικοποίησης (Database Normalization).

```
CREATE TABLE `game_rounds` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `session_id` bigint(20) unsigned NOT NULL,
  `game_id` bigint(20) unsigned NOT NULL,
  `round_success` tinyint(4) NOT NULL,
  `round_event` int(10) unsigned NOT NULL,
  `round_level` int(10) unsigned NOT NULL,
  `round_start_dt` datetime NOT NULL,
  `round_end_dt` datetime DEFAULT NULL,
  `round_time` int(10) unsigned NOT NULL,
  `round_points` int(10) unsigned NOT NULL,
  `level_repeat` int(10) unsigned NOT NULL,
  `game_retries` int(10) unsigned NOT NULL,
  PRIMARY KEY (`id`),
  KEY `game_rounds_fk` (`session_id`),
  KEY `game_rounds_fk_1` (`game_id`),
  CONSTRAINT `game_rounds_fk` FOREIGN KEY (`session_id`) REFERENCES `game_sessions` (`id`) ON UPDATE CASCADE,
  CONSTRAINT `game_rounds_fk_1` FOREIGN KEY (`game_id`) REFERENCES `games` (`id`) ON UPDATE CASCADE
);
```

Εικόνα 12 SQL DDL Command για τη δημιουργία του πίνακα game_rounds

Το νέο Schema με τον τρόπο που σχεδιάστηκε πληροί τις προϋποθέσεις για να θεωρείται Normalized σε επίπεδο 3NF (Third Normal Form). Πιο αναλυτικά για το επίπεδο Normalization της βάσης μπορούμε να πούμε τα εξής.

Όλοι οι πίνακες διαθέτουν πρωτεύον κλειδί (Primary Key) οπότε το νέο Schema είναι τουλάχιστον κανονικοποιημένο σε επίπεδο 1NF (First Normal Form).

Σε κανέναν πίνακα όπου υπάρχει κάποιο Foreign Key δεν υπάρχει παράλληλα πεδίο το οποίο να είναι σχετικό με την οντότητα που αντιπροσωπεύει το αντίστοιχο Foreign Key. Παράδειγμα από την υλοποίηση είναι το ότι ο πίνακας των Rounds έχει μόνο το Foreign Key των Sessions χωρίς να υπάρχει άλλο πεδίο που να αφορά τα Sessions. Αντίστοιχα το ίδιο ισχύει και για τον πίνακα των Sessions όπου υπάρχει το Foreign Key των Users χωρίς να υπάρχει κανένα άλλο πεδίο σχετικό με τους Users. Συνεπώς, μπορούμε να πούμε ότι το Schema είναι κανονικοποιημένο σε επίπεδο 2NF (Second Normal Form).

Επίσης έχει γίνει αποφυγή έμμεσων συσχετίσεων, με άλλα λόγια έχει προβλεφθεί τα στοιχεία της κάθε οντότητας να είναι αποκλειστικά και μόνο στον εκάστοτε πίνακα που αντιπροσωπεύει την οντότητα και η συσχέτιση να γίνεται μόνο με τη χρήση Foreign Keys. Οπότε μπορούμε να πούμε ότι το Schema είναι κανονικοποιημένο σε επίπεδο 3NF (Third Normal Form).

Τέλος, για την ολοκλήρωση της διαδικασίας της μετάπτωσης, το τελευταίο Transformation είναι η συγγραφή των απαραίτητων SQL Queries για τη μεταφορά των δεδομένων. Στη προκειμένη περίπτωση, λόγω του ότι το νέο Schema βρίσκεται σε διαφορετική βάση δεδομένων, η διαδικασία για κάθε πίνακα δε γίνεται να ολοκληρωθεί με ένα μοναδικό Query, αλλά θα πρέπει κάθε φορά να εκτελείται ένα Query για την ανάκτηση των δεδομένων και ένα για την εισαγωγή.

Για την ανάκτηση χρησιμοποιούμε πάντα Union ώστε να πάρουμε αποτελέσματα με Select και από τα τρία αρχικά Schemas, εκτελώντας την εντολή στην PostgreSQL βάση δεδομένων, όπως φαίνεται ενδεικτικά στη παρακάτω εντολή για την επιλογή των Sessions.

```
select l.session_id, l.register_id, l.login_date, l.logout_date from tablet1.login l
union
select l2.session_id, l2.register_id, l2.login_date, l2.logout_date from tablet2.login l2
union
select l3.session_id, l3.register_id, l3.login_date, l3.logout_date from tablet3.login l3;
```

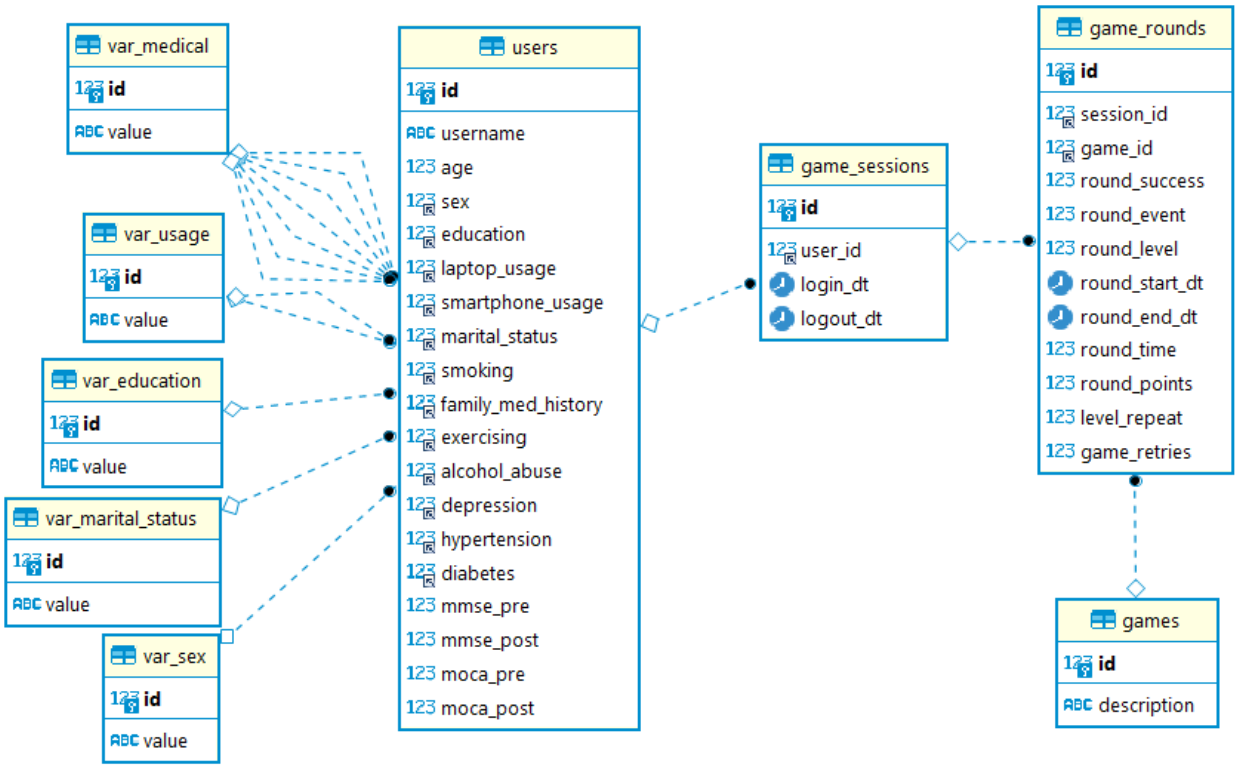
Εικόνα 13 Εντολή SQL για ανάκτηση των Sessions από πολλαπλά Database Schemas

Ενώ στη συνέχεια εκτελούμε την παρακάτω εντολή Insert στη νέα βάση δεδομένων MariaDb, για την εισαγωγή των δεδομένων στο νέο Schema.

```
insert into mci_db.game_sessions (id, user_id, login_dt, logout_dt) values
(1211389,33,'2019-07-20 17:19','2019-07-20 17:34'),
(4880526,8,'2019-05-08 14:54','2019-05-08 15:30'),
(1910652,7,'2019-05-10 17:28','2019-05-10 17:59'),
(1271340,23,'2019-06-12 10:33','2019-06-12 10:53');
```

Εικόνα 14 Εντολή SQL για εισαγωγή των Sessions στο νέο Database Schema

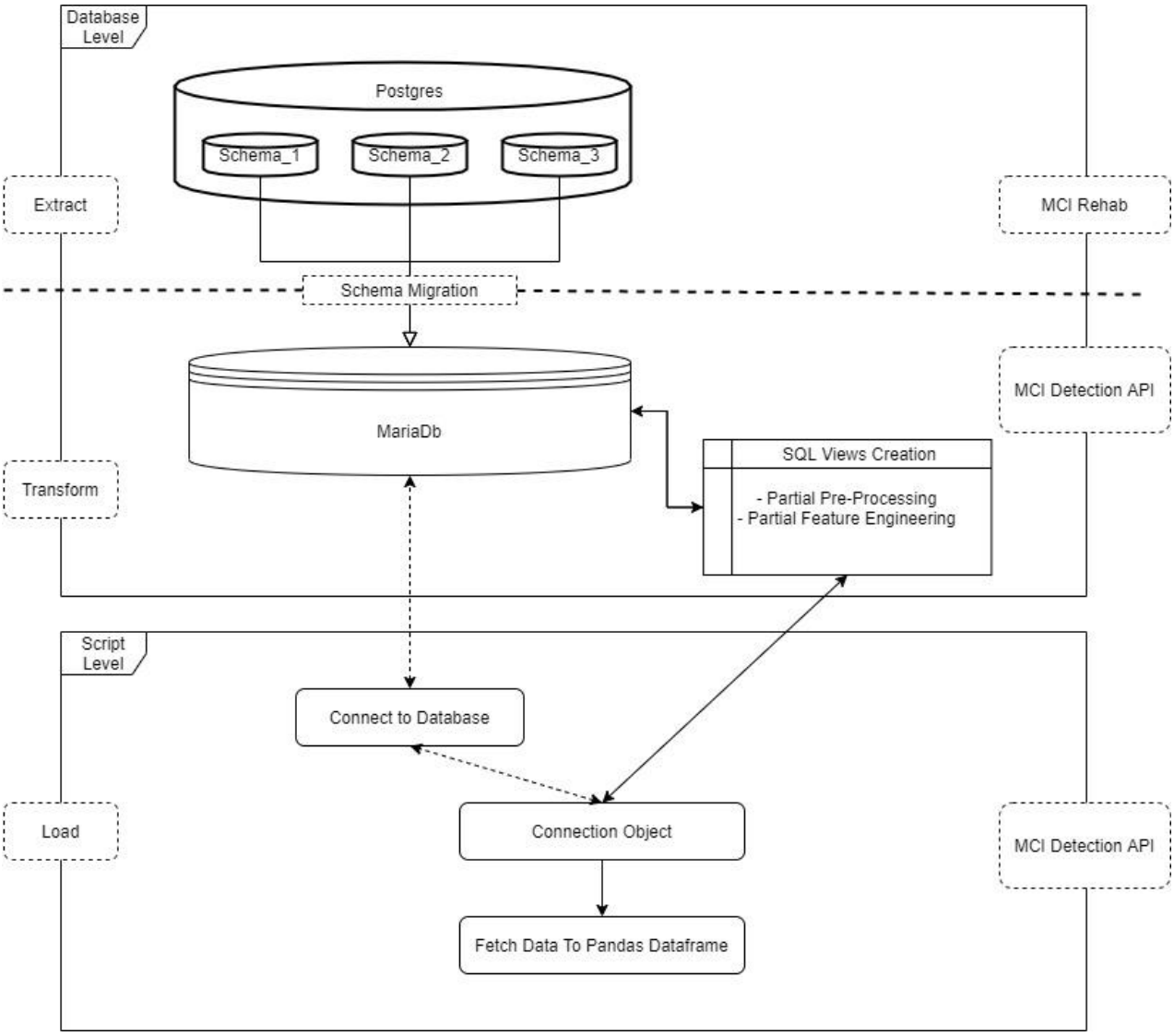
Το τελικό αποτέλεσμα της διαδικασίας του ETL όσον αφορά τους πίνακες και τη συσχέτιση μεταξύ τους φαίνεται στο παρακάτω ERD (Entity Relationship Diagram) διάγραμμα.



Εικόνα 15 Διάγραμμα τύπου Entity Relationship Diagram (ERD) για το Schema της ΒΔ

Η γενική εικόνα των δεδομένων με την ολοκλήρωση της μετάπτωσης είναι 9 Users με πλήρη στοιχεία, 119 Game Sessions με τουλάχιστον ένα Game Round και συνολικά 2951 Game Rounds.

Η διαδικασία του ETL, όπως περιγράφεται και από το παρακάτω διάγραμμα, ολοκληρώνεται σε επίπεδο Python Script, με την ανάκτηση των δεδομένων (Loading), σε ένα Pandas Dataframe.



Εικόνα 16 Διάγραμμα διαδικασίας ETL

4.2.3 Exploratory Data Analysis

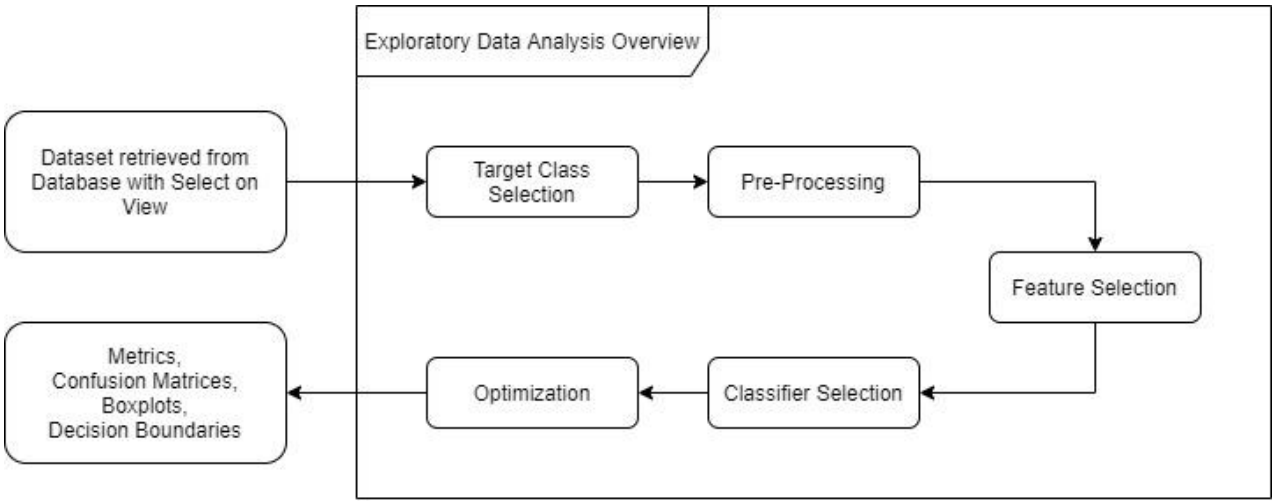
Η διαδικασία Exploratory Data Analysis (EDA) είναι ίσως το σημαντικότερο σημείο της μεθοδολογίας που θα ακολουθήσουμε για να απαντήσουμε στο βασικό ερευνητικό ερώτημα.

Αυτό που μπορούμε να θεωρήσουμε ως είσοδο σε αυτή τη διαδικασία είναι τα δεδομένα του Dataset στην μορφή που γίνονται Load κατά τη διαδικασία του ETL, στη προκειμένη περίπτωση όπως ανακτώνται με χρήση SQL View από τη ΒΔ.

Τα βήματα της διαδικασίας EDA τα οποία παρουσιάζονται αναλυτικά στις αμέσως επόμενες ενότητες είναι το Target Selection, το Preprocessing, το Feature Selection, το Classifier Selection και το Optimization. Η έξοδος σε κάθε βήμα είναι είτε κάποιο Plot είτε κάποιο Metric που θα μας βοηθήσει να πάρουμε μια απόφαση, αναλόγως με το τι διερευνούμε σε κάθε βήμα.

Το επιθυμητό αποτέλεσμα της διαδικασίας EDA είναι να καταλήξουμε σε μια λίστα από ιδανικά Features, ένα αλγόριθμο καθώς και τις ιδανικές τιμές για τις παραμέτρους του αλγορίθμου.

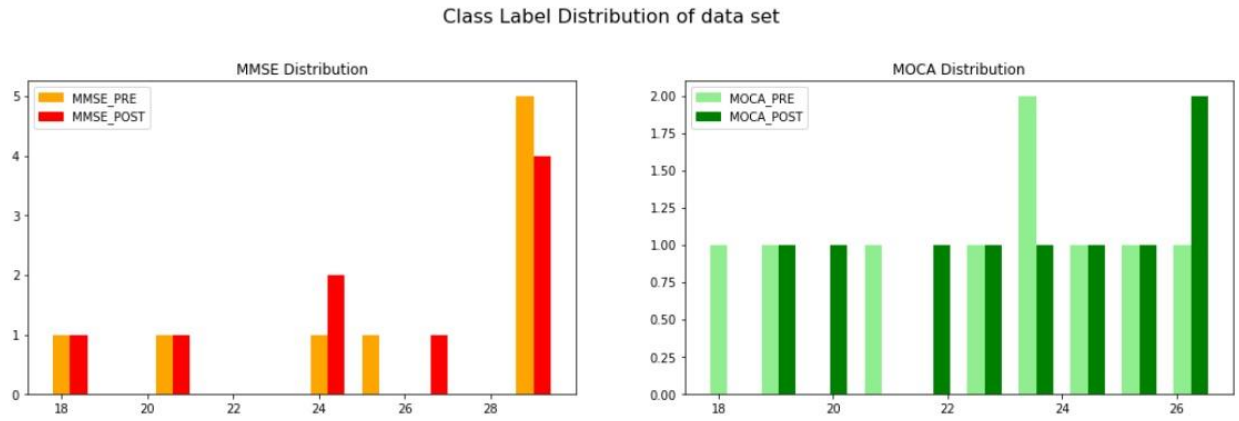
Τέλος, είναι σημαντικό να τονίσουμε το ότι η διαδικασία EDA, είναι ουσιαστικά μια σειρά πειραμάτων, μέσω των οποίων διερευνούμε τα ιδανικά Features, τους αλγορίθμους και τις παραμέτρους τους για το παραγωγικό μοντέλο. Συνεπώς είναι θεμιτό κατά τη διάρκεια των πειραμάτων και μέχρι να καταλήξουμε στο βέλτιστο αποτέλεσμα, να επιστρέφουμε σε προηγούμενα βήματα για να κάνουμε τροποποιήσεις.



Εικόνα 17 Γενική επισκόπηση της διαδικασίας Exploratory Data Analysis

4.2.3.1 Target Class Selection

Σε αυτό το στάδιο, ζητούμενο είναι η επιλογή της κατάλληλης εξαρτημένης μεταβλητής ως Target Class. Δεδομένου του ότι η ομάδα εστίασης συμμετείχε σε συνολικά 4 νευροψυχολογικές δοκιμασίες στο πλαίσιο της έρευνας που έγινε για το MCI Rehab, το αρχικό μας Dataset διαθέτει 4 εξαρτημένες μεταβλητές από τις οποίες μπορούμε να επιλέξουμε ένα Target Class.



Εικόνα 18 Κατανομή βαθμολογίας για τις νευροψυχολογικές δοκιμασίες MMSE και MOCA πριν και μετά τη χρήση της εφαρμογής MCI Rehab

Τα αποτελέσματα αυτών των δοκιμασιών, βρίσκονται ως πεδία στον πίνακα των Users και τα αντίστοιχα ονόματα των πεδίων τους χαρακτηρίζονται από το όνομα της δοκιμασίας και το πότε διεξήχθη συγκριτικά με τα Sessions παιχνιδιών των χρηστών. Έτσι έχουμε τα MMSE_PRE_INIT, MOCA_PRE_INIT, MMSE_POST_INIT, MOCA_POST_INIT. Το τελευταίο συνθετικό του ονόματος INIT, χαρακτηρίζει το ότι πρόκειται για τις αρχικές τιμές οι οποίες έχουν Range τιμών από 0 έως 30 και για

τις δύο δοκιμασίες και χρησιμεύει ώστε να διαχωρίζει τα πεδία με τις αρχικές τιμές από τα αντίστοιχα Discretized πεδία.

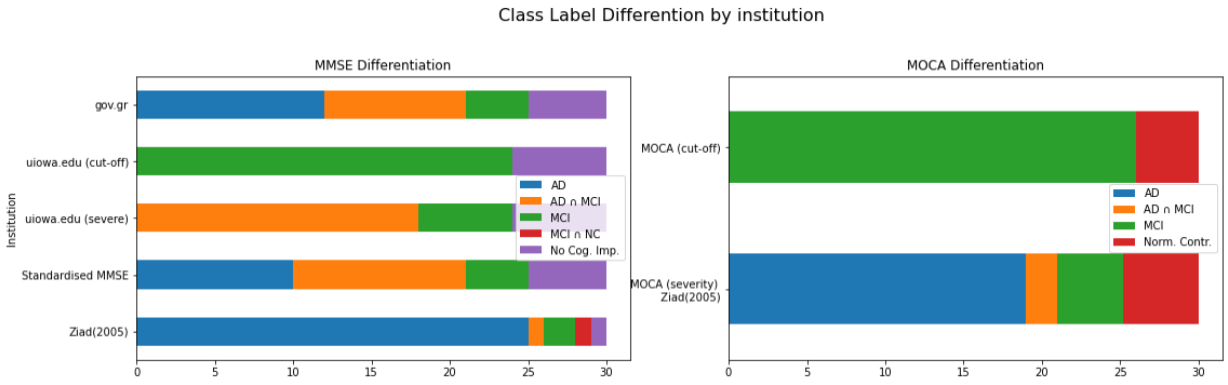
Για να μπορέσουμε να επιλέξουμε το ιδανικό Target Class, θα λάβουμε υπόψη μας τρεις παράγοντες.

Πρώτον, την κατανομή των αποτελεσμάτων για κάθε μια από τις 4 κλάσεις. Βάση του παρακάτω Plot, η κατανομή της κλάσης MOCA_PRE εμφανίζει με πιο ομοιόμορφη κατανομή συγκριτικά με τις υπόλοιπες κλάσεις.

Δεύτερον, το επόμενο κριτήριο για την επιλογή του Target Class είναι το πόσο καλά διαχωρίσιμα είναι τα Labels της κάθε κλάσης. Τα Labels των κλάσεων για το Context της διπλωματικής αντιπροσωπεύουν ουσιαστικά τα διάφορα γνωστικά επίπεδα. Συνεπώς μας ενδιαφέρει το πόσο εύκολα διαχωρίσιμα είναι μεταξύ τους αυτά τα επίπεδα.

Με βάση το παρακάτω Plot παρατηρούμε ότι τα αποτελέσματα της δοκιμασίας MMSE διαχωρίζονται σε διαφορετικά Set από Labels ανάλογα με την έρευνα ή το ίδρυμα που αναφέρεται στη συγκεκριμένη δοκιμασία. Επίσης αναφορικά με το MMSE, παρατηρούμε ότι το Label MCI, είτε αμιγώς είτε σε συνδυασμό με το Label AD, παρουσιάζει αρκετά διαφορετικό εύρος τιμών κατά περίπτωση, όπως φαίνεται και στο Plot της παρακάτω εικόνας [19], [20], [21].

Σε αντίθεση με τα Labels της δοκιμασίας MOCA, των οποίων το Range των τιμών έχει καθοριστεί από την έρευνα που έγινε στο πλαίσιο της δημοσίευσης όπου παρουσιάστηκε η δοκιμασία για πρώτη φορά. Βασικό επίτευγμα της συγκεκριμένης έρευνας ήταν η ανίχνευση ατόμων με MCI όπου με βάση τα αποτελέσματά τους στη MMSE δοκιμασία παρουσίαζαν φυσιολογικό γνωστικό επίπεδο [18]. Για τους παραπάνω λόγους, μεταξύ των δύο δοκιμασιών, επιλέγουμε τα αποτελέσματα της δοκιμασίας MOCA.



Εικόνα 19 Διαφοροποίηση των ορίων που καθορίζουν τις κατηγορίες στις νευροψυχολογικές δοκιμασίες MMSE και MOCA αντίστοιχα

Τρίτον, μεταξύ των αποτελεσμάτων MOCA_PRE και MOCA_POST, επιλέγουμε το πεδίο MOCA_PRE, καθώς έχει λίγο καλύτερη κατανομή αλλά και επειδή θα μας ενδιέφερε περισσότερο να δούμε σε τι γνωστικό επίπεδο ήταν τα υποκείμενα προτού εξασκηθούν μέσω της εφαρμογής MCI Rehab.

Οι πιθανές κατηγορίες του Target Class βασίζονται στα διαφορετικά γνωστικά επίπεδα και τους συνδυασμούς τους, αναλόγως με τις τιμές Cut-Off που θα επιλέξουμε για τον διαχωρισμό τους. Οι βασικές κατηγορίες είναι οι εξής.

- Alzheimer's Disease (AD). Άτομα με γνωστική επάρκεια αντίστοιχου επιπέδου με αυτή ατόμων που έχουν διαγνωστεί με Alzheimer's Disease.

- Mild Cognitive Impairment (MCI). Άτομα που παρουσιάζουν γνωστική επάρκεια αντίστοιχου επιπέδου με αυτή ατόμων που έχουν διαγνωστεί με κάποιας μορφής MCI.
- Normal Cognition / Normal Controls (NC). Άτομα που παρουσιάζουν φυσιολογική γνωστική επάρκεια.

Τέλος, όσον αφορά την επιλογή του Target Class, μεταξύ των δύο διαφορετικών διαχωρισμών των γνωστικών επιπέδων που εφαρμόζει η δοκιμασία MOCA {AD-MCI, NC} και {AD, MCI, NC}, έγινε επιλογή του Set που διαχωρίζει τα Labels σε δύο κατηγορίες, {AD-MCI, NC}. Αυτό παίζει καθοριστικό ρόλο στη διαμόρφωση της υλοποίησης καθώς επιλέγοντας ένα Target Class με δύο πιθανά Labels, το πρόβλημα που έχουμε να επιλύσουμε θεωρείται Binary Classification. Αυτό με τη σειρά του επηρεάζει τα Metrics που μπορούμε να χρησιμοποιήσουμε για την αξιολόγηση των μοντέλων.

4.2.3.2 Preprocessing

Έχοντας αποφασίσει για το πιο θα είναι το Target Class, μπορούμε να προχωρήσουμε στη διαδικασία της προετοιμασίας του Dataset ώστε τα δεδομένα μας να είναι σε θέση να επεξεργαστούν από τους αλγόριθμους οι οποίοι θα μας βοηθήσουν να καταλήξουμε στα κατάλληλα Features για το μοντέλο.

Πέρα όμως από τη προετοιμασία των δεδομένων από την άποψη των Transformations των τιμών των πεδίων ώστε να έρθουν στο κατάλληλο Format, υπάρχουν διαδικασίες που πρέπει να γίνουν στο στάδιο του Preprocessing. Όπως θα δούμε στην ενότητα για το Scaling, η οποία είναι μια από τις διαδικασίες που έχουν να κάνουν με την ικανότητα του μοντέλου να γενικεύει (Model Generalization), με άλλα λόγια να μπορεί να διατηρεί την ακρίβεια των προβλέψεων του και σε νέα Datasets.

4.2.3.2.1 Handle Missing Values

Για την υλοποίηση αυτής της διπλωματικής, η λύση που επιλέχθηκε για αυτό το πρόβλημα ήταν η αφαίρεση των εγγραφών. Θα περίμενε κανείς λόγω του σχετικά μικρού όγκου των εγγραφών να χρησιμοποιηθεί κάποια μέθοδος αντικατάστασης, ωστόσο οι εγγραφές που τυχαίνει να έχουν ελλιπή στοιχεία περιοριζόταν μόνο στους πίνακες των Game Rounds και στο σύνολο τους έχουν ελλιπή σχεδόν όλα τα πεδία τους. Αυτό συμβαίνει διότι πρόκειται για Rounds στα οποία παρουσιάστηκε κάποιο Exception κατά τη διάρκεια του παιχνιδιού.

Εκ πρώτης όψης θα μπορούσε να πει κανείς ότι αυτό δεν είναι εμπόδιο για τις μεθόδους που εκτελούν Imputation βρίσκοντας την ιδανική τιμή λαμβάνοντας υπόψη τις υπόλοιπες εγγραφές του πεδίου. Και πράγματι θα μπορούσε να γίνει Imputation στο πεδίο της διάρκειας του Round και στη συνέχεια να υπολογιστεί με κάποια μέθοδο η βαθμολογία για αυτά τα Rounds, δυστυχώς όμως ούτε αυτό είναι εφικτό καθώς σε όλες αυτές τις περιπτώσεις δεν έχει καταγραφεί το Timestamp της λήξης του Round.

Ο τρόπος αναγνώρισης αυτών των εγγραφών, πέρα από τα κενά πεδία είναι η τιμή 5 στο πεδίο game_event, το οποίο ισοδυναμεί με το ότι παρουσιάστηκε κάποιο Exception κατά τη διάρκεια του παιχνιδιού. Ο τρόπος με τον οποίο διαχειρίστηκαν αυτές οι εγγραφές, ήταν με το να μην συμπεριληφθούν στη διαδικασία της μετάπτωσης.

4.2.3.2.2 Encoding

Στο πλαίσιο της διπλωματικής έχει εφαρμοστεί encoding με δύο διαφορετικούς τρόπους, σε επίπεδο βάσης δεδομένων και σε επίπεδο Python Script. Ενώ συνολικά μπορούμε να βρούμε διαδικασίες Encoding σε τρία διαφορετικά στάδια, κατά το ETL, το EDA αλλά και το PMC.

Ο πρώτος τρόπος αφορά το Encoding που έχει γίνει σε επίπεδο βάσης δεδομένων και αφορά μια σειρά από Features και τα υποψήφια Target Classes. Για τα Features, όπως είναι η οικογενειακή κατάσταση, το επίπεδο εκπαίδευσης, το φύλλο, ο βαθμός χρήσης ηλεκτρονικών συσκευών αλλά και οι απαντήσεις που αφορούν τα ιατρικά δεδομένα, έχουν όλα συνδεθεί σε παραμετρικούς πίνακες της μορφής Id, Label, όπου το Id είναι τύπου Long και το Label τύπου Varchar. Οπότε στον πίνακα των Users δεν υπάρχουν αλφαριθμητικά αλλά τα Ids των αντίστοιχων Labels.

Επίσης σε επίπεδο βάσης δεδομένων γίνεται και το Encoding των υποψήφιων Target Classes. Πιο συγκεκριμένα κατά τη διαδικασία του Discretization αυτών των πεδίων, αναλόγως με την αρχική τιμή του πεδίου προσδίδουμε σε αυτό μια αριθμητική τιμή. Για παράδειγμα, έχοντας το πεδίο MOCA_PRE_INIT με τις αρχικές συνεχείς αριθμητικές τιμές, δημιουργούμε το πεδίο MOCA_PRE_BINARY_BINNED με τον παρακάτω κανόνα.

case when (`u`.`moca_pre` >= 0 and `u`.`moca_pre` <= 25) then 1 when (`u`.`moca_pre` >= 26 and `u`.`moca_pre` <= 30) then 2 else 0 end as `moca_pre_binary_binned`

Όπου ουσιαστικά έχουμε ως αποτέλεσμα τη δημιουργία ενός νέου πεδίου, με τιμή 1 για όσους χρήστες είχαν βαθμολογία 0 έως 25 και άρα βρίσκονται στη κατηγορία MCI-AD και τιμή 2 για όσους είχαν βαθμολογία από 26 έως και 30 και άρα βρίσκονται στη κατηγορία NC.

Μια ακόμη λεπτομέρεια αναφορικά με το Encoding έχει να κάνει με το πώς θα γίνεται η ανάκτηση του Label του Target Class, ώστε ο χρήστης να βλέπει ως αποτέλεσμα του Classification “AD-MCI”, “NC” αντί των αριθμητικών τιμών 1, 2. Αυτό πραγματοποιείται στο Classification Service κάνοντας απλά ένα SQL Query και στη συνέχεια την αντικατάσταση πριν την αποστολή του Response στο χρήστη.

Ο δεύτερος τρόπος αφορά το Encoding που γίνεται στο στάδιο των EDA και PMC διαδικασιών, δηλαδή σε επίπεδο Python Script. Όσον αφορά τη διπλωματική, έχει εφαρμοστεί One-Hot-Encoding στο Feature marital_status με χρήση της μεθόδου OneHotEncoder της βιβλιοθήκης του Scikit-learn. Πιο συγκεκριμένα για το marital_status, λόγω του ότι σε αυτό έχουμε εφαρμόσει One-Hot-Encoding, έχουν προκύψει νέα πεδία το καθένα από τα οποία αντιπροσωπεύει μια από τις πιθανές τιμές του. Η αντιστοιχία μεταξύ των πραγματικών τιμών του αρχικού πεδίου και της ονομασίας των πεδίων που προκύπτουν, είναι η εξής, [['Άγαμος', marital_status_0], ['Έγγαμος', marital_status_1], ['Διαζευγμένος', marital_status_2], ['Χήρος', marital_status_3]]. Οι αριθμοί 0 έως 3 δεν είναι τυχαία αλλά αντιπροσωπεύουν το id της εκάστοτε τιμής στον παραμετρικό πίνακα var_marital_status στη βάση δεδομένων.

Τέλος, για ότι Encoding Transformation έχει εφαρμοστεί σε Features τα οποία τελικά θα χρησιμοποιηθούν για το παραγωγικό μοντέλο, συμπεριλαμβάνουμε τα Instances αυτών των Transformers στη διαδικασία PMC, ώστε τα νέα δεδομένα να υφίστανται τους ίδιους μετασχηματισμούς.

4.2.3.2.3 Handling Outliers

Όσον αφορά τη διπλωματική έχει δημιουργηθεί η μέθοδος handle_outliers στο αρχείο modulePreProcessing, η οποία δέχεται ως παράμετρο ένα Dataframe, υπολογίζει τις τιμές Q1, Median, Q3,

IQR, Min, Max, ενώ στη συνέχεια για κάθε Feature του Dataframe εφαρμόζει μια διαφορετική στρατηγική για τη διαχείριση των Outliers.

Η συνήθης πρακτική περιλαμβάνει την αφαίρεση εγγραφών που παρουσιάζουν Outliers σε Features που μας ενδιαφέρουν. Ωστόσο, δεδομένου του ότι το μέγεθος του Dataset που έχουμε στη διάθεση μας είναι σχετικά μικρό, για να αποφύγουμε τη περαιτέρω μείωση εγγραφών, αντί της αφαίρεσης επιχειρήθηκε η μεταβολή των τιμών των Outliers. Οι στρατηγικές μεταβολής των τιμών έχουν ως εξής.

Για τα Features που αποτελούν κάποιο σύνολο, όπως για παράδειγμα το σύνολο των γύρων που έχει παίξει ένας User κατά τη διάρκεια ενός Session, εάν μια τιμή βρίσκεται πέραν του Max της εκάστοτε κατανομής, τότε η τιμή μεταβάλλεται στη τιμή που έχει το Q3 κατά την έναρξη της διαδικασίας. Όπως φαίνεται και στα παρακάτω διαγράμματα, που αφορούν τα Features του Session, ο συνολικός αριθμός κερδισμένων πόντων σε ένα Session παρουσιάζει μερικές αρκετά ακραίες τιμές οι οποίες οφείλονται στο ότι σε ένα Session ο User είχε το δικαίωμα να παίξει όσους γύρους επιθυμούσε.

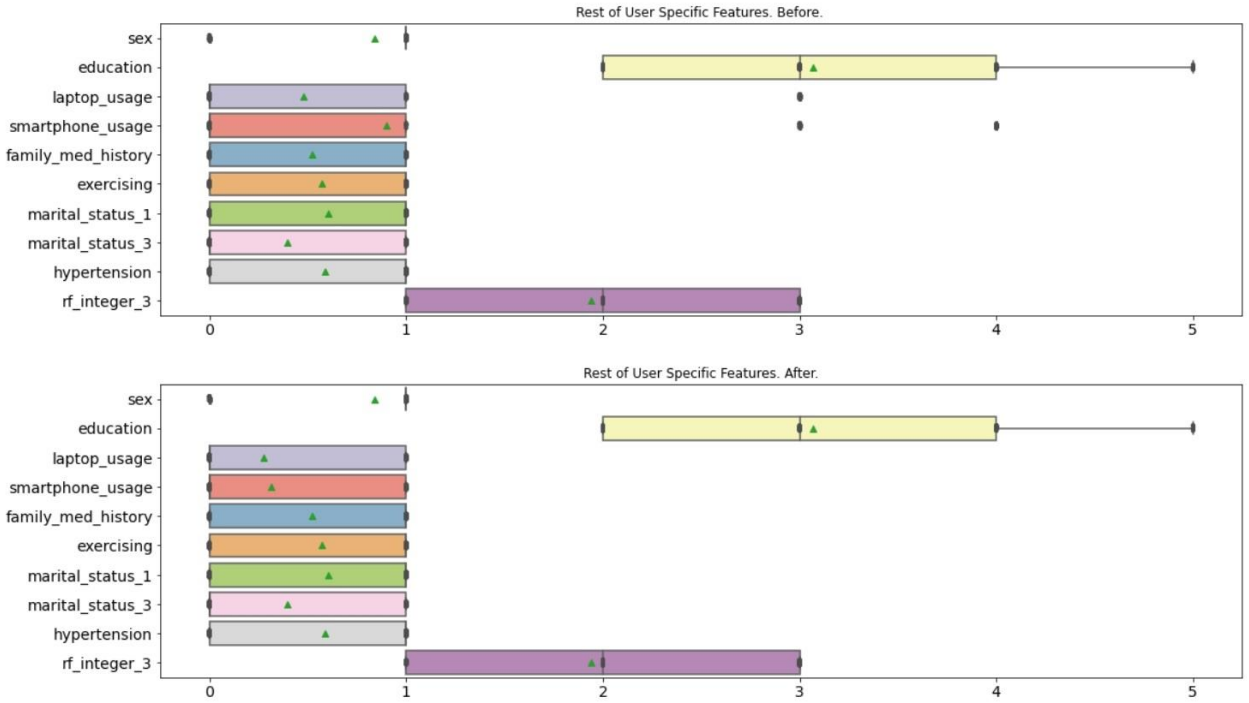
Για τα Features που αποτελούν κάποιο μέσο όρο, όπως για παράδειγμα ο μέσος όρος ενός γύρου για ένα Session, ως στρατηγική επιλέχθηκε η αντικατάσταση των τιμών που είναι εκτός ορίων Min και Max με την τιμή που βρίσκεται στο Q2 δηλαδή τη διάμεση (Median) τιμή της κατανομής. Και σε αυτή τη περίπτωση η τιμή με την οποία αντικαθιστούμε αφορά τη τιμή στην αρχή της διαδικασίας.

Για τα Features που αφορούν τον User ακολουθήθηκε παρόμοια στρατηγική με αυτή για τα Totals, δηλαδή οι εκτός ορίων Min και Max τιμές αντικαταστάθηκαν με τις τιμές των Q1 και Q3 αντίστοιχα για το εκάστοτε Feature. Επιπλέον για τη συγκεκριμένη ομάδα από Features παρατηρούμε ότι μόνο για τη χρήση Laptop και Smartphone έχουμε κάποιες εγγραφές που μπορούν να θεωρηθούν Outliers.

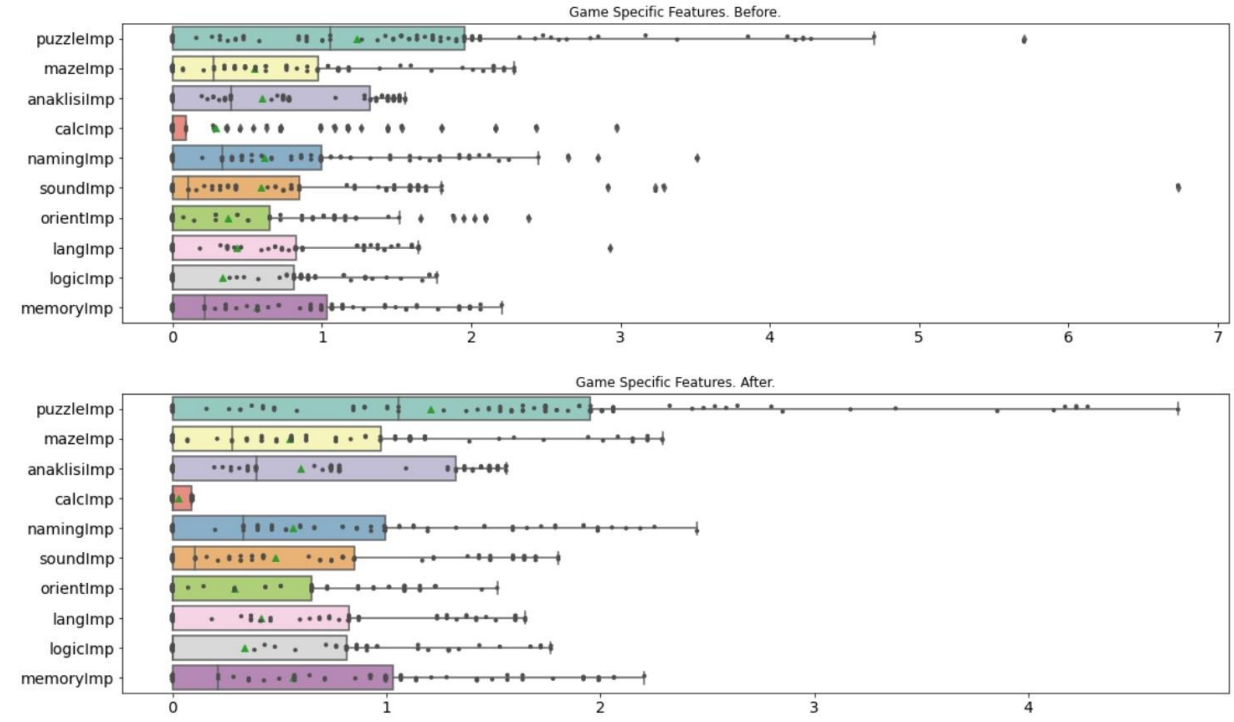
Σε αυτό το σημείο θα πρέπει να διευκρινιστεί ότι οι τιμές με τις οποίες αντικαθιστούμε είναι οι τιμές όπως προκύπτουν κατά την έναρξη της διαδικασίας και δεν γίνεται επανυπολογισμός των στατιστικών μεγεθών Min, Max, Q1, Q2, Q3, με κάθε μεταβολή, δεδομένου του ότι, η διαφοροποίηση στη κατανομή που μπορεί να προκαλέσει μια αντικατάσταση είναι πολύ μικρή.

Στα διαγράμματα των παρακάτω εικόνων βλέπουμε τη κατανομή των Features ομαδοποιημένα σε User και Session Specific Features, πριν και μετά τη διαχείριση των Outliers.

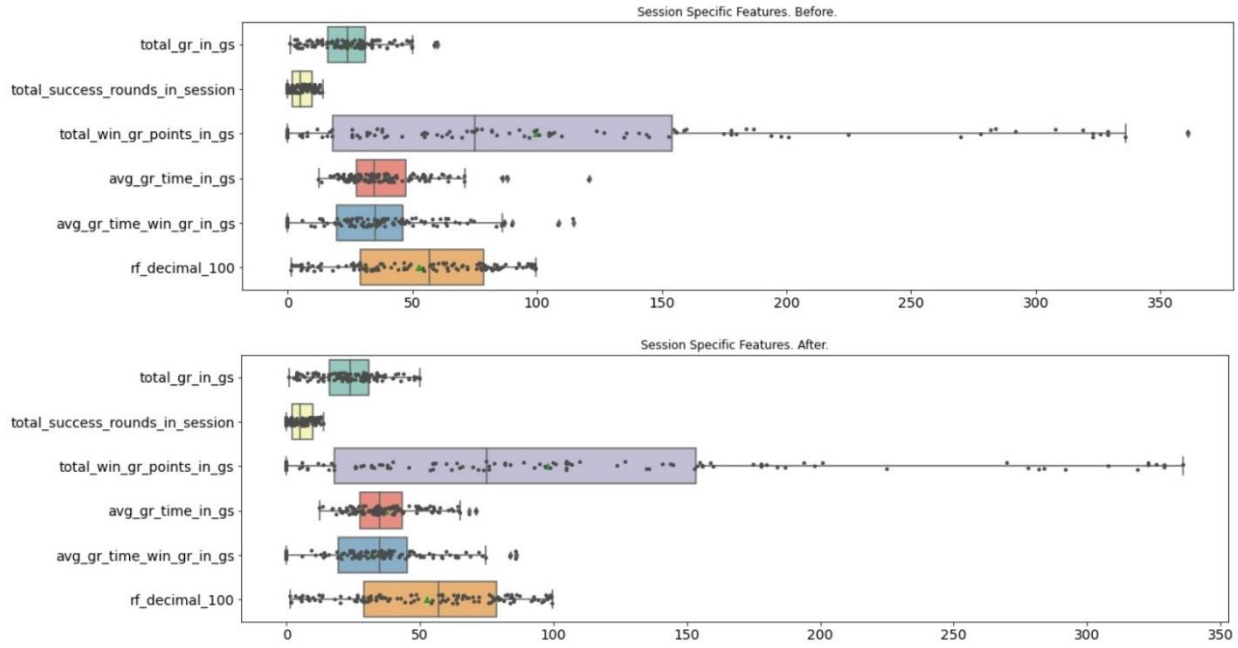
Εικόνα 20 Κατανομή τιμών ηλικίας των χρηστών, πριν και μετά την εκτέλεση της μεθόδου αφαίρεσης των Outliers.



Εικόνα 21 Κατανομή τιμών για User Specific Features καθώς και για ένα Feature με τυχαίες ακέραιες τιμές από 1 έως 3.



Εικόνα 22 Κατανομή τιμών για Game Specific Features



Εικόνα 23 Κατανομή τιμών για Session Specific Features καθώς και για ένα Feature με τυχαίες δεκαδικές τιμές από 0 έως 100

Τέλος όσον αφορά το Outlier Handling, το πλεονέκτημα της εφαρμογής της διαδικασίας και της αφαίρεσης αυτών των ακραίων τιμών γίνεται πολύ περισσότερο εμφανές στη διαδικασία του Scaling όπου μπορούμε να συγκρίνουμε πλέον τη κατανομή του κάθε Feature, μετά από Scaling, με ή χωρίς Outliers, όπως παρουσιάζεται και στην αντίστοιχη ενότητα.

4.2.3.2.4 Discretization

Στη περίπτωση της παρούσας διπλωματικής, έχει εφαρμοστεί Discretization τόσο στο επίπεδο της βάσης δεδομένων όσο και σε επίπεδο Python Script και συγκεκριμένα κατά τη διαδικασία της EDA.

Στο επίπεδο της βάσης δεδομένων, για κάθε ένα από τα υποψήφια Target Class, έχει εφαρμοστεί Custom Discretization σε διακριτά επίπεδα που αντιπροσωπεύουν τις κατηγορίες στις οποίες μπορούν να ομαδοποιηθούν τα αποτελέσματα των νευροψυχολογικών δοκιμασιών MOCA και MMSE. Πιο συγκεκριμένα, για κάθε Session, έχοντας ως αρχική πληροφορία το αποτέλεσμα της εκάστοτε δοκιμασίας που αντιστοιχεί στο χρήστη του Session σε συνεχείς τιμές, παράγουμε τα επιπλέον πεδία με τις τιμές στα διακριτά επίπεδα. Τα πεδία που παράγουμε για κάθε μια από τις δοκιμασίες είναι δύο, ένα με τις τιμές σταθμισμένες σε δύο επίπεδα (MCI, No-MCI) και ένα με τις τιμές σταθμισμένες σε επίπεδα που αντιπροσωπεύουν τη σοβαρότητα της έκπτωσης των γνωστικών λειτουργιών (AD, MCI, NC). Ο τρόπος με τον οποίο έγινε είναι με την εντολή CASE στο SELECT Clause του SQL View από το οποίο ανακτούμε το Dataset κάθε φορά που θέλουμε να τρέξουμε τη διαδικασία της EDA. Επίσης στο επίπεδο της βάσης δεδομένων, μπορούμε να εφαρμόσουμε αντίστοιχα Discretization για το κάθε Numerical Feature με συνεχείς τιμές, όπως για παράδειγμα η ηλικία. Ωστόσο, αυτό θα είχε δύο μειονεκτήματα, πρώτον δεν θα ήταν εφικτό να κάνουμε την αντίστροφη διαδικασία, δηλαδή το Discretization για τις τιμές των νέων δεδομένων, ενώ δεύτερον θα ήμασταν περιορισμένοι στο να εφαρμόσουμε είτε Equal Width είτε Custom Discretization.

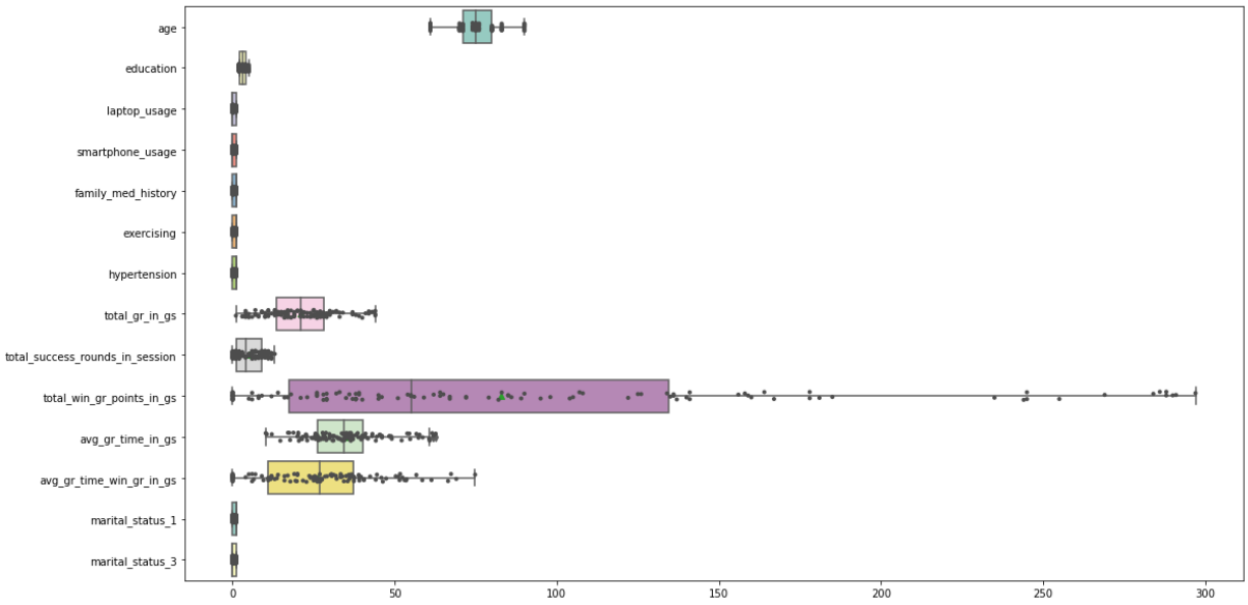
Στο επίπεδο του κώδικα, έχει εφαρμοστεί για την ηλικία και όλα τα Session Specific Features, τα οποία είναι στο σύνολο τους Numerical με συνεχείς τιμές. Η μέθοδος που έχει χρησιμοποιηθεί είναι η KBinsDiscretizer της βιβλιοθήκης Scikit-learn, με παραμέτρους Ordinal και Quantile. Ordinal έτσι ώστε τα αποτελέσματα να επιστρέψουν ως ένα πεδίο με τη κάθε στάθμη να είναι ένας διαφορετικός ακέραιος. Quantile έτσι ώστε να έχουμε εφαρμογή Equal Frequency Discretization. Σημαντικό στοιχείο εδώ είναι το ότι τρέχουμε τη διαδικασία μετά τη διαχείριση των Outliers έτσι ώστε αυτές οι τιμές να μην επηρεάσουν στο ελάχιστο τις στάθμες που θα προκύψουν. Στη παρακάτω εικόνα βλέπουμε δύο πίνακες με τις τιμές των συγκεκριμένων Features πριν και μετά την εφαρμογή του Discretization.

gsId	age	avg_gr_time_win_gr_in_gs	avg_gr_time_win_gr_in_gs	avg_gr_time_in_gs	total_win_gr_points_in_gs	total_gr_in_gs	total_success_rounds_in_session
18245	70.0	17.0	17.0	14.7143	33.0	7.0	1.0
76067	61.0	53.0	53.0	41.7568	67.0	37.0	4.0
79781	83.0	51.5	51.5	38.1818	17.0	11.0	4.0
80144	71.0	4.0	4.0	34.4324	39.0	37.0	1.0
09758	75.0	11.0	11.0	29.9583	26.0	24.0	1.0

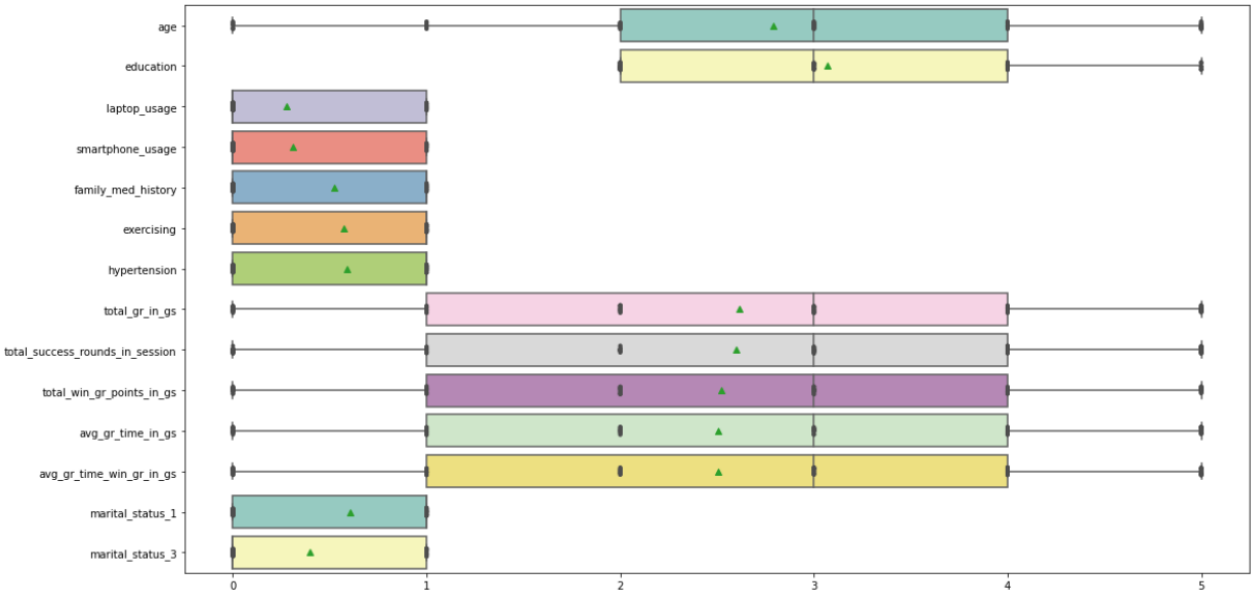
gsId	age	avg_gr_time_win_gr_in_gs	avg_gr_time_win_gr_in_gs	avg_gr_time_in_gs	total_win_gr_points_in_gs	total_gr_in_gs	total_success_rounds_in_session
18245	1	2	2	0	2	0	1
76067	0	5	5	4	3	5	3
79781	5	5	5	4	1	1	3
80144	2	1	1	3	2	5	1
09758	3	1	1	2	1	3	1

Εικόνα 24 Ενδεικτικό αποτέλεσμα της διαδικασίας Discretization για τα πεδία στα οποία έχει εφαρμοστεί

Σε αυτό το σημείο έχει ενδιαφέρον να δούμε δύο γραφήματα που αφορούν τη κατανομή, πριν και μετά, για τα Features στα οποία εφαρμόσαμε Discretization. Εκ πρώτης όψεως θα μπορούσαμε να πούμε ότι εφαρμόζοντας Discretization το αποτέλεσμα είναι παρόμοιο με το αποτέλεσμα μετά από τη διαδικασία του Scaling, που θα αναλύσουμε παρακάτω, ωστόσο αυτό συμβαίνει διότι έχουμε κάνει επιλεκτικά Discretize τα Features που έχουν μεγάλο Range τιμών σε μικρό αριθμό από στάθμες. Οπότε φαινομενικά οι τιμές όλων των Features δείχνουν να βρίσκονται σε κοντινή κλίμακα, αλλά ουσιαστικά η κατανομή που προκύπτει δεν έχει κανένα από τα ποιοτικά χαρακτηριστικά μιας κατανομής που προκύπτει από διαδικασία Scaling.



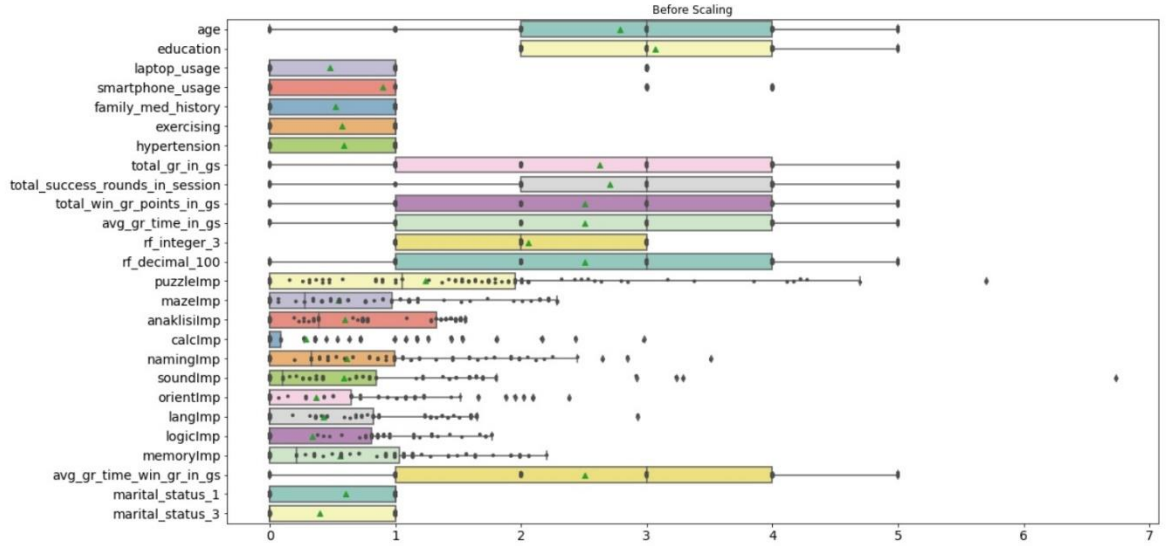
Εικόνα 25 Κατανομή των Features πριν από την εφαρμογή Discretization



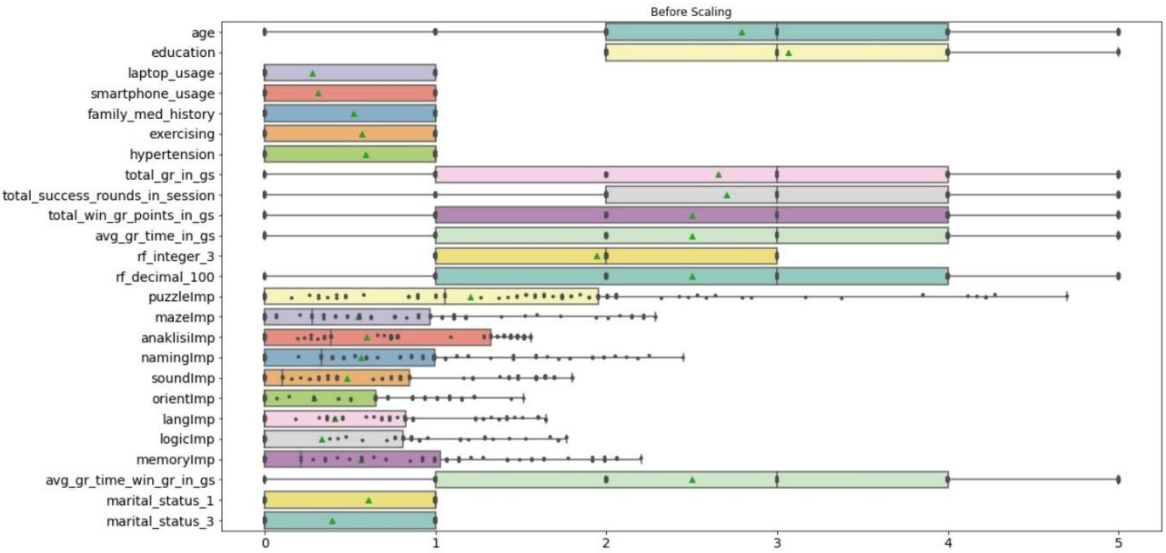
Εικόνα 26 Κατανομή των Features μετά από την εφαρμογή Discretization

4.2.3.2.5 Scaling

Στο πλαίσιο της διπλωματικής εφαρμόστηκαν οι δύο βασικές μέθοδοι για Scaling, Normalization και Standardization, εφαρμόζοντας τις μεθόδους fit_transform των κλάσεων MinMaxScaler και StandardScaler αντίστοιχα, της βιβλιοθήκης Scikit-learn. Σε όλα τα διαγράμματα που παρατίθενται στις παρακάτω εικόνες έχει προηγηθεί Discretization στα Features που αφορούν τα Sessions των χρηστών.

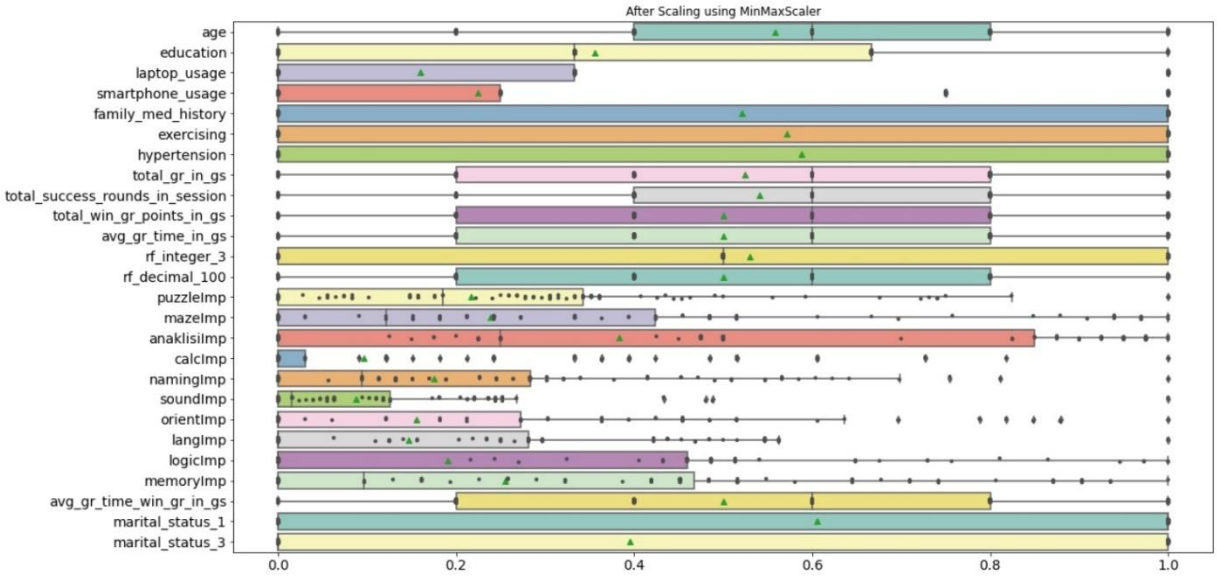


Εικόνα 27 Κατανομές των Features πριν την εφαρμογή Scaling, χωρίς την αφαίρεση των Outliers

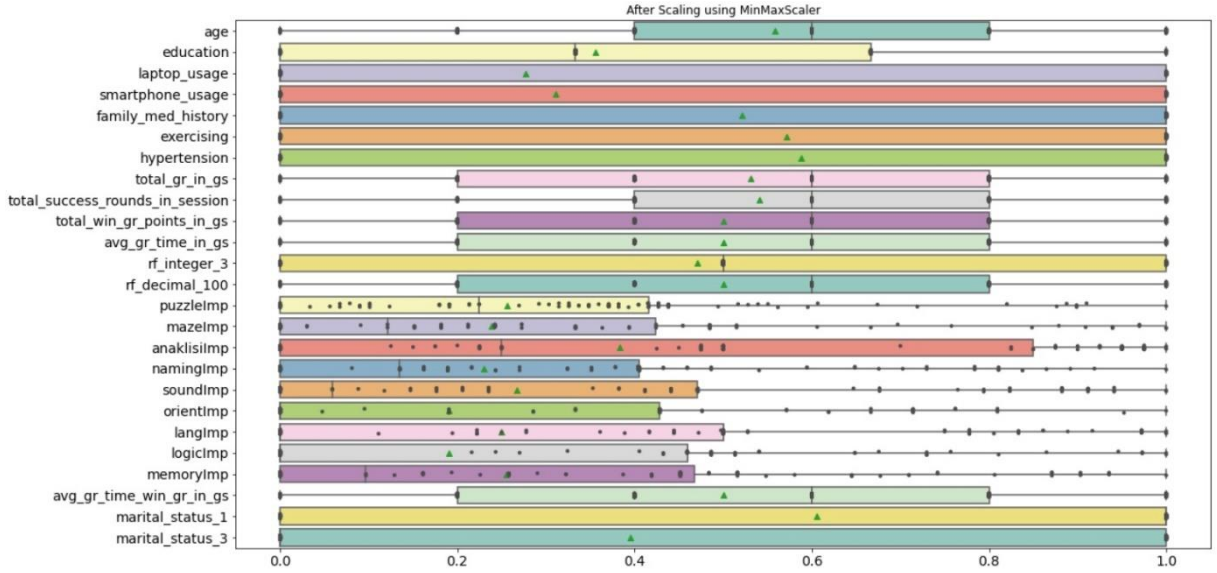


Εικόνα 28 Κατανομές των Features πριν την εφαρμογή Scaling, έχοντας αφαιρέσει τις Outlier τιμές

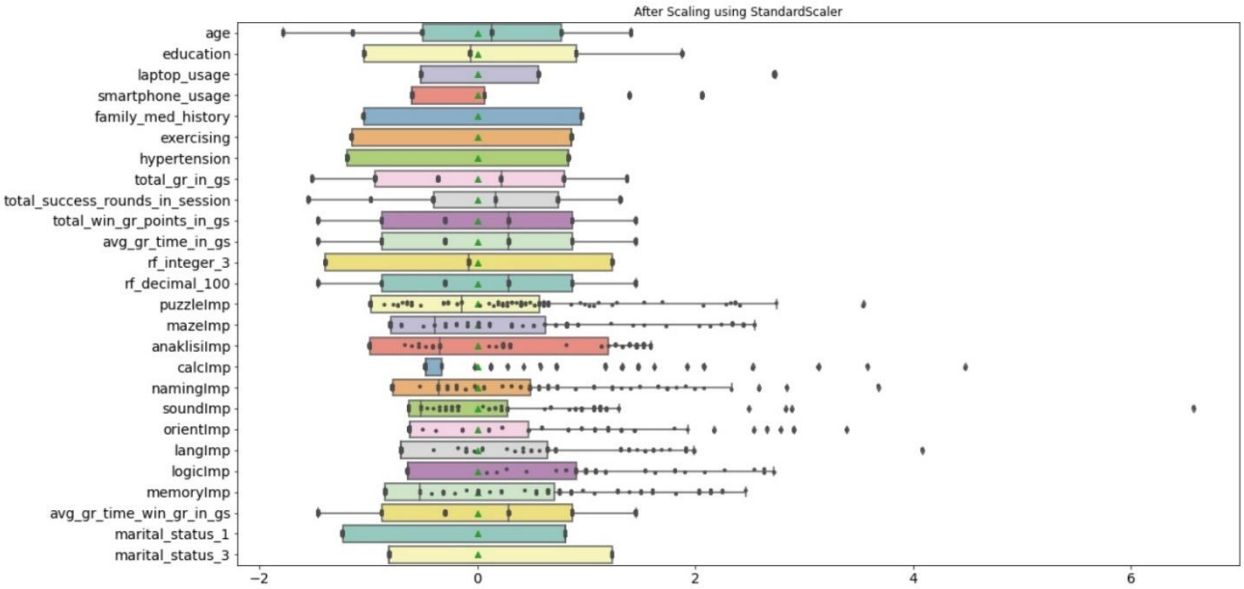
Παρατηρούμε σε αυτό το σημείο πρώτον, το χαρακτηριστικό του MinMaxScaler που είναι το συγκεκριμένο Range που έχουν πλέον οι κατανομές όλων των Feature και δεύτερον, τη χρησιμότητα της αφαίρεσης των Outliers. Συγκεκριμένα βλέπουμε στα Features που έχουν να κάνουν με το πόσο εξοικειωμένοι είναι οι χρήστες στη χρήση Smartphone και Laptop ότι οι κατανομές έχουν γίνει κανονικές αφαιρώντας τις Outlier τιμές. Επίσης παρατηρούμε ότι λόγω του Discretization που εφαρμόζουμε το οποίο είναι τύπου Equal Frequency, οι Outlier τιμές συμπεριλαμβάνονται στα ακραία Bins, πράγμα που σημαίνει πως ακόμη και αν δεν διαχειριστούμε τους Outliers, επειδή έχουμε κάνει Discretization θα έχουμε ακραίες τιμές μόνο στα Features των χρηστών όπως φαίνεται και στο διάγραμμα της εικόνας 27.



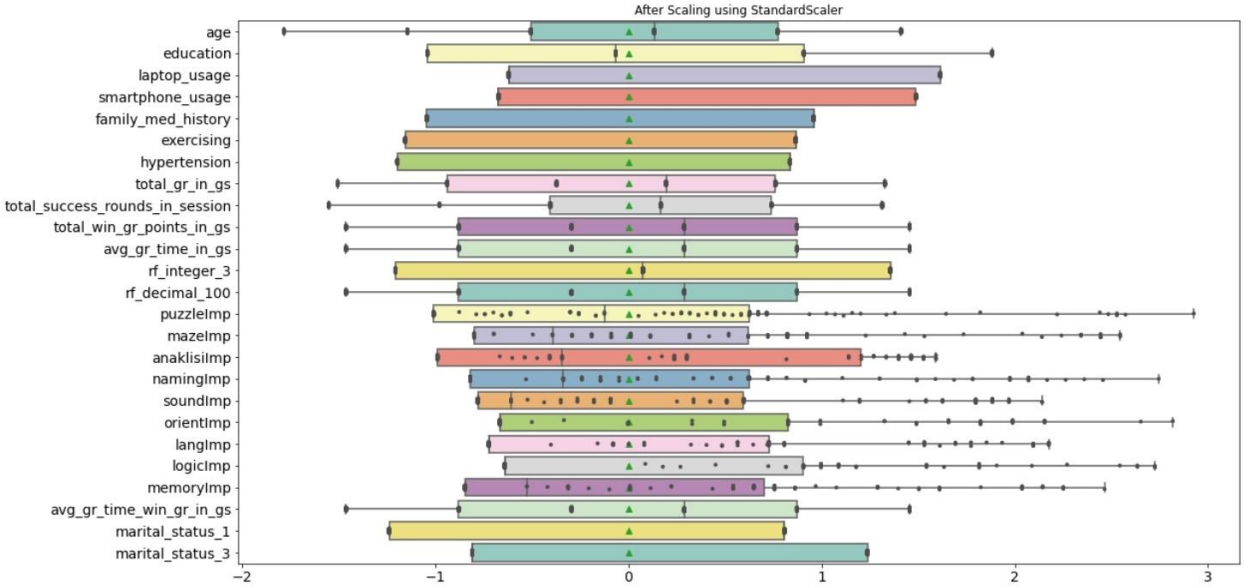
Εικόνα 29 Μετά την εφαρμογή Scaling με MinMaxScaler συμπεριλαμβανομένων των Outliers



Εικόνα 30 Μετά την εφαρμογή Scaling με MinMaxScaler χωρίς Outliers



Εικόνα 31 Μετά την εφαρμογή Scaling με StandardScaler συμπεριλαμβανομένων των Outliers



Εικόνα 32 Μετά την εφαρμογή Scaling με StandardScaler χωρίς Outliers

Στα διαγράμματα που αφορούν το Transformation με StandardScaler παρατηρούμε το χαρακτηριστικό αποτέλεσμα της μετατόπισης της μέσης τιμής (πράσινο τρίγωνο) στη τιμή 0. Αυτό για παράδειγμα στη κατανομή ενός Feature συμβαίνει γιατί από κάθε τιμή της κατανομής αφαιρούμε τη μέση τιμή και διαιρούμε με το Standard Deviation. Για αυτό το λόγο και η μονάδα μέτρησης των τιμών στον άξονα x'x, μετά από Standardization, αναφέρεται και ως Standard Deviation Unit [25].

4.2.3.3 Feature Selection

Σε αυτή την ενότητα θα εξετάσουμε τη διαδικασία της επιλογής των Features που θα χρησιμοποιήσουμε για την δημιουργία τόσο των πειραματικών μοντέλων στη διαδικασία της EDA, όσο και του παραγωγικού μοντέλου.

Ως είσοδο, στη διαδικασία Feature Selection, έχουμε ένα Dataframe με τα δεδομένα όπως είναι κατόπιν της εφαρμογής των μεθόδων Preprocessing στο πρώτο στάδιο της διαδικασίας EDA. Ως έξοδο, θα μπορούσαμε είτε να εξάγουμε αυτοματοποιημένα μια λίστα με τα ονόματα των επιλεγμένων Features, είτε να δημιουργήσουμε χειροκίνητα μια λίστα με αυτά που θεωρούμε ως τα ιδανικά Features.

Πριν όμως ξεκινήσουμε να αναλύουμε την κάθε μέθοδο, θα πρέπει να αναφερθούμε σε μια σημαντική πτυχή που αφορά το Business Logic της υλοποίησης. Συγκεκριμένα, θα πρέπει να διασφαλίσουμε ότι ανάμεσα στα Features που θα επιλέξουμε τελικά, θα πρέπει να βρίσκονται και Features που αφορούν τα Game Sessions που έπαιξαν οι χρήστες της εφαρμογής MCI Rehab και όχι μόνο τα Features που έχουν να κάνουν με τα στοιχεία των χρηστών.

Σε αυτό το σημείο θα πρέπει επίσης να αναφερθεί ότι υπάρχουν πολλές μέθοδοι αυτοματοποιημένης επιλογής κατάλληλων Features, με τις αντίστοιχες υλοποιήσεις τους, με τις πιο γνωστές από αυτές να είναι οι Forward Selection, Backward Elimination, Recursive Feature Elimination και επιλογή με βάση την υψηλότερη βαθμολογία σε κάποια στατιστική τιμή, όπως είναι για παράδειγμα το F-Score. Όλες αυτές οι μέθοδοι περιλαμβάνουν τη δημιουργία κάποιου μοντέλου, ώστε σε κάθε επανάληψη της διαδικασίας να υπάρχουν Metrics (Scorer Functions) για το μοντέλο βάση των οποίων αξιολογούνται τα Features.

Για την υλοποίηση της διπλωματικής χρησιμοποιήθηκε τόσο μια Custom λογική για την επιλογή μιας λίστας από Features, όσο και μια αυτοματοποιημένη μέθοδος, όπως θα δούμε στην ενότητα Feature Inspection. Ο λόγος που επιλέχθηκαν και οι δύο μέθοδοι είναι έτσι ώστε να υπάρχει μια διασταύρωση των αποτελεσμάτων για τα μοντέλα που έχουν δημιουργηθεί με Manually επιλεγμένα Features με αυτά των μοντέλων που έχουν δημιουργηθεί με αυτόματη επιλογή.

Η ακολουθία των βημάτων για να καταλήξουμε στο ιδανικό Set από Features, είναι η εξής.

- Πρώτον, έχουμε ήδη δημιουργήσει ότι επιπλέον Feature μπορεί να προκύψει από τα αρχικά δεδομένα σε επίπεδο βάσης δεδομένων, ενώ παράλληλα στο ίδιο επίπεδο έχουμε δημιουργήσει και κάποια με τυχαίες τιμές (Feature Engineering).
- Στη συνέχεια αφαιρούμε τα Features που έχουν χαμηλό Variance και δεν θα μπορούσαν να προσθέσουν κάποια αξία στο μοντέλο (Low Variance Features Removal).
- Έπειτα πραγματοποιούμε μια επισκόπηση των συσχετίσεων των Features μεταξύ τους αλλά και μεταξύ αυτών και του Target Class, έτσι ώστε να γνωρίζουμε ποια θεωρούνται σημαντικά για μια σειρά από Metrics που αφορούν Features, αλλά και για να γνωρίζουμε ποια έχουν υψηλή συσχέτιση μεταξύ τους (Feature Inspection).
- Ενώ στο πλαίσιο του Optimization θα εφαρμόσουμε Dimensionality Reduction, όπου τα επιλεγμένα, μέχρι εκείνη τη στιγμή, Features θα μετασχηματιστούν σε ένα Set από Principal Components, όπως θα δούμε αναλυτικά στην αντίστοιχη ενότητα.

Το ιδανικό αποτέλεσμα, αυτής της διαδικασίας για τη παρούσα υλοποίηση, θα ήταν ένα Set που θα συμπεριελάμβανε τόσο Features του υποκειμένου από το ιατρικό και εκπαιδευτικό του προφίλ όσο και Features από τις επιδόσεις που κατέγραψε στα παιχνίδια σοβαρού σκοπού. Επίσης θα πρέπει τα επιλεγμένα Features να έχουν μεταξύ τους όσο το δυνατόν μικρότερη συσχέτιση, ενώ παράλληλα θα πρέπει να έχουν όσο το δυνατόν μεγαλύτερη συσχέτιση με το Target Class, κάτι που εξετάζουμε με τις διαδικασίες Feature Correlation και Feature Importance Inspection αντίστοιχα.

4.2.3.3.1 Feature Engineering

Μια βασική διαδικασία που αφορά την ενότητα Feature Selection είναι η διαδικασία της δημιουργίας επιπλέον Features (Feature Engineering) που προκύπτουν από επεξεργασία των ήδη υπαρχόντων με σκοπό τη βελτιστοποίηση της απόδοσης του μοντέλου σε νέες προβλέψεις. Η διαδικασία αυτή περιλαμβάνει αριθμητικούς και αθροιστικούς μετασχηματισμούς για τη παραγωγή νέων Features. Οι μετασχηματισμοί αυτοί βοηθούν στο να δημιουργούνται πολλά νέα Features τα οποία στη συνέχεια εξετάζονται για το κατά πόσο γραμμική είναι η συσχέτιση τους με το Target Class [27].

Υπάρχουν πολλές διαφορετικές μέθοδοι για τη δημιουργία νέων Features, μερικές από τις πιο γνωστές είναι μέθοδοι τύπου, Brute Force, Model Evaluation, Random και Learning Feature Engineering (LFE). Όλες αυτές οι μέθοδοι έχουν το κοινό χαρακτηριστικό της δημιουργίας πληθώρας νέων Feature τα οποία στη συνέχεια περνάνε από διαλογή μέσω διαδικασιών Feature Selection και πολλές φορές, εάν ο αριθμός τους παραμένει υψηλός, από επιπλέον διαδικασίες τύπου Dimensionality Redaction.

Ωστόσο για την υλοποίηση της διπλωματικής δημιουργήθηκε μια σειρά από Features με στοχευμένες ιδιότητες, τα οποία θα εξετάσουμε στη συνέχεια στη διαδικασία του Feature Inspection για το κατά πόσο σημαντικά είναι στην δημιουργία ενός μοντέλου με το επιλεγμένο Target Class.

Επειδή ένας από τους στόχους είναι η δημιουργία ενός παραγωγικού μοντέλου και όχι απλά να περιοριστούμε στο διερευνητικό επίπεδο της EDA, θα πρέπει να έχουμε κατά νου πως ότι νέο Feature κατασκευάζουμε, εάν τελικά συμπεριληφθεί στην εκπαίδευση του παραγωγικού μοντέλου, θα πρέπει να μπορεί να εξαχθεί από τα στοιχεία ενός Session που μελλοντικά θα φτάσει για Classification στο Service σε ένα πραγματικό σενάριο.

Τα Features που έχουμε δημιουργήσει μπορούν να χωριστούν σε δύο κύριες κατηγορίες, πρώτον αυτά που προκύπτουν ως Ratios από τα ήδη υπάρχοντα πεδία και δεύτερον αυτά που έχουμε δημιουργήσει με τυχαίες τιμές.

Για τα Features με τυχαίες τιμές έχουμε ως στόχο να τα χρησιμοποιήσουμε ως σημεία αναφοράς για τη σημαντικότητα ενός Feature. Εάν για παράδειγμα μέσα από τη διαδικασία του Feature Inspection παρατηρήσουμε κάποια από τα Features να είναι λιγότερο σημαντικά από ένα Feature με τυχαίες τιμές τότε τα συγκεκριμένα δεν υπάρχει λόγος να τα χρησιμοποιήσουμε στην εκπαίδευση του μοντέλου. Τα Features με τυχαίες τιμές που δημιουργήθηκαν είναι τα εξής.

- Όνομα πεδίου: rf_integer_3, Τύπος: Integer, Εύρος τιμών 1 έως 3
- Όνομα πεδίου: rf_decimal_100 Τύπος: Decimal, Εύρος τιμών x>=0, x<100

Για τα Features που προκύπτουν από υπάρχοντα πεδία, αυτά είναι συνολικά 16 και αναλύονται παρακάτω.

- Όνομα πεδίου: total_gr_in_gs, Τύπος: Integer. Σύνολο Game Rounds σε ένα Game Session
- Όνομα πεδίου: total_success_rounds_in_session, Τύπος: Integer, Σύνολο κερδισμένων Game Rounds σε ένα Game Session
- Όνομα πεδίου: total_win_gr_points_in_gs, Τύπος: Integer, Σύνολο πόντων σε κερδισμένα Game Rounds σε ένα Game Session
- Όνομα πεδίου: avg_gr_time_in_gs, Τύπος: Decimal, Μέσος χρόνος ενός Game Round σε ένα Game Session

- Όνομα πεδίου: avg_gr_time_win_gr_in_gs, Τύπος: Decimal, Μέσος χρόνος ενός κερδισμένου Game Round σε ένα Game Session
- Όνομα πεδίου: ΟΝΟΜΑ_ΠΑΙΧΝΙΔΙΟΥ+'Imp', Τύπος: Decimal, Τέλος μπορούμε να ομαδοποιήσουμε τα 10 Features που δημιουργήθηκαν με στόχο να μετρήσουν το πόσο σημαντικό είναι ένα παιχνίδι. Λόγω του Aggregation που γίνεται σε επίπεδο Session δεν θα μπορούσαμε να έχουμε αυτή τη πληροφορία σε μια στήλη, οπότε έχει δημιουργηθεί ένα ξεχωριστό πεδίο για κάθε παιχνίδι. Το κάθε ένα από αυτά τα Features αποτελεί ένα Ratio το οποίο υπολογίζεται ως εξής.

$$\text{Game Importance} = \frac{\text{Σύνολο πόντων για κερδισμένα Game Rounds του συγκεκριμένου παιχνιδιού στο Session μιας εγγραφής}}{\text{Μέσος όρος πόντων που παρουσιάζει σε κερδισμένα Game Rounds το συγκεκριμένο παιχνίδι σε όλα τα Sessions}}$$

Όλα τα παραπάνω Features δημιουργούνται στο επίπεδο της βάσης δεδομένων και συγκεκριμένα στο Select Clause του SQL View που χρησιμοποιούμε για να διαβάσουμε τα δεδομένα μας στη διαδικασία της EDA.

Δεδομένου του ότι κάνουμε Select σε επίπεδο Game Session, κάνοντας παράλληλα Join τα Game Rounds μέσω του session_id, αυτό μας δίνει τη δυνατότητα να κάνουμε Aggregate, δηλαδή μετασχηματισμούς όπως AVG/COUNT/SUM, σε στοιχεία που αφορούν τα Game Rounds, προβάλλοντας έτσι το αποτέλεσμα στο επίπεδο του Game Session.

Στη συνέχεια ακολουθούν δύο πίνακες όπου παρουσιάζονται τα ποσοτικά (Quantile) και τα περιγραφικά (Descriptive) στατιστικά αντίστοιχα, για τα Engineered Features που αναφέρθηκαν προηγουμένως.

Feature	Distinct Values	Quantile Statistics								
		Min	5th Percentile	Q1	Median	Q3	95th Percentile	Max	Range	IQR
total_gr_in_gs	46	1	4	16	24	31	48	60	59	15
total_success_rounds_in_session	15	0	0	2	5	10	12	14	14	8
total_win_gr_points_in_gs	80	0	0	18	75	154	323	361	361	136
avg_gr_time_in_gs	118	12.448	20.385	27.620	34.760	47.314	65.469	120.800	108.351	19.694
avg_gr_time_win_gr_in_gs	96	0.000	0.000	19.636	35.000	46.250	84.020	114.500	114.500	26.613
rf_decimal_100	119	1.140	6.093	24.322	51.228	74.437	93.357	97.827	96.687	50.114
puzzleImp	44	0.000	0.000	0.000	1.056	1.953	4.123	5.702	5.703	1.953
mazeImp	26	0.000	0.000	0.000	0.277	0.971	2.150	2.289	2.289	0.971
anaklisiImp	20	0.000	0.000	0.000	0.389	1.324	1.519	1.558	1.558	1.324
calcImp	18	0.000	0.000	0.000	0.000	0.090	1.560	2.976	2.976	0.090
namingImp	31	0.000	0.000	0.000	0.331	0.993	2.126	3.510	3.510	0.993
soundImp	26	0.000	0.000	0.000	0.106	0.848	1.707	6.734	6.734	0.848
orientImp	22	0.000	0.000	0.000	0.000	0.650	1.683	2.384	2.384	0.650
langImp	23	0.000	0.000	0.000	0.000	0.824	1.602	2.930	2.930	0.824
logicImp	20	0.000	0.000	0.000	0.000	0.812	1.443	1.768	1.768	0.812
memoryImp	25	0.000	0.000	0.000	0.213	1.031	1.927	2.204	2.204	1.031

Πίνακας 3 Ποσοτικά στατιστικά για τα Engineered Features

Feature	Distinct Values	Descriptive Statistics						
		STD	Coeff. Of Variation	Kurtosis	Mean	Median Abs. Dev.	Skewness	Variance
total_gr_in_gs	46	12.730	0.513	-0.044	24.798	8	0.347	162.060
total_success_rounds_in_session	15	4.488	0.806	-1.468	5.563	4	0.230	20.146
total_win_gr_points_in_gs	80	97.787	0.982	0.289	99.529	63	1.073	9562.420
avg_gr_time_in_gs	118	16.395	0.422	4.929	38.828	8.527	1.666	268.799
avg_gr_time_win_gr_in_gs	96	24.752	0.714	0.560	34.665	13.750	0.638	612.710
rf_decimal_100	119	28.731	0.579	-1.222	49.550	24.837	-0.066	825.518
puzzleImp	44	1.266	1.024	0.896	1.235	1.003	1.031	1.604
mazeImp	26	0.687	1.258	0.398	0.546	0.277	1.182	0.472
anaklisiImp	20	0.606	1.016	-1.434	0.596	0.389	0.426	0.368
calcImp	18	0.603	2.110	5.178	0.285	0.000	2.343	0.363
namingImp	31	0.790	1.288	1.072	0.613	0.331	1.295	0.624
soundImp	26	0.938	1.595	15.207	0.588	0.106	3.071	0.880
orientImp	22	0.596	1.613	1.531	0.367	0.000	1.553	0.355
langImp	23	0.614	1.434	1.246	0.428	0.000	1.334	0.378
logicImp	20	0.527	1.569	0.363	0.336	0.000	1.280	0.278
memoryImp	25	0.668	1.188	-0.546	0.563	0.213	0.837	0.447

Πίνακας 4 Περιγραφικά στατιστικά για τα Engineered Features

4.2.3.3.2 Low Variance Features Removal

Στο πλαίσιο της διπλωματικής για να αφαιρέσουμε τα Features με χαμηλό ή μηδενικό Variance έγιναν τα παρακάτω βήματα.

Πρώτον, απεικονίσαμε το Variance για κάθε Feature, σε ένα πίνακα αλλά και σε ένα Barplot, υπολογίζοντας τις τιμές με τη μέθοδο var της βιβλιοθήκης Pandas και στη συνέχεια, χρησιμοποιήσαμε τη μέθοδο fit_transform της κλάσης VarianceThreshold της βιβλιοθήκης του Scikit-learn, για να αποκλείσουμε τα Features που βρίσκονται κάτω από το Threshold που ορίσαμε.

Τόσο η μέθοδος var όσο και η fit_transform χρησιμοποιούν τη μέθοδο nanvar της βιβλιοθήκης Numpy για τον υπολογισμό του Variance. Αυτό σημαίνει πως οι τιμές που έχουμε τυπώσει στον πίνακα είναι αυτές που χρησιμοποιεί η μέθοδος fit για να αποκλείσει τα Features με βάση το Threshold. Η μέθοδος που χρησιμοποιεί η nanvar, περιγράφεται από την παρακάτω μέθοδο.

```
var = mean( abs( x - x.mean() ) **2 )
```

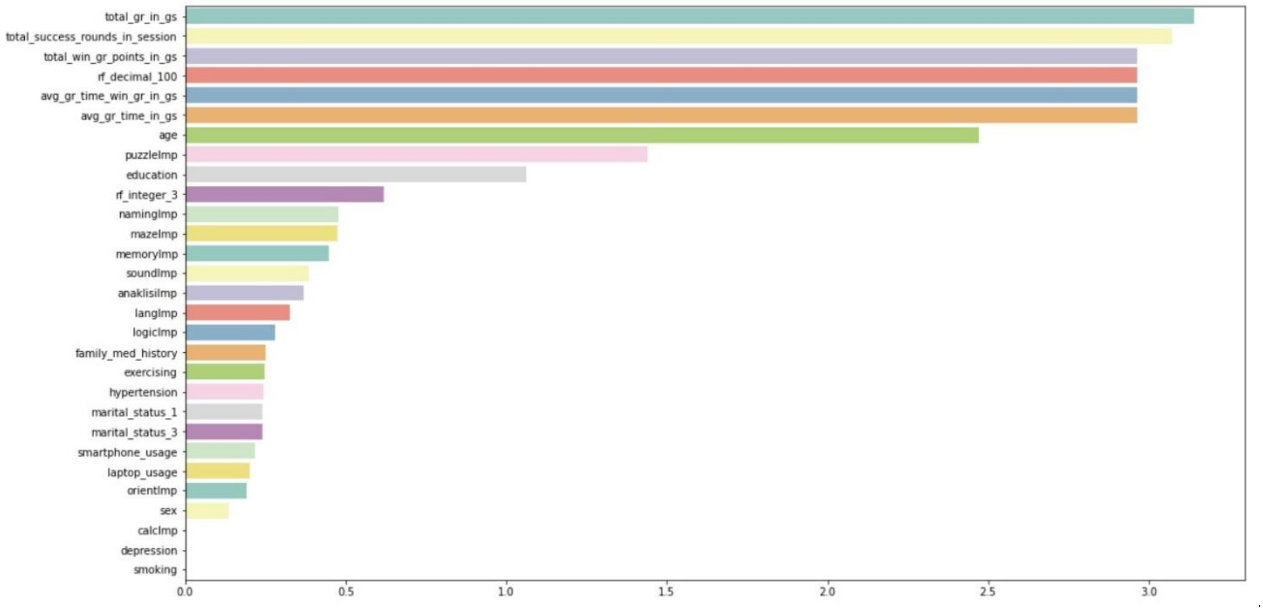
Με αυτό τον τρόπο υπολογίζουμε τη μέση τιμή του τετραγώνου της διαφοράς από τη μέση τιμή, όπου x είναι οι τιμές ενός Feature, και x.mean() η μέση τιμή της κατανομής του. Με άλλα λόγια, με αυτό το τρόπο υπολογίζουμε το τετράγωνο της τυπικής απόκλισης σ^2 και μπορούμε να το διαπιστώσουμε συγκρίνοντας με το αποτέλεσμα που προκύπτει χρησιμοποιώντας τη μέθοδο std της βιβλιοθήκης Pandas.

```
dataframe.std()['x']**2
```

Για τον υπολογισμό του Threshold κάνουμε τον συλλογισμό ότι θέλουμε να αποκλείσουμε τα Features των οποίων η συχνότερα εμφανιζόμενη τιμή εμφανίζεται σε πάνω από το 80% της κατανομής. Το Variance μιας τέτοιας κατανομής, με βάση το Documentation του Scikit-learn, θα πρέπει να υπολογίζεται

υποθέτοντας Binomial κατανομή. Οπότε υπολογίζουμε το Variance με βάση τον τύπο (p * (1 - p)), όπου p το ποσοστό των εμφανίσεων της πιο συχνά εμφανιζόμενης τιμής. Η τιμή του Variance που προκύπτει από τον παραπάνω τύπο για 80% ίδιες τιμές στη κατανομή ενός Feature, την οποία θα χρησιμοποιήσουμε ως Threshold, είναι 0.16 και έχει ως αποτέλεσμα τον αποκλεισμό των εξής Features ['sex', 'smoking', 'depression', 'calcImp'].

Η διαδικασία της αφαίρεσης των Features με χαμηλό Variance μπορεί να εντάσσεται στην ενότητα του Feature Selection, ωστόσο σε επίπεδο Python Script γίνεται αμέσως πριν από το Scaling. Αυτό διότι όταν επιλέξουμε τη μέθοδο του Normalization στο σύνηθες Range τιμών 0 έως 1 το Variance κυμαίνεται σε τιμές χαμηλότερες του Threshold για όλα τα Features. Ενώ όταν επιλέξουμε τη μέθοδο του Standardization, λόγω του Transformation που υφίστανται τα δεδομένα όλα τα Features έχουν όλα το ίδιο Variance.



Εικόνα 33 Κατάταξη των Features με βάση το Variance

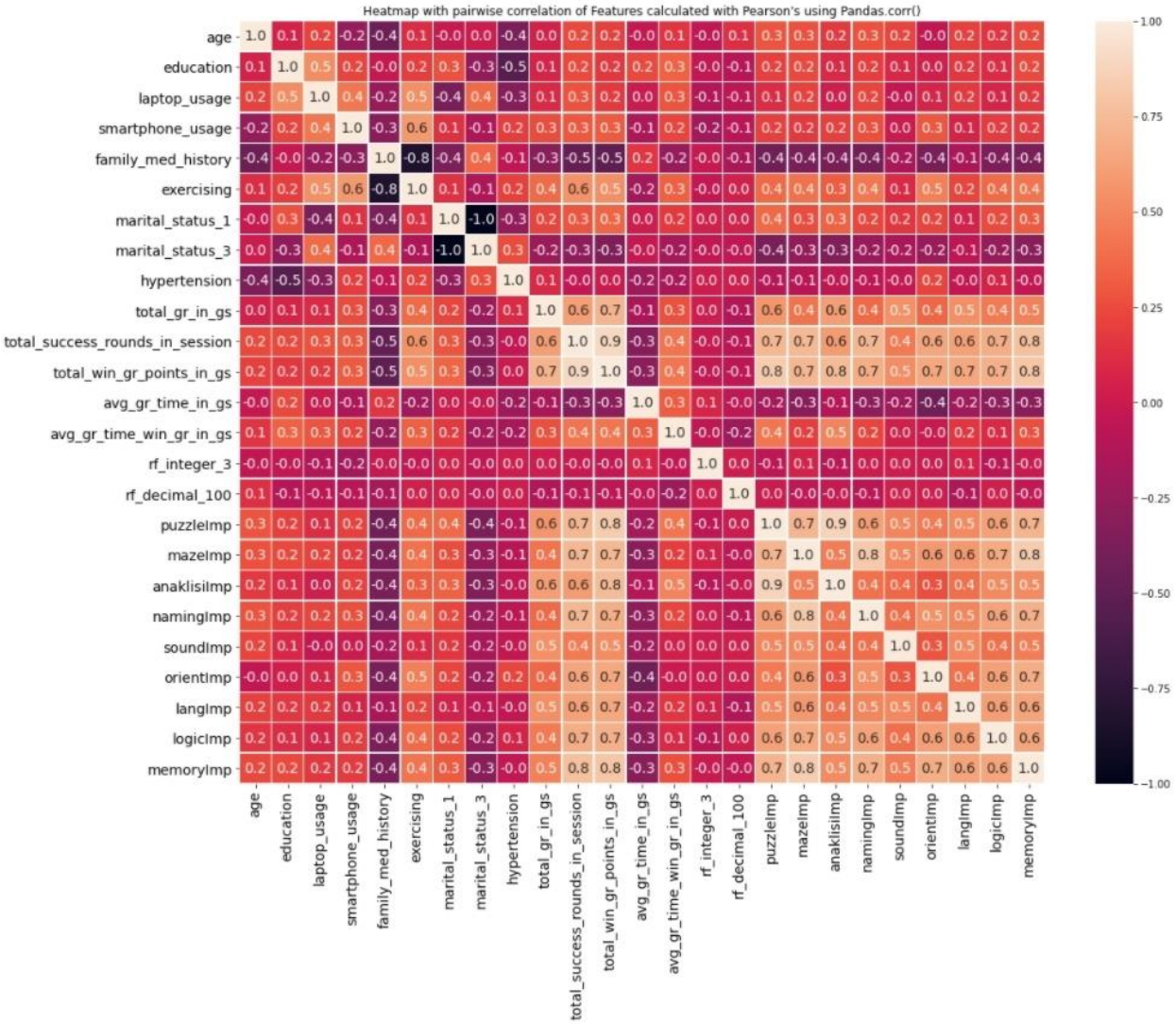
Features	Variance
total_gr_in_gs	3.143000
total_success_rounds_in_session	3.073779
total_win_gr_points_in_gs	2.963965
rf_decimal_100	2.963965
avg_gr_time_win_gr_in_gs	2.963965
avg_gr_time_in_gs	2.963965
age	2.472440
puzzleImp	1.438895
education	1.063239
rf_integer_3	0.618573
namingImp	0.477181
mazeImp	0.472789
memoryImp	0.447543
soundImp	0.384338
anaklisiImp	0.368072
langImp	0.326057
logicImp	0.278467
family_med_history	0.251674
exercising	0.246973
hypertension	0.244267
marital_status_1	0.240991
marital_status_3	0.240991
smartphone_usage	0.216066
laptop_usage	0.202108
orientImp	0.191396
sex	0.135308
calcImp	0.001607
depression	0.000000
smoking	0.000000

Πίνακας 5 Αναλυτικές τιμές Variance για όλα τα Features

4.2.3.3.3 Feature Correlation Inspection

Στο επόμενο βήμα της διερευνητικής διαδικασίας EDA, θέλουμε να εξερευνήσουμε το βαθμό συσχέτισης που έχουν τα Features μεταξύ τους.

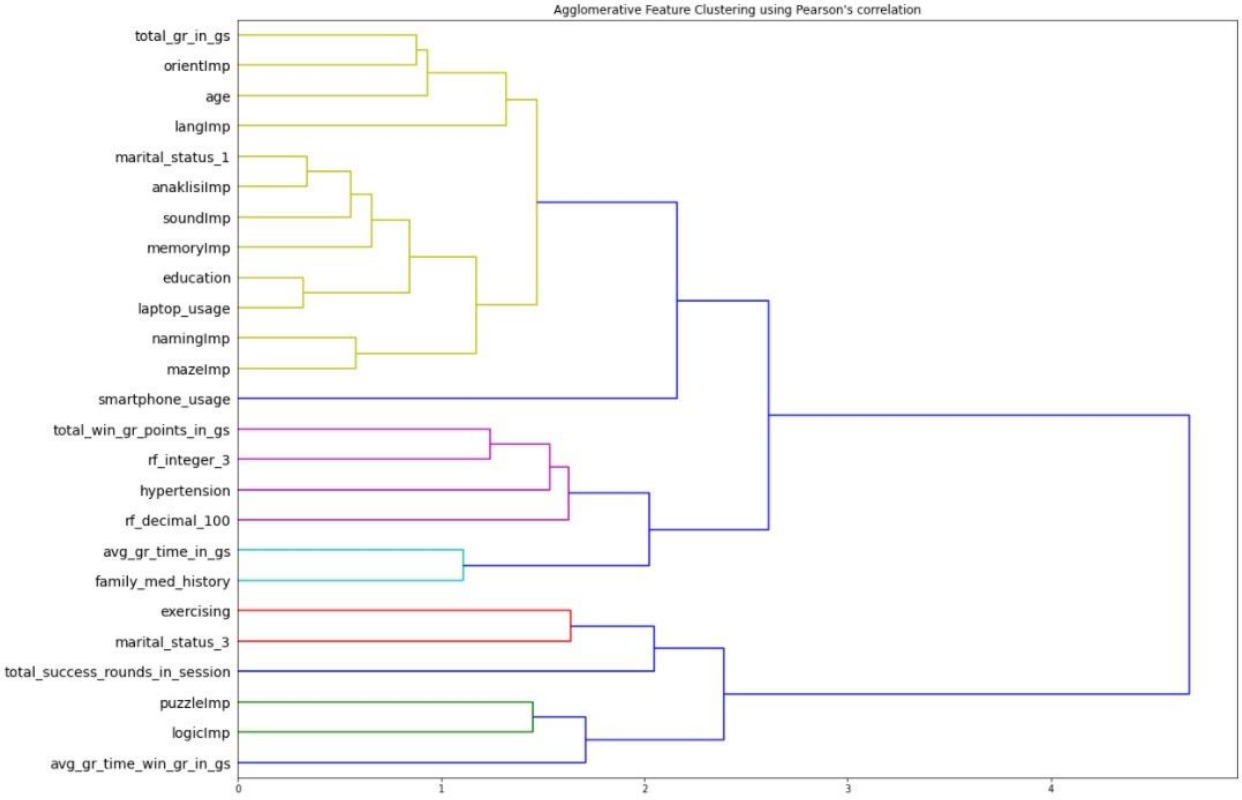
Για αυτό το σκοπό θα χρησιμοποιήσουμε τη τιμή του Correlation για κάθε ζεύγος από Feature. Ο υπολογισμός του Correlation γίνεται με χρήση της συνάρτηση corr() της βιβλιοθήκης Pandas, ενώ έχουμε την δυνατότητα να επιλέξουμε εάν ο υπολογισμός θα γίνει με τη προεπιλεγμένη μέθοδο Pearson's ή με κάποια από τις εναλλακτικές, όπως είναι η μέθοδοι Kendall's, Spearman's, ή κάποια δική μας μέθοδος. Στο διάγραμμα της παρακάτω εικόνας παρουσιάζεται η τιμή του Correlation αποτυπωμένη σε ένα Heatmap για κάθε ζευγάρι από Features.



Εικόνα 34 Heatmap της τιμής του Correlation μεταξύ όλων των Features, έχοντας υπολογιστεί με τη μέθοδο Pearson's Correlation

Στη συνέχεια, χρησιμοποιούμε τη συνάρτηση fcluster της βιβλιοθήκης SciPy ώστε να χωρίσουμε τα Features σε συστάδες (Clusters) με βάση το βαθμό συσχέτισης τους. Η συνάρτηση fcluster πραγματοποιεί Agglomerative Clustering, ουσιαστικά αθροίζει τις αποστάσεις των επιμέρους Features που συμπεριλαμβάνει σε ένα Cluster. Η μέγιστη απόσταση υπολογίζεται όταν όλα τα Features ομαδοποιηθούν σε ένα Cluster, οπότε δίνοντας ένα ποσοστό αυτής της τιμής επιλέγουμε την αντίστοιχη ομαδοποίηση εκείνου του επιπέδου. Για παράδειγμα δίνοντας 0 η συνάρτηση επιστρέφει ένα Cluster για κάθε Feature, ενώ δίνοντας τη μέγιστη αθροιστική απόσταση η συνάρτηση επιστρέφει ένα Cluster με όλα τα Features.

Το διάγραμμα στη παρακάτω εικόνα είναι τύπου Dendrogram και απεικονίζει τα Clusters που δημιουργούνται χρησιμοποιώντας ως όριο το 36% από τη μέγιστη απόσταση, η οποία σε αυτή την περίπτωση έχει τιμή περίπου 4.68. Η μέθοδος που χρησιμοποιείται για τον υπολογισμό, είναι η μέθοδος "linkage" του πακέτου "cluster.hierarchy" της βιβλιοθήκης "scipy" και ο τρόπος με τον οποίο γίνεται η ομαδοποίηση είναι με συσσωμάτωση (Agglomeration), στην αρχή των μεμονωμένων Features και στη συνέχεια των Clusters, βάση των μεταξύ τους αποστάσεων (Pairwise distances), οι οποίες υπολογίζονται με την μέθοδο "pdist", του πακέτου "spatial.distance" της βιβλιοθήκης "scipy". Ως είσοδος στην μέθοδο "pdist", έχουμε τον δυσδιάστατο πίνακα με τα Correlations των Features όπως ακριβώς υπολογίστηκαν για να εκτυπωθούν και στο Heatmap που είδαμε παραπάνω. Το όριο, 36%, επιλέχθηκε αυθαίρετα μετά από εμπειρική παρατήρηση αναζητώντας μια τιμή που μεγιστοποιεί τον αριθμών των Clusters.



Εικόνα 35 Ιεραρχικό Δενδρόγραμμα Συστάδων (Hierarchical Dendrogram Clusters)

Τέλος, για κάθε Cluster συγκεντρώνουμε τα Features σε μια λίστα την οποία κρατάμε σε ένα Dictionary όπου κλειδί είναι το Index του Cluster, με σκοπό να χρησιμοποιήσουμε τα Features με αυτή την ομαδοποίηση στην αμέσως επόμενη ενότητα.

Το κέρδος αυτής της διαδικασίας είναι ότι γνωρίζουμε ποια Features έχουν μεγάλο βαθμό Correlation μεταξύ τους, οπότε επιλέγοντας ένα Feature από ένα Cluster αυτομάτως γνωρίζουμε ποια θα ήταν καλό να αφήσουμε εκτός της διαδικασίας της εκπαίδευσης.

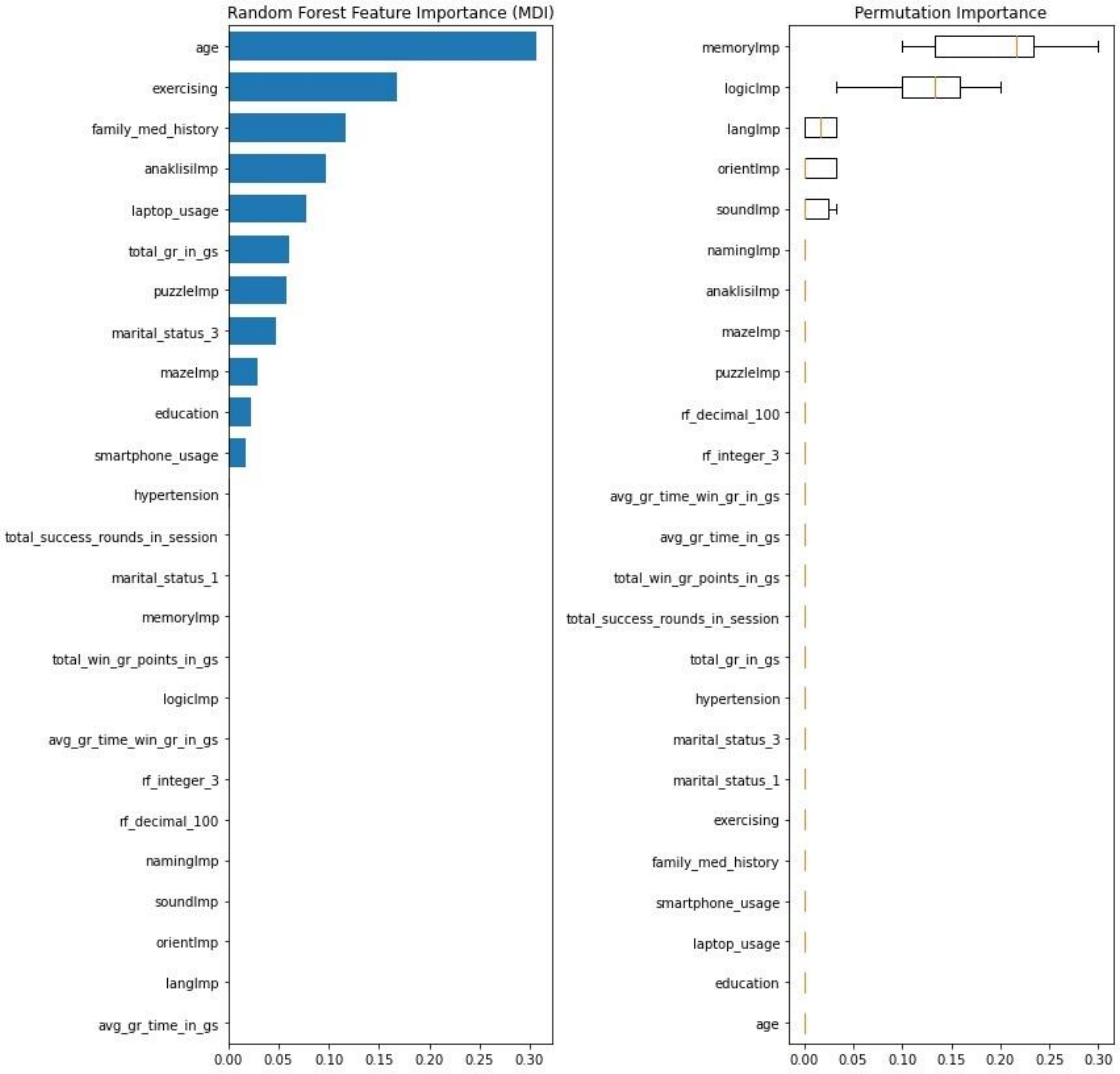
4.2.3.3.4 Feature Importance Inspection

Σε αυτό το σημείο θέλουμε να εξετάσουμε τη σημαντικότητα του κάθε Feature ως προς το Target Class. Για να μπορέσουμε να εξάγουμε ένα τέτοιο αποτέλεσμα θα χρειαστεί να εκπαιδεύσουμε ένα μοντέλο για αυτό και μόνο το σκοπό, για την εκπαίδευση του οποίου χρησιμοποιήθηκε ο αλγόριθμος Random Forest Classifier. Οι πληροφορίες που μπορούν να μας βοηθήσουν στην εύρεση των σημαντικότερων Features χρησιμοποιώντας ένα μοντέλο Classifier είναι το Mean Decrease in Impurity (MDI) γνωστό επίσης και ως Gini Importance και το Permutation Importance γνωστό επίσης ως Mean Decrease in Accuracy (MDA).

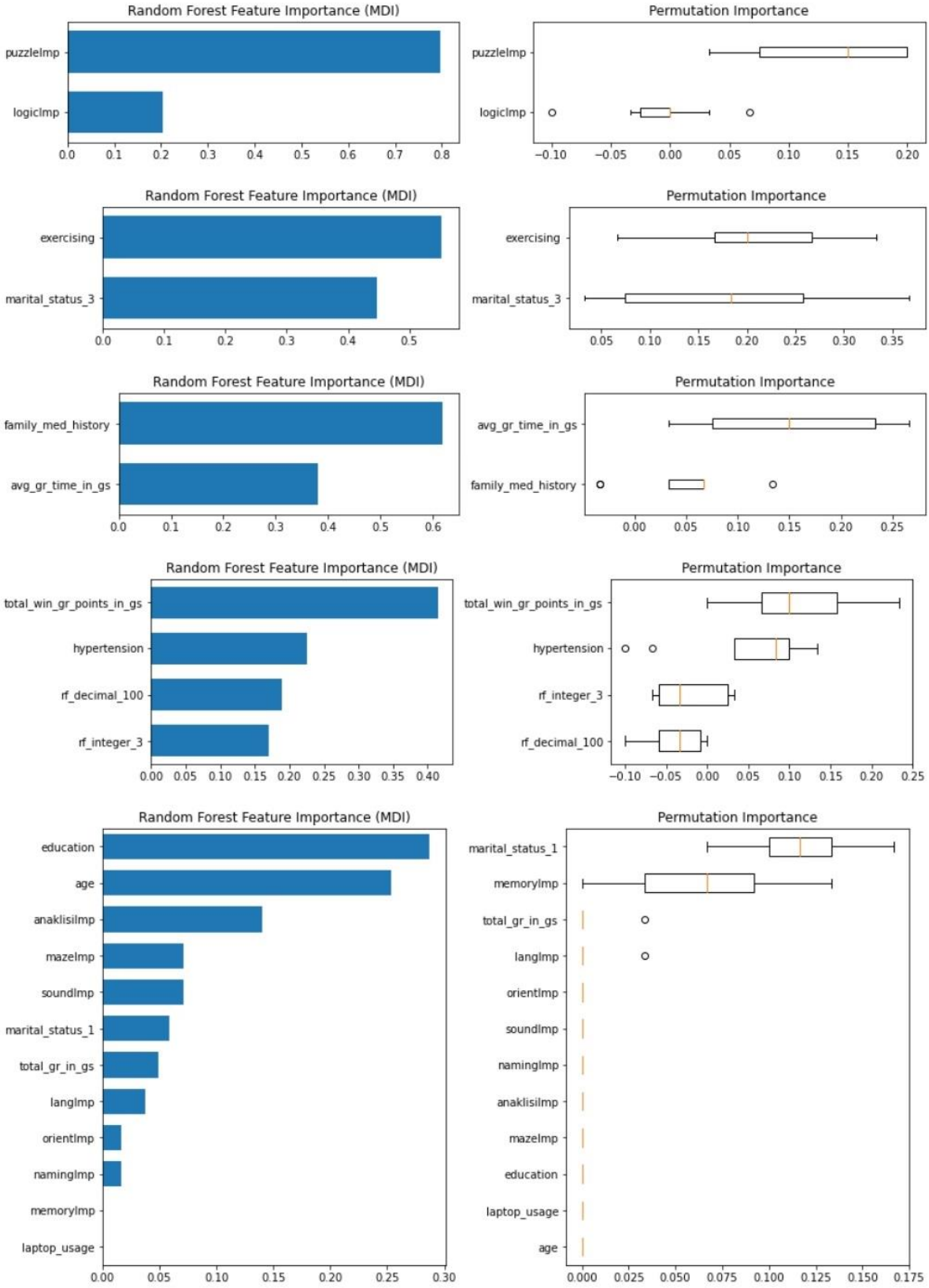
Όσον αφορά το MDI, ως πληροφορία βρίσκεται στο Property feature_importances_ του μοντέλου που εξάγουμε. Υπολογίζεται ως, ο μέσος όρος του συνόλου των Nodes στα οποία έχει χρησιμοποιηθεί το Feature ως κριτήριο για το επόμενο Split, ως προς τον αριθμό των εγγραφών που διαχωρίζουν τα συγκεκριμένα Nodes, για όλα τα Decision Trees του Random Forest Classifier.

Όσον αφορά το MDA, αυτό δεν υπάρχει ως πληροφορία στο μοντέλο που εξάγουμε αλλά καλούμε τη συνάρτηση permutation_importances του πακέτου Scikit-learn για να το υπολογίσουμε. Ουσιαστικά πρόκειται για έναν αλγόριθμο ο οποίος σε κάθε επανάληψη αντικαθιστά τις τιμές ενός Feature με τυχαίες τιμές και υπολογίζει ξανά το Accuracy του μοντέλου. Όταν το Target Class εξαρτάται σε μεγάλο βαθμό από ένα Feature τότε αυτό θα παρουσιάσει υψηλές τιμές MDA καθώς η αλλαγή των τιμών του με τυχαίες θα είχε μεγάλο αντίκτυπο στο Accuracy του μοντέλου.

Στα παρακάτω γραφήματα βλέπουμε στα αριστερά τη κατάταξη ως προς MDI και στα δεξιά τη κατάταξη ως προς MDA, αρχικά για όλα τα Features συγκεντρωτικά και στη συνέχεια για το εκάστοτε Cluster από Features όπως αυτά δημιουργήθηκαν κατά το προηγούμενο βήμα.



Εικόνα 36 Τιμές των MDI και MDA Metrics όλων των Features



Εικόνα 37 Τιμές MDI και MDA Metrics για κάθε συστάδα από Features του προηγούμενου βήματος

Ωστόσο στη βιβλιογραφία υπάρχουν αναφορές οι οποίες υποστηρίζουν ότι το Permutation Importance κάποιες φορές είναι πιθανό να παρέχει Biased αποτελέσματα οπότε αντί αυτής της μεθόδου προτείνονται εναλλακτικοί τρόποι διαλογής των Features [29].

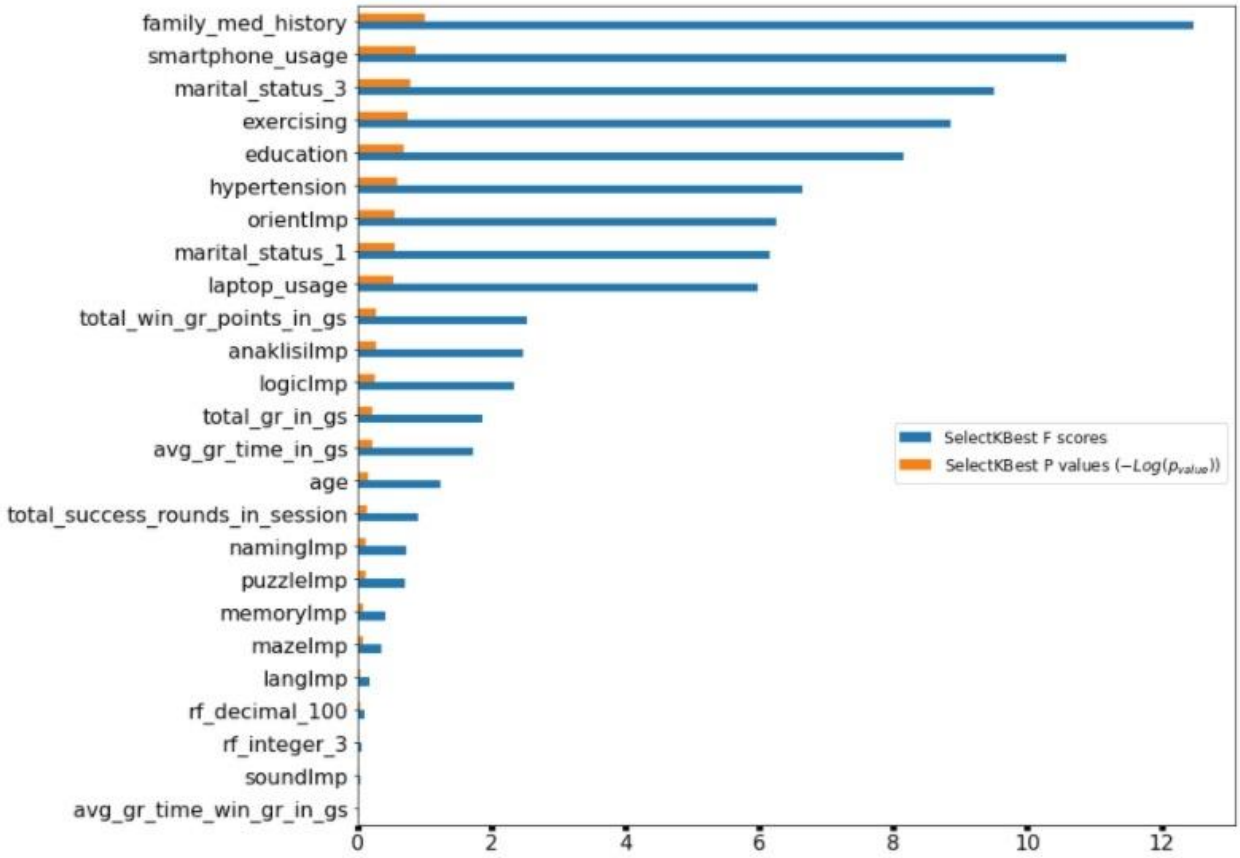
Με βάση τις πληροφορίες από τα παραπάνω γραφήματα και ακολουθώντας τη παρακάτω μεθοδολογία για τη διαλογή, καταλήγουμε σε 12 Features όπως αυτά παρουσιάζονται στη στήλη Custom Feature Selection στον πίνακα Feature Selection Scenarios.

- Πρώτον, αφαιρούμε όλα τα Features που έχουν χαμηλότερο MDI από αυτά των Random Features.
- Δεύτερον, από τα εναπομείναντα και μόνο για όσα έχουν καλές τιμές MDI ή MDA επιλέγουμε αυτά τα οποία ο αλγόριθμος δεν έχει συμπεριλάβει σε κάποιο Cluster, καθώς αυτά έχουν το πλεονέκτημα ότι πληρούν το κριτήριο να μην είναι Correlated με άλλα Features.
- Για τα εναπομείναντα, εξετάζοντας πρώτα τα Clusters τα οποία περιλαμβάνουν λίγα Features, επιλέγουμε το σημαντικότερο με βάση τις τιμές MDI, MDA και αποκλείουμε τα υπόλοιπα.
- Για τα εναπομείναντα, τα οποία βρίσκονται σε μεγαλύτερα Clusters και άρα έχουν κάποιο βαθμό Correlation, θα μπορούσαμε να προχωρήσουμε είτε σε επιπλέον Clustering μεταξύ τους, είτε να συμβουλευτούμε το Pairwise Correlation Heatmap ώστε επιλέγοντας ένα να ψάχνουμε για το λιγότερο Correlated εντός του Cluster.

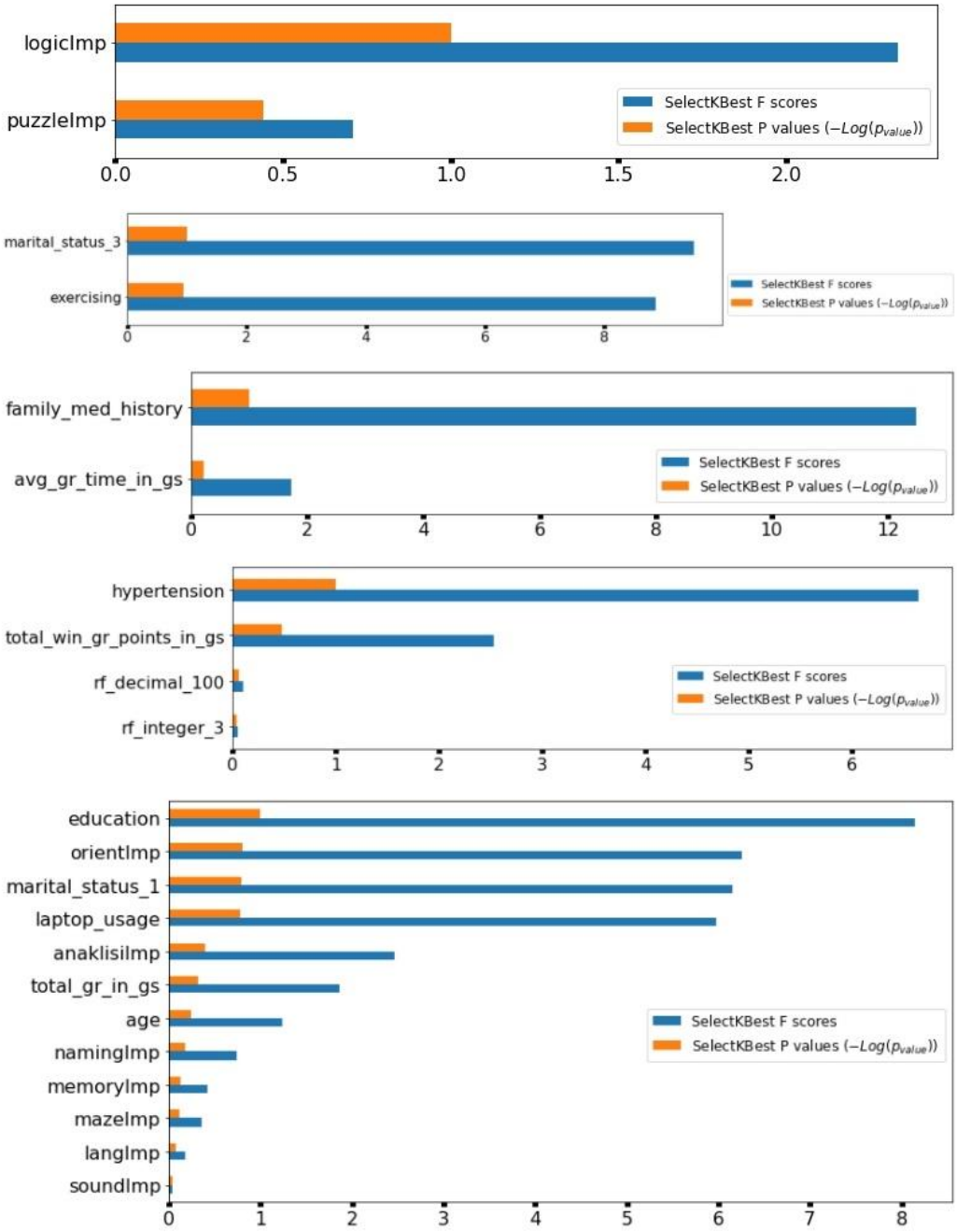
Επιπλέον έγινε χρήση της μέθοδο SelectKBest της βιβλιοθήκης Scikit-learn για να εξάγουμε αυτοματοποιημένα μια λίστα από τα ιδανικά Features με βάση τη συνάρτηση βαθμολόγησης (Scoring Function) chi2.

Η μέθοδος SelectKBest είναι μια από τις Wrapper μεθόδους που παρέχει το Scikit-learn για αυτοματοποιημένη επιλογή Features. Η στρατηγική με την οποία λειτουργεί η SelectKBest ονομάζεται Univariate Feature Selection και ουσιαστικά αυτό που κάνει είναι να δημιουργεί ένα μοντέλο και βαθμολογεί το εκάστοτε Feature αναλόγως τη Scoring Function που έχουμε επιλέξει.

Το αποτέλεσμα αυτής της διαδικασίας είναι η εξαγωγή δύο Metrics, του F-Score και του P-Value για το κάθε Feature, ενώ επίσης υπάρχει και η αυτόματη εξαγωγή των k ιδανικών Features βάση αυτών των δύο Metrics. Στα παρακάτω διαγράμματα παρατηρούμε τα F-Scores και P-Values για όλα τα Features και στη συνέχεια για κάθε Cluster από Features. Συγκεκριμένα για τα P-Values, όπως προτείνεται και από το Documentation του Scikit-learn, πριν την αποτύπωση τους στο διάγραμμα, υπολογίζουμε την αρνητική τιμή του λογάριθμου του με βάση το 10. Ουσιαστικά θέλουμε την αρνητική τιμή διότι στη πραγματικότητα όσο μικρότερη είναι η τιμή P-Value τόσο πιο σημαντικό είναι ένα Feature. Ενώ υπολογίζοντας τον λογάριθμό με βάση το 10 φέρνουμε τη τιμή σε μεγαλύτερη κλίμακα ώστε να έχει νόημα να τυπωθεί στο ίδιο διάγραμμα με τις τιμές του F-Score.



Εικόνα 38 Απεικόνιση διαγράμματος των τιμών F-Score και P-Value του συνόλου των Features. Ταξινόμηση βάση του F-Score



Εικόνα 39 Διαγράμματα τιμών F-Score και P-Value για κάθε Feature Cluster. Ταξινόμηση του εκάστοτε διαγράμματος με βάση τις τιμές F-Score

Σε αυτό το σημείο παρατηρούμε πως για αρκετά από τα Features που είχαν υψηλές τιμές MDI και MDA έχουν υψηλές τιμές και όσον αφορά τα F-Score και P-Value Metrics.

Ακολουθεί συγκεντρωτικός πίνακας των Features που έχουν επιλεγεί για το επόμενο βήμα, για κάθε έναν από τους τρόπους επιλογής τους.

Custom Feature Selection	**SelectKBest & Chi2**
Age	Education
Family Medical History	Laptop Usage
Exercising	Smartphone Usage
Education	Family Medical History
Average Game Round Time in Game Session	Exercising
Orientation Importance	Marital Status 1 (Έγγαμος)
Naming Importance	Marital Status 3 (Χήρος)
Memory Importance	Hypertension
Anaklisi Importance	Total Game Round Points in Win Game Rounds in a Game Session
	Anaklisi Importance
	Logic Importance
	Memory Importance

Πίνακας 6 Set από Features ανάλογα με τον τρόπο που έχουν επιλεγεί

4.2.3.4 Classifier Selection

Σε αυτό το σημείο, έχουμε ολοκληρώσει τις διαδικασίες Preprocessing και Feature Selection και είμαστε έτοιμοι να περάσουμε στο επόμενο σημαντικό βήμα της EDA, την εύρεση του κατάλληλου αλγορίθμου για την εκπαίδευση του μοντέλου. Τα κριτήρια που θέτουμε στη προκειμένη περίπτωση για να κρίνουμε έναν αλγόριθμο ως κατάλληλο, είναι τα εξής.

- Πρώτον, το μοντέλο που εξάγεται, θα πρέπει να μην δίνει καμία ένδειξη ότι είναι Overfitted ή Underfitted, για τα επιλεγμένα Features και για τις επιλεγμένες παραμέτρους του αλγορίθμου, βάση της θεωρίας που περιγράφεται στην αντίστοιχη ενότητα στο κεφάλαιο της ορολογίας.
- Δεύτερον και εξίσου σημαντικό, θα πρέπει να έχει καλή απόδοση όσον αφορά το Metric Specificity. Διότι, όπως αναφέρεται και στην αντίστοιχη ενότητα των Metrics στο κεφάλαιο της ορολογίας, μια χαμηλή τιμή στο Specificity, θα σήμαινε υψηλές πιθανότητες κάποιοι χρήστες να κατηγοριοποιηθούν ως False Positive, όπου στη προκειμένη περίπτωση αντιστοιχεί στο να έχουμε κατατάξει έναν χρήστη που είναι NC ως MCI-AD.

Η συγκεκριμένη διαδικασία έχει ως είσοδο το Dataset και τη λίστα των επιλεγμένων Features. Ως έξοδο για τη διαδικασία Classifier Selection δεν έχουμε κάποιο αντικείμενο, αλλά την εκτύπωση μιας σειράς από Metrics, που αποτυπώνουν την ακρίβεια του κάθε μοντέλου για κάθε Testing Dataset, τα οποία θα μας βοηθήσουν να επιλέξουμε τον κατάλληλο αλγόριθμο.

Τα μοντέλα εκπαιδεύονται με τη διαδικασία που έχει δημιουργηθεί στη Wrapper μέθοδο train_models, της κλάσης TrainingMethods στο αρχείο moduleModelTraining.py, η οποία δέχεται ως παράμετρο ένα Splitted Dataset, ένα Python Dictionary με κλειδί το όνομα του αλγορίθμου και τιμή το εκπαιδευμένο μοντέλο. Πιο συγκεκριμένα οι αλγόριθμοι που έχουν επιλεγεί για την εκπαίδευση των μοντέλων είναι οι εξής.

- Logistic Regression
- Decision Tree
- Random Forest
- Support Vector Classifier

- K Neighbors Classifier
- Gaussian Naive Bayes. Ο αλγόριθμος Gaussian Naive Bayes, είναι ο μόνος Classifier τύπου Probabilistic, από αυτούς που χρησιμοποιήθηκαν.
- Multi-layer Perceptron. Ο αλγόριθμος Multi-layer Perceptron, είναι ο μόνος Classifier από αυτούς που χρησιμοποιήθηκαν, ο οποίος βασίζεται σε νευρωνικά δίκτυα.
- Custom Ensemble. Ο αλγόριθμος Custom Ensemble είναι ο μόνος Classifier τύπου Ensemble από αυτούς που χρησιμοποιήθηκαν και οι επιμέρους αλγόριθμοι που τον απαρτίζουν είναι οι Logistic Regression, Decision Tree, GaussianNB, KNeighborsClassifier και Random Forest. Ενώ για την υλοποίηση του χρησιμοποιήθηκε η μέθοδος VotingClassifier της βιβλιοθήκης Scikit-learn.

Σε αυτό το στάδιο, τα μοντέλα που θα εκπαιδεύσουμε θα τα θεωρήσουμε ως σημεία αναφοράς (Baseline Models), για τις μετέπειτα διαδικασίες βελτιστοποίησης (Optimization) που θα εκτελέσουμε.

Επίσης, μη έχοντας πραγματοποιήσει ακόμη καμία διαδικασία Optimization, είναι φυσιολογικό και αναμενόμενο να παρατηρήσουμε ακραίες τιμές στα Metrics των Baseline Models, κάτι που σημαίνει πως εάν σταματούσαμε εδώ τη διαδικασία θα είχαμε καταλήξει με κάποιο Overfitted ή Underfitted μοντέλο.

Οι επιλεγμένες τιμές για τις παραμέτρους των Baseline μοντέλων είναι οι προεπιλεγμένες με βάση το Documentation του Scikit-learn.

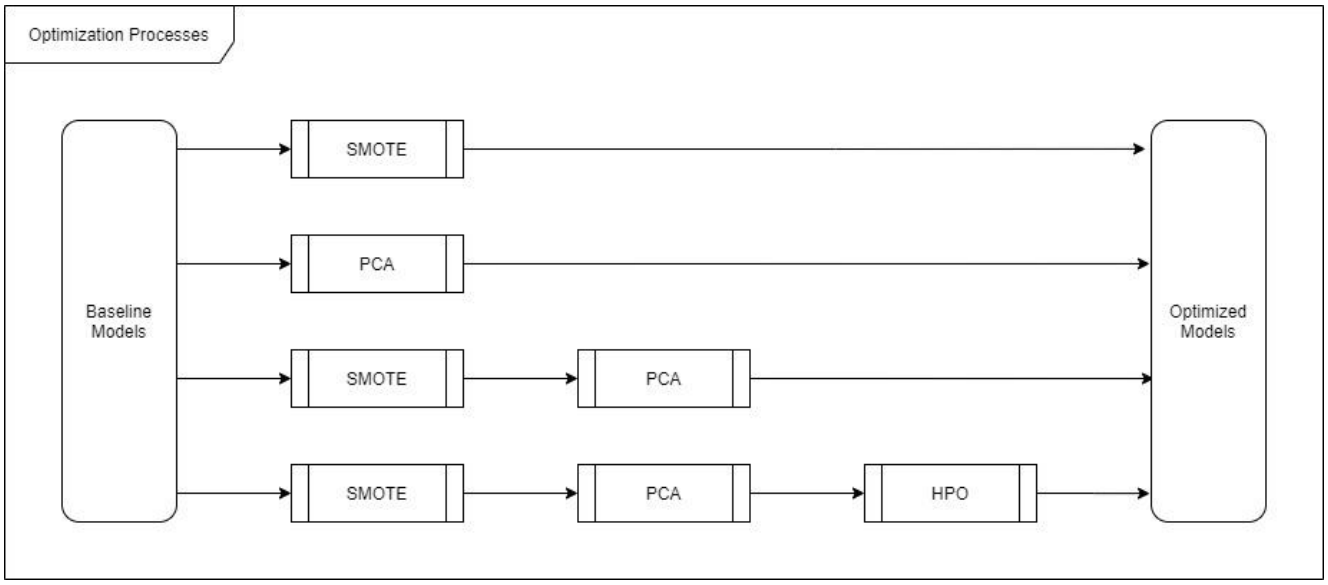
Κρίνοντας από τους συγκεντρωτικούς πίνακες των Metrics όπως παρουσιάζονται στο κεφάλαιο της αξιολόγησης, παρατηρούμε ότι για τα Features που επιλέχθηκαν Manually, τα Metrics για 5 από τα 8 Baseline μοντέλα παρουσιάζουν μεγάλο Variance ενώ ένα παρουσιάζει υψηλό Bias. Αντιθέτως για τα Features που επιλέχθηκαν μέσω SelectKBest, παρατηρούμε ότι όλα πλην ενός, παρουσιάζουν υψηλές τιμές Accuracy χωρίς να είναι Overfitted.

4.2.3.5 Optimization

Με τον όρο Optimization αναφερόμαστε σε όλες τις διαδικασίες που μπορούμε να εκτελέσουμε για να βελτιώσουμε την απόδοση των πειραματικών μοντέλων πριν προχωρήσουμε στη δημιουργία ενός παραγωγικού μοντέλου.

Μερικές από αυτές τις διαδικασίες μπορεί να περιλαμβάνουν είτε την περαιτέρω επεξεργασία των δεδομένων μέσω μεθόδων εμπλουτισμού των δεδομένων (Data Augmentation), το Feature Extraction που περιλαμβάνει τη προβολή υπαρχόντων Features σε λιγότερες διαστάσεις από το πλήθος τους (Dimensionality Reduction), την εύρεση των ιδανικών παραμέτρων για τους αλγόριθμους που χρησιμοποιούμε για να εκπαιδεύσουμε το μοντέλο (Hyper-Parameter Optimization, HPO), είτε τον συνδυασμό των παραπάνω μεθόδων.

Στο πλαίσιο τη διπλωματικής, ακολουθήθηκαν 4 τρόποι για τη βελτίωση των Metrics των Baseline μοντέλων. Αυτοί όπως περιγράφονται και στο διάγραμμα της παρακάτω εικόνας είναι οι εξής, SMOTE, SMOTE-PCA, PCA, SMOTE-PCA-HPO.



Εικόνα 40 Συνδυασμοί μεθόδων για το Optimization των Baseline μοντέλων

4.2.3.5.1 Data Augmentation

Αρχικά, όσον αφορά τη διαδικασία βελτιστοποίησης του Dataset, θα μπορούσαμε να πούμε ότι το μεγαλύτερο μέρος αυτής γίνεται κατά τη διαδικασία του Preprocessing, όπως είδαμε για παράδειγμα στη διαχείριση των Outliers και στο Scaling των τιμών. Ωστόσο, αυτές οι διαδικασίες ήταν μέρος της απολύτως απαραίτητης διαχείρισης του Dataset ώστε να προκύψει ένα βασικό μοντέλο το οποίο μπορούμε να χαρακτηρίσουμε ως μέτρο σύγκρισης (Baseline Model). Από εκεί και έπειτα, υπάρχουν επιπλέον τρόποι για να βελτιώσουμε το Dataset και κατ' επέκταση την απόδοση των μοντέλων. Οι πλέον γνωστοί τρόποι, όταν έχουμε να κάνουμε με Dataset όπου υπάρχει ανισοκατανομή μεταξύ των κατηγοριών του Target Class, είναι το Oversampling και το Undersampling.

Με το Oversampling δημιουργούμε ουσιαστικά επιπλέον πλασματικές εγγραφές ώστε οι εγγραφές των Labels που αποτελούν μειοψηφία στο Target Class να αριθμούν το ίδιο με τα Labels που είναι πλειοψηφία. Αντίστοιχα η τεχνική του Undersampling, προσπαθεί να ισορροπήσει σε πλήθος τα Labels του Target Class, εφαρμόζοντας όμως τυχαία επιλογή λιγότερων συνολικά εγγραφών από τις ήδη υπάρχουσες εγγραφές. Ο λόγοι για τους οποίους θέλουμε να εφαρμόσουμε Oversampling ή Undersampling στα δεδομένα μας είναι δύο.

Πρώτον διότι με αυτό το τρόπο διασφαλίζουμε ότι τα μοντέλα που θα εκπαιδεύσουμε δεν θα είναι Biased προς τη το πλειοψηφικό Label κατά τη ταξινόμηση νέων δεδομένων.

Δεύτερον, διότι κατά την εκπαίδευση των μοντέλων και πιο συγκεκριμένα κατά την εφαρμογή του Cross Validation, ο μέγιστος αριθμός κομματιών στα οποία μπορεί να χωριστεί το Training Dataset είναι τόσα ακριβώς όσο και το πλήθος του Label που αποτελεί μειοψηφία. Δεδομένου του ότι σε κάτι κομμάτι είναι απαραίτητο να υπάρχει τουλάχιστον ένα δείγμα από κάθε Label του Target Class.

Τέλος, όσον αφορά το Oversampling, οι περισσότερες βιβλιογραφικές αναφορές ορίζουν πως η εφαρμογή της μεθόδου θα πρέπει να γίνεται μετά το πέρας του Feature Selection, έτσι ώστε τα πλασματικά δεδομένα που θα εισαχθούν το Dataset να μην επηρεάσουν την επιλογή των κατάλληλων Features [28].

Στο πλαίσιο της διπλωματικής για τη βελτιστοποίηση του Dataset πραγματοποιήθηκε Oversampling, αυτό διότι η κατανομή μεταξύ των δύο Labels του Target Class είναι αρκετά άνιση. Πιο συγκεκριμένα, αντιστοιχούν 94 Sessions σε χρήστες όπου στη πρώτη δοκιμασία MOCA κατέγραψαν βαθμολογία η οποία τους κατατάσσει στις κατηγορίες AD-MCI, ενώ μόλις 25 Sessions αντιστοιχούν σε χρήστες όπου για την ίδια δοκιμασία κρίνονται ως NC.

Το Dataset, έχοντας 26 Features θα μπορούσαμε να το χαρακτηρίσουμε ως Low-Dimensional, κατ' επέκταση με βάση τη βιβλιογραφία [11], ίσως θα ήταν προτιμότερο να εφαρμόσουμε κάποια από τις μεθόδους Undersampling. Ωστόσο, επειδή εξετάζουμε τα δεδομένα μας με βάση τα Sessions, έχουμε περιορισμένο αριθμό εγγραφών για την εκπαίδευση των μοντέλων, συνεπώς για να μην αφαιρέσουμε από τις ήδη λίγες εγγραφές, επιλέχθηκε η μέθοδος Oversampling. Η υλοποίηση έγινε με τη χρήση της μεθόδου SMOTE του πακέτου της βιβλιοθήκης "imblearn ".

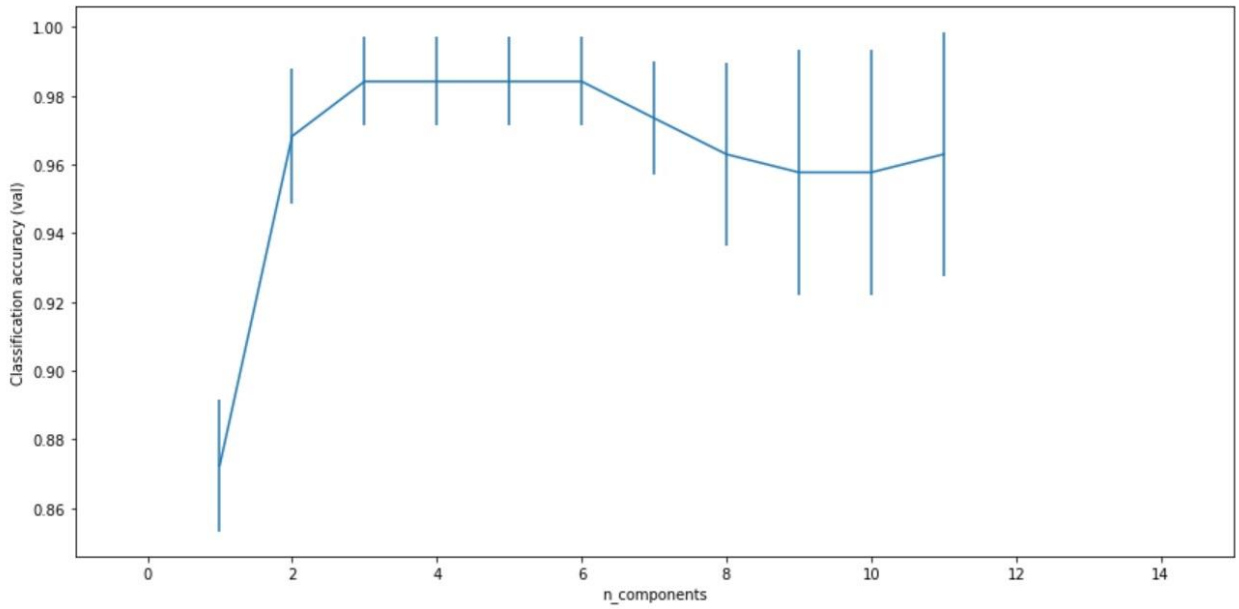
4.2.3.5.2 Dimensionality Reduction

Για την υλοποίηση της διπλωματικής και συγκεκριμένα για τη διαδικασία του Optimization χρησιμοποιήθηκε η τεχνική Principal Component Analysis (PCA). Όπως περιγράφεται και από το διάγραμμα στην αρχή της ενότητας του Optimization, έγιναν τόσο δοκιμές με το Dataset όπως αυτό εξάγεται από τη διαδικασία του Preprocessing, όσο και με το Dataset όπως αυτό προκύπτει από την εφαρμογή της μεθόδου SMOTE.

Η διαδικασία για την εφαρμογή PCA, δέχεται ως είσοδο ένα Dataset με τις μεταβλητές όπως αυτές είναι είτε στο τέλος του Preprocessing, είτε στο τέλος της εφαρμογής της SMOTE μεθόδου. Ως έξοδος, παράγεται ένα νέο Dataset το οποίο αποθηκεύεται ως Dataframe, στο Repository της εφαρμογής για να χρησιμοποιηθεί παρακάτω. Ενώ τα βήματα αναλυτικά για τη διαδικασία PCA είναι τα εξής.

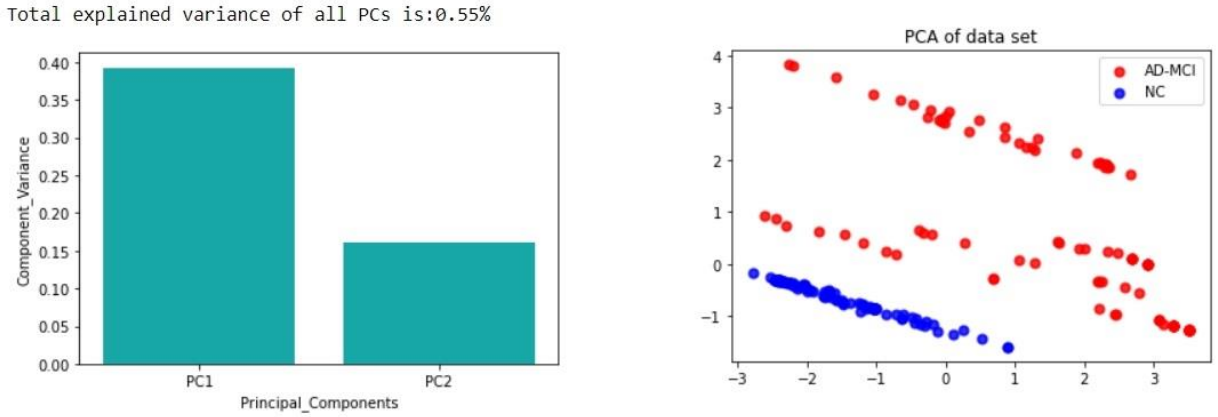
Πρώτον, θα πρέπει να βρούμε ποιος είναι ο ιδανικός αριθμός από Principal Components, οπότε θα χρησιμοποιήσουμε τη κλάση GridSearchCV της βιβλιοθήκης Scikit-learn. Αυτή η διαδικασία θα πραγματοποιηθεί για μια σειρά από διαφορετικές τιμές όσον αφορά το πλήθος των Principal Components PCs του PCA Transformer.

Η έξοδος από αυτή τη διαδικασία, όπως φαίνεται και στο παρακάτω διάγραμμα, αποτελείται από το πλήθος των PCs για τον άξονα x και από την απόδοση στο Testing Dataset με βάση τον αλγόριθμο που επιλέξαμε. Ο αλγόριθμος στη προκειμένη περίπτωση είναι ο Gaussian Naive Bayes και το πλήθος των PCs που εξετάζουμε είναι από 1 έως το πλήθος των Features μειωμένο κατά 1. Παράλληλα η μέθοδος fit της κλάσης GridSearchCV προσφέρει τη δυνατότητα για Cross Validation, το οποίο θα αφήσουμε στη Default ρύθμιση για 5-fold, κάτι το οποίο φαίνεται και από το Standard Deviation που αποτυπώνεται με τις κάθετες γραμμές στο διάγραμμα, για κάθε μια από τις εκπαιδεύσεις του μοντέλου.



Εικόνα 41 Αποτελέσματα Accuracy της μεθόδου GridSearchCV, ως προς τον αριθμό PC

Στη συνέχεια, αφού έχουμε αποφασίσει για τον ιδανικό αριθμό από PC, για να έχουμε μια καλύτερη εικόνα, μπορούμε να εκτυπώσουμε το διάγραμμα το οποίο απεικονίζει το Variance που αντιστοιχεί στο εκάστοτε PC καθώς και τις τιμές των δύο PC στους άξονες x και y με τα σημεία να απεικονίζουν τις αντίστοιχες κατηγορίες του Target Class, όπως φαίνεται στη παρακάτω εικόνα.



Εικόνα 42 Διάγραμμα στα αριστερά: Συνολικό Variance ανά PC. Διάγραμμα στα δεξιά: Εκτύπωση των σημείων των PC1, PC2 στους άξονες x, y αντίστοιχα, ως προς το Target Class

4.2.3.5.3 Hyperparameter Optimization

Οι περισσότεροι από τους αλγόριθμους μηχανικής μάθησης δέχονται μια σειρά παραμέτρων οι οποίες καθορίζουν το τρόπο λειτουργίας τους. Με τον όρο Hyperparameter Optimization (HPO), περιγράφουμε τη διαδικασία αναζήτησης εκείνων των παραμέτρων οι οποίες βελτιστοποιούν την απόδοση ενός μοντέλου που έχει εκπαιδευτεί από κάποιο αλγόριθμο μηχανικής μάθησης.

Όταν αναφερόμαστε στη βελτίωση της απόδοση του μοντέλου, αυτό που ουσιαστικά προσπαθούμε να κάνουμε είναι να βελτιώσουμε τον λόγο (Ratio) μεταξύ Bias και Variance του μοντέλου. Με άλλα λόγια

να διασφαλίσουμε ότι το μοντέλο που θα εκπαιδεύσει ο αλγόριθμος δεν θα είναι ούτε Overfitted, ούτε Underfitted για το τρέχον Dataset.

Επίσης, πέρα από τη βελτίωση της απόδοσης, υπάρχουν επιπλέον λόγοι για τους οποίους θα θέλαμε να τροποποιήσουμε τις παραμέτρους ενός αλγόριθμου, όπως είναι για παράδειγμα η μείωση του χρόνου εκπαίδευσης.

Βιβλιοθήκες όπως η Scikit-learn και η Yellowbrick μας παρέχουν μεθόδους ειδικά σχεδιασμένες για διαδικασίες τύπου HPO.

Η πλέον γνωστή μέθοδος για HPO, είναι η GridSearchCV της βιβλιοθήκης Sciki-learn, το χαρακτηριστικό της οποίας είναι ότι έχει τη δυνατότητα να ψάχνει τις βέλτιστες τιμές για πολλαπλές παραμέτρους ταυτόχρονα. Το αποτέλεσμα αυτής της Exhaustive search διαδικασίας, όπως αναφέρεται στο Documentation, είναι η επιστροφή ενός Set από τις βέλτιστες τιμές για κάθε παράμετρο. Ο τρόπος με τον οποίο, η GridSearchCV, αξιολογεί την επιρροή που έχει μια συγκεκριμένη τιμή μιας παραμέτρου είναι βάση των Metrics του μοντέλου που εκπαιδεύει κατά τη διάρκεια της διαδικασίας.

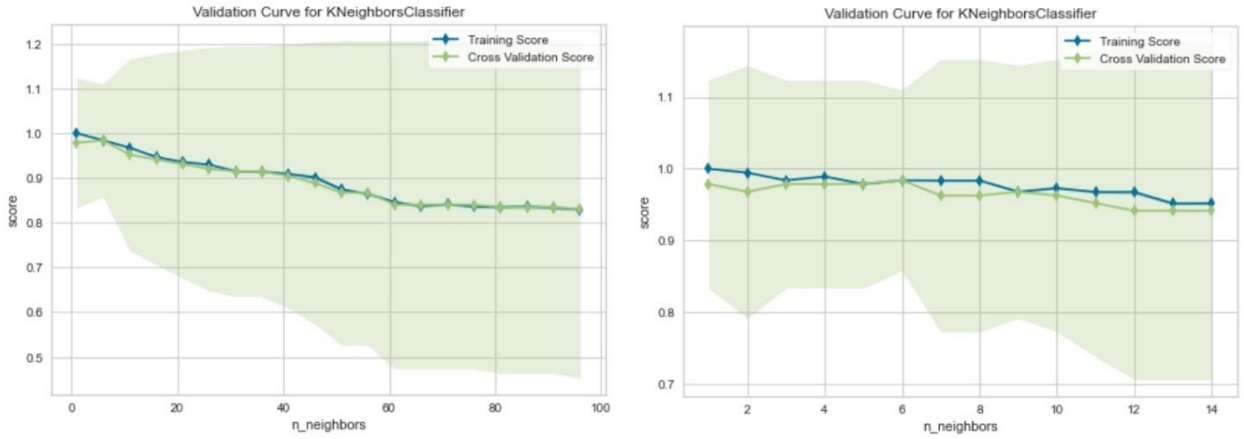
```
Best CV Score for Logistic Regression: 0.98 Parameters: {'penalty': 'none', 'solver': 'newton-cg'}.
Best CV Score for Decision Tree: 0.96 Parameters: {'criterion': 'gini', 'max_depth': 7}.
Best CV Score for Random Forest: 0.97 Parameters: {'criterion': 'gini', 'max_depth': 5, 'n_estimators': 13}.
Best CV Score for Support Vector Classifier: 0.99 Parameters: {'C': 2.5, 'degree': 1, 'kernel': 'linear'}.
Best CV Score for Gaussian Naive Bayes: 0.91 Parameters: {'var_smoothing': 0.0}.
Best CV Score for Multi-layer Perceptron: 0.98 Parameters: {'activation': 'relu', 'solver': 'lbfgs'}.
Best CV Score for K Neighbors Classifier: 0.98 Parameters: {'algorithm': 'auto', 'n_neighbors': 3, 'weights': 'uniform'}.
```

Εικόνα 43 Αποτελέσματα GridSearchCV για βελτιστοποίηση των κυριότερων παραμέτρων για τα Manually επιλεγμένα Features, με τις τιμές όπως είναι μετά από την εφαρμογή SMOTE και PCA

```
Best CV Score for Logistic Regression: 1.00 Parameters: {'penalty': 'l1', 'solver': 'liblinear'}.
Best CV Score for Decision Tree: 0.99 Parameters: {'criterion': 'entropy', 'max_depth': 5}.
Best CV Score for Random Forest: 0.99 Parameters: {'criterion': 'entropy', 'max_depth': 3, 'n_estimators': 10}.
Best CV Score for Support Vector Classifier: 1.00 Parameters: {'C': 0.5, 'degree': 1, 'kernel': 'linear'}.
Best CV Score for Gaussian Naive Bayes: 0.97 Parameters: {'var_smoothing': 0.0}.
Best CV Score for Multi-layer Perceptron: 1.00 Parameters: {'activation': 'identity', 'solver': 'lbfgs'}.
Best CV Score for K Neighbors Classifier: 1.00 Parameters: {'algorithm': 'auto', 'n_neighbors': 1, 'weights': 'uniform'}.
```

Εικόνα 44 Αποτελέσματα GridSearchCV για βελτιστοποίηση των κυριότερων παραμέτρων για τα Automatically επιλεγμένα Features, με τις τιμές όπως είναι μετά από την εφαρμογή SMOTE και PCA

Ενώ για τη βιβλιοθήκη Yellowbrick, η αντίστοιχη μέθοδος είναι η ValidationCurve, η οποία έχει το μειονέκτημα ότι μας επιτρέπει να ψάχνουμε την ιδανική τιμή μιας μόνο παραμέτρου, για έναν μόνο αλγόριθμο κάθε φορά. Μας προσφέρει όμως τη δυνατότητα να εκτυπώσουμε την απόδοση του μοντέλου για ένα συγκεκριμένο Metric ως προς τις τιμές της παραμέτρου.



Εικόνα 45 Δύο διαδοχικά πειράματα με εκτέλεση της ValidationCurve μεθόδου για την εύρεση του ιδανικού αριθμού Nearest Neighbors(n_neighbors) για τον αλγόριθμο KNN και χρήση του Metric F1-weighted για την αξιολόγηση του μοντέλου στο εκάστοτε Iteration

Για τις ανάγκες της διπλωματικής, χρησιμοποιήθηκαν τόσο η GridSearchCV της βιβλιοθήκης Scikit-learn για μαζική εξαγωγή βέλτιστων παραμέτρων για τους αλγόριθμους που έχουν χρησιμοποιηθεί μέχρι αυτό το στάδιο, όσο και η ValidationCurve της βιβλιοθήκης Yellowbrick για επιλεκτική επιβεβαίωση μερικών από τις παραμέτρους που δίνει η GridSearchCV ως βέλτιστες.

Τα αποτελέσματα των αλγορίθμων με τη χρήση των βέλτιστων παραμέτρων βρίσκονται στον πίνακα αποτελεσμάτων στο κεφάλαιο της αξιολόγησης.

4.2.4 Production Model Creation

Βάση του ερευνητικού ερωτήματος θα μπορούσαμε να σταματήσουμε σε αυτό το σημείο και να ασχοληθούμε μόνο με την αξιολόγηση των αποτελεσμάτων των μοντέλων που εκπαιδεύτηκαν κατά τη διαδικασία της EDA. Ωστόσο θα είχε ενδιαφέρον να προχωρήσουμε στην δημιουργία ενός τελικού μοντέλου το οποίο θα ονομάσουμε παραγωγικό (Production Model), έτσι ώστε μέσα από τη διαδικασία να εξετάσουμε τις προκλήσεις που θα μπορούσαν να προκύψουν.

Με τον όρο Production Model Creation (PMC) χαρακτηρίζουμε τη διαδικασία που λαμβάνει χώρα από τη στιγμή που έχουμε καταλήξει σε κάποια ασφαλή συμπεράσματα μέσα από τη διαδικασία της EDA.

Αυτό που θα μπορούσαμε να χαρακτηρίσουμε ως είσοδο στη διαδικασία PMC πέρα από το Dataset είναι τα συμπεράσματα στα οποία έχουμε καταλήξει κατά τη διαδικασία EDA, όπως είναι για παράδειγμα:

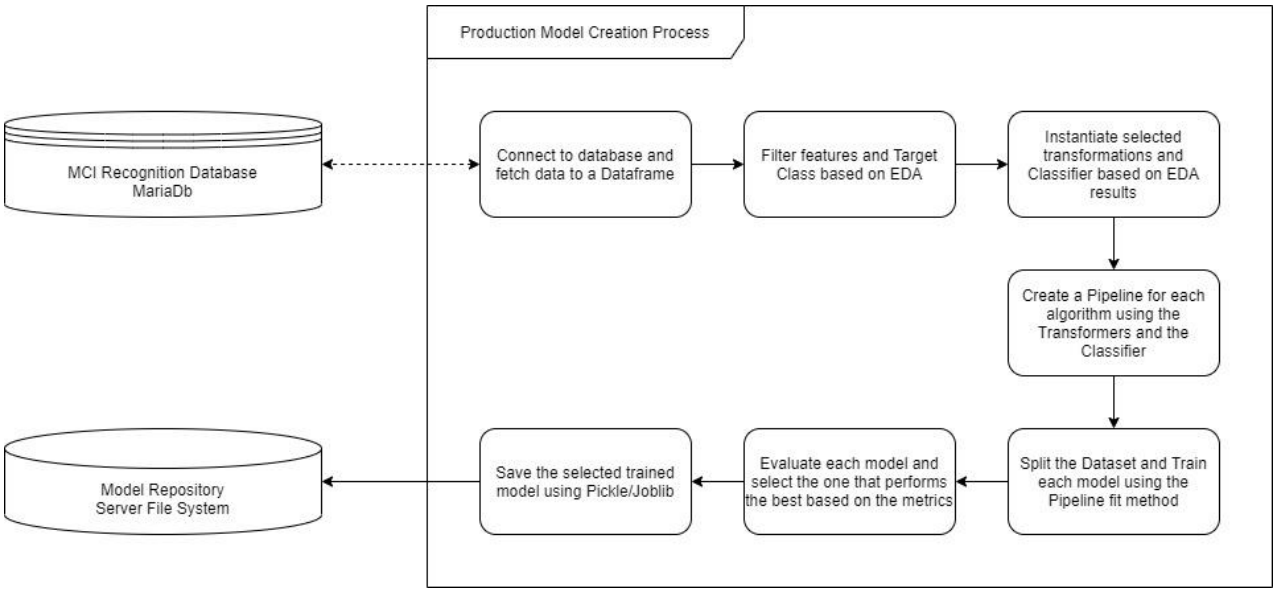
- Τα Features που έχουν αναδειχθεί ως τα πιο σημαντικά
- Το επιθυμητό Target Class
- Οι τεχνικές βελτιστοποίησης που παρατηρήσαμε ότι βελτιστοποιούν την απόδοση των μοντέλων

Ο λόγος για τον οποίο ως είσοδο δεν συμπεριλαμβάνουμε και τον βέλτιστο αλγόριθμο με βάση τα πειραματικά μοντέλα της διαδικασίας EDA, είναι διότι κατά τη PMC διαδικασία διαχειριζόμαστε το πρόβλημα του Data Leakage με τη χρήση των Pipelines και θέλουμε να αξιολογήσουμε την απόδοση του κάθε αλγόριθμου μετά από αυτή τη διαδικασία πριν καταλήξουμε σε ένα τελικό μοντέλο.

Ως έξοδο της διαδικασίας PMC, έχουμε την αποθήκευση ενός μόνο παραγωγικού μοντέλου, το οποίο έχουμε επιλέξει με βάση την απόδοση των παραγωγικών μοντέλων που δημιουργήσαμε στα Metrics

Accuracy για το Training και το Testing, καθώς και για τα Metrics Specificity και Sensitivity για το Testing. Τα επιμέρους βήματα της PMC διαδικασίας είναι τα ακόλουθα:

- Ανάκτηση δεδομένων από τη βάση δεδομένων.
- Επιλογή του Target Class που κρίναμε ως κατάλληλο κατά τη διαδικασία της EDA.
- Διαλογή των Features ώστε να παραμείνουν μόνο αυτά που επιλέξαμε κατά τη διαδικασία της EDA ως τα πιο σημαντικά.
- Δημιουργία πανομοιότυπων Transformer και Classifier Instances με αυτά που δημιουργήσαμε κατά τη διαδικασία της EDA.
- Διαχωρισμός του Dataset μεταξύ Training και Testing Dataset, καθώς θα χρειαστεί να αξιολογήσουμε το Production μοντέλο που παράγει ο κάθε αλγόριθμος.
- Έπειτα δημιουργία ενός Pipeline, της βιβλιοθήκης imblearn, για κάθε έναν από τους Classifiers.
- Εκπαίδευση και αξιολόγηση των παραγωγικών μοντέλων.
- Επιλογή του βέλτιστου μοντέλου και αποθήκευση του με τη χρήση της βιβλιοθήκης Pickle ώστε να μπορεί να ανακτηθεί από το Classification Service.



Εικόνα 46 Διάγραμμα διαδικασίας δημιουργίας ενός παραγωγικού μοντέλου

4.2.5 Classification Service API

Συγκριτικά με τα μοντέλα CRISP-DM και TDSP, θα μπορούσαμε να κατατάξουμε την υλοποίηση του Classification Service API στην ενότητα την οποία αναφέρουν και τα δύο μοντέλα ως Deployment. Στο σύνολο της μπορούμε να δούμε τη διαδικασία ως ένα Application Programming Interface (API), το οποίο αποτελείται από μια υπηρεσία (Service) το Classification Service και ένα Repository όπου βρίσκεται αποθηκευμένο το μοντέλο.

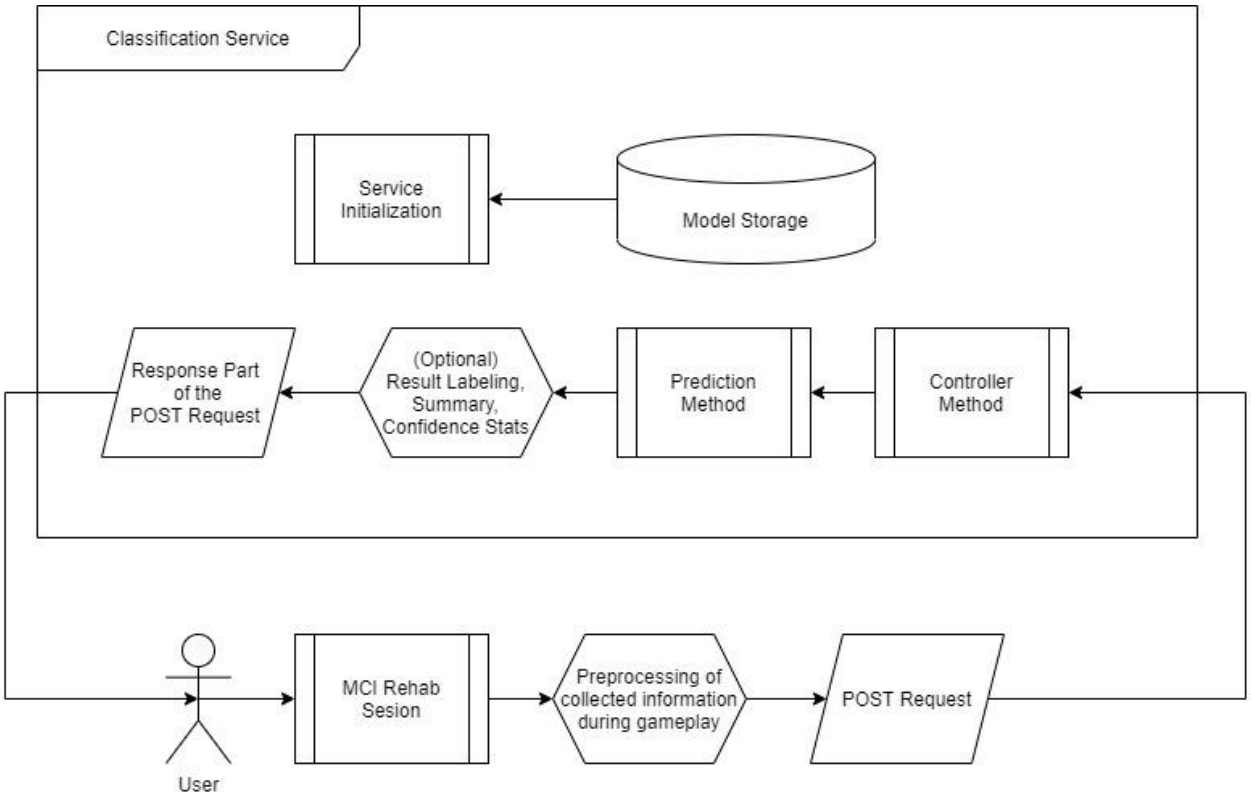
Το κύριο δομικό στοιχείο του Service είναι ένας Flask Server ο οποίος δέχεται REST Requests, επεξεργάζεται τα δεδομένα και επιστρέφει το αποτέλεσμα του Classification. Βασικό χαρακτηριστικό αυτής της σχεδίασης είναι ότι το μοντέλο βρίσκεται αποθηκευμένο στο ίδιο File System με αυτό του Flask Server.

Μια εναλλακτική σχεδίαση θα μπορούσε να είχε το μοντέλο ενσωματωμένο στην εφαρμογή MCI Rehab, δηλαδή στη μεριά του Client.

Πιο αναλυτικά, όσον αφορά τα REST Requests, αυτά τα προσομοιώνουμε με τη χρήση του προγράμματος Postman, λόγω του ότι κάνει την παραμετροποίηση ενός Request πιο εύκολη, ωστόσο θα μπορούσαμε να χρησιμοποιήσουμε οποιοδήποτε Browser. Αυτό που αντιπροσωπεύει ένα REST Request είναι ουσιαστικά η κλήση που θα κάνει η εφαρμογή MCI Rehab από τη συσκευή του χρήστη, μετά την ολοκλήρωση ενός Session. Τα δεδομένα θα αποστέλλονται προς το Server με ένα POST Request υπό τη μορφή ενός JSON Object στο Body του Request.

Για την εξυπηρέτηση αυτών των αιτημάτων έχει δημιουργηθεί ένας Server με χρήση της βιβλιοθήκης Flask. Ως υλοποίηση, έχει περιοριστεί στα απολύτως απαραίτητα τα οποία είναι πρώτον, η main μέθοδος όπου γίνεται η παραμετροποίηση του Server και η ανάκτηση του μοντέλου με χρήση της βιβλιοθήκης Joblib και δεύτερον, μια μέθοδος τύπου Controller για λήψη των POST Request στο Path που έχουμε ορίσει. Η ανάκτηση του μοντέλου μπορεί να γίνεται είτε με την αρχικοποίηση του Service, όπως και συμβαίνει στη υλοποίηση της διπλωματικής, είτε με κάθε λήψη Request εάν υποθέσουμε ότι μπορεί στο διάστημα μεταξύ δύο Request να έχουμε προχωρήσει σε εξαγωγή νέου μοντέλου. Με την λήψη του Request από τον Controller του Server, γίνεται η ανασύνθεση του αντικειμένου με χρήση της Utility Class SimpleNamespace και ενός νέου Dataframe της βιβλιοθήκης Pandas. Στη συνέχεια, χρησιμοποιούμε τη μέθοδο Predict του μοντέλου για να εκτελέσουμε Classification για αυτή τη νέα εγγραφή που βρίσκεται πλέον υπό τη μορφή Pandas Dataframe. Δεδομένου του ότι έχουμε εκπαιδεύσει το μοντέλο με Target Class τη δυαδική μορφή των αποτελεσμάτων του MOCA και μάλιστα έχοντας κάνει αντικατάσταση των Labels στο επίπεδο της βάσης δεδομένων και συγκεκριμένα στο View, μπορούμε προαιρετικά να κάνουμε ένα επιπλέον ερώτημα προς τη βάση δεδομένων για να ανακτήσουμε τα Labels, έτσι ώστε να επιστρέψουμε το αποτέλεσμα σε μια πιο ευπαρουσίαστη μορφή.

Τέλος, όσον αφορά το Classification Service, ένα εξίσου σημαντικό στοιχείο, στο οποίο αναφερθήκαμε και στο προηγούμενο κεφάλαιο, είναι η αυτόματη εκτέλεση των διεργασιών προεπεξεργασίας που επιλέξαμε να συμπεριλάβουμε στο Pipeline κατά τη δημιουργία του μοντέλου. Με άλλα λόγια, τα Transformations τύπου Discretization και Scaling είναι πλέον μέρος του μοντέλου που ανακτούμε διασφαλίζοντας έτσι ότι τα δεδομένα που θα δώσουμε στο μοντέλο για τη διαδικασία του Classification θα υποστούν τη παραπάνω προεπεξεργασία κατά την εκτέλεση της μεθόδου Prediction.



Εικόνα 47 Διάγραμμα που περιγράφει τη ροή της πληροφορίας μεταξύ χρήστη, εφαρμογής MCI Rehab και Classification Service, καθώς επίσης και της ροής της πληροφορίας στο εσωτερικό του Classification Service

4.2.6 Περιβάλλον ανάπτυξης

Όσον αφορά το περιβάλλον ανάπτυξης της εφαρμογής αυτό περιλαμβάνει όλα τα εργαλεία και τις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση. Ομαδοποιώντας όλα αυτά τα στοιχεία προκύπτουν τα εξής.

- Γλώσσες προγραμματισμού. Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για το κυρίως κομμάτι της υλοποίησης είναι η Python, ενώ παράλληλα έγινε χρήση της SQL για τη δημιουργία τόσο των Scripts της μετάπτωσης των δεδομένων, αλλά και της ανάκτησης από τη νέα βάση σε κάθε επανάληψη της EDA διαδικασίας.
- Βιβλιοθήκες. Η βιβλιοθήκη που χρησιμοποιήθηκε κατά κόρον είναι το Scikit-learn τόσο για τις διαδικασίες του Preprocessing όσο και για την εκπαίδευση των μοντέλων. Ωστόσο υπήρξε και μια σειρά άλλων βιβλιοθηκών που χρησιμοποιήθηκαν σε όλο το εύρος της υλοποίησης, όπως για παράδειγμα η βιβλιοθήκη Pandas για τη διαχείριση των δεδομένων. Οι βιβλιοθήκες Numpy και Scipy για μια σειρά υπολογισμών. Οι βιβλιοθήκες Seaborn και Matplotlib για εκτύπωση διαγραμμάτων. Η βιβλιοθήκη mysql για επικοινωνία με τη βάση δεδομένων. Οι βιβλιοθήκες joblib και pickle για την αποθήκευση και την ανάκτηση των μοντέλων, των Features και των Datasets. Η βιβλιοθήκη imblearn για τη χρήση ενός Pipeline που θα μπορεί να υποστηρίζει τη μέθοδο SMOTE. Και τέλος η βιβλιοθήκη re για τη χρήση Regular Expressions.
- Integrated Development Environments (IDEs). Στα πρώτα στάδια της υλοποίησης χρησιμοποιήθηκε το IDE Spyder, το οποίο στη συνέχεια αντικαταστάθηκε από το συνδυασμό των PyCharm και Jupyter Notebook. Ο τρόπος με τον οποίο χρησιμοποιήθηκαν είναι ο εξής, με

το PyCharm έγινε η συγγραφή μιας σειράς κλάσεων που περιείχαν Wrapper μεθόδους για τις μεθόδους του Scikit-learn, ενώ στα Jupyter Notebooks γίνεται το Instantiation αυτών των κλάσεων και η χρήση των Wrapper μεθόδων. Τέλος κάθε διακριτή διαδικασία της προσαρμοσμένης μεθοδολογίας αντιπροσωπεύεται από ένα Jupyter Notebook.

- Services. Επίσης για την υλοποίηση γίνεται χρήση ενός SQL Server τύπου MariaDB πάνω στον οποίο δημιουργείται το σχήμα της παρούσας διπλωματικής. Ενώ επίσης γίνεται χρήση ενός Flask Server ο οποίος παίζει το ρόλο του API όπως αυτό περιγράφεται στην ενότητα §4.2.5.
- Code Versioning. Τέλος ένα από τα σημαντικότερα στοιχεία της υλοποίησης είναι το Code Versioning και συγκεκριμένα το σύστημα GIT, το οποίο χρησιμοποιήθηκε ως μέθοδος ενημέρωσης του κώδικα για την εκάστοτε αλλαγή.

5

Αξιολόγηση Μοντέλων Ανίχνευσης MCI

Για την αξιολόγηση των πειραματικών και παραγωγικών μοντέλων που έχουμε δημιουργήσει, έχουμε στη διάθεσή μας πληθώρα Metrics τα οποία θα μας βοηθήσουν να συμπεράνουμε εάν τελικά υπάρχει θετική απάντηση στο βασικό ερευνητικό ερώτημα και ποιες συνθήκες οδηγούν στη δημιουργία ενός αποδοτικού μοντέλου.

5.1 Επεξήγηση αποτελεσμάτων

Στις δύο επόμενες ενότητες βρίσκονται οι συγκεντρωτικοί πίνακες με τις τιμές των Metrics για κάθε σενάριο που εξετάστηκε καθώς και οι αναλυτικές εκτυπώσεις από την αξιολόγηση των μοντέλων.

Στο πλαίσιο της αξιολόγησης των μοντέλων έχουν δημιουργηθεί δύο Custom Wrapper μέθοδοι για την εκτύπωση διαγραμμάτων που αφορούν τα Metrics και το Decision Surface του κάθε μοντέλου.

Όσον αφορά τα Metrics η μέθοδος είναι η "generate_metrics" της κλάσης MetricsMethods, η οποία για κάθε έναν από τους αλγόριθμους εκτυπώνει τα εξής διαγράμματα.

- Ένα Boxplot με τις τιμές των Metrics που θα επιλέξουμε. Ο συγκεκριμένος τύπος διαγράμματος, Boxplot, έχει επιλεγεί λόγω του ότι οι τιμές των Metrics υπολογίζονται εντός της Cross Validation μεθόδου "cross_val_score" της βιβλιοθήκης Scikit-learn, για την οποία έχουμε επιλέξει k-fold=5.
- Ένα Confusion Matrix μέσω της μεθόδου "confusion_matrix" της βιβλιοθήκης Scikit-learn. Τα αποτελέσματα της συγκεκριμένης μεθόδου δε βασίζονται σε Cross Validation διαδικασία, κατ' επέκταση τα αποτελέσματα αφορούν μια μεμονωμένη κατηγοριοποίηση.

Τέλος, εκτυπώνονται συγκεντρωτικά για όλους τους αλγόριθμους ένα διάγραμμα Area Under Curve και ένα διάγραμμα Precision-Recall Curve.

Ενώ όσον αφορά το Decision Surface, η Wrapper μέθοδος που δημιουργήθηκε είναι η "print_decision_surface" στη κλάση MetricsMethods η οποία εκτυπώνει το Decision Boundary ή αλλιώς Decision Surface του μοντέλου ως προς το Dataset. Σκοπός της συγκεκριμένης μεθόδου είναι να παρέχει μια επιπλέον οπτική όσον αφορά τον λόγο μεταξύ Bias και Variance καθώς και την απόδοση ενός μοντέλου. Η συγκεκριμένη εκτύπωση γίνεται σε γράφημα 2 διαστάσεων, κατ' επέκταση μπορεί να απεικονίσει σημεία

μόνο δύο μεταβλητών του Dataset, για αυτό το λόγο έχει περισσότερο νόημα να γίνεται μετά από την εφαρμογή μιας μεθόδου Dimensionality Reduction, όπου πλέον όλη η πληροφορία βρίσκεται συγκεντρωμένη σε δύο μεταβλητές, υποθέτοντας ότι εξάγουμε δύο Principal Components. Στις εικόνες που εκτυπώνεται το Decision Surface κάθε γραμμή αντιστοιχεί σε έναν αλγόριθμο. Το Decision Boundary είναι ίδιο και για τα τρία plot της κάθε γραμμής, ενώ αυτό που διαφέρει είναι τα σημεία που αποτυπώνονται, καθώς στα αριστερά έχουμε τα σημεία του Training Dataset με το οποίο εκπαιδεύτηκε το εκάστοτε μοντέλο, στο κέντρο έχουμε το Testing Dataset ώστε να έχουμε μια εικόνα για το πως κατηγοριοποιούνται τα συγκεκριμένα σημεία ενώ στα δεξιά έχουμε ολόκληρο το Dataset.

5.2 Αξιολόγηση Πειραματικών Μοντέλων

Στους Πίνακες 7 και 8 βρίσκονται αναλυτικά οι τιμές των Metrics βάσει των οποίων αξιολογήθηκαν συνολικά 8 αλγόριθμοι μηχανικής μάθησης για τη δημιουργία των πειραματικών μοντέλων.

Κάθε ένας από τους δύο πίνακες αποτελείται από μοντέλα που δημιουργήθηκαν σε 5 διαδοχικά πειράματα. Όπως είδαμε και στο κεφάλαιο της μεθοδολογίας, πρώτα δημιουργήθηκαν τα Baseline μοντέλα τα οποία αξιολογήθηκαν με βάση τα εξής Metrics.

- Accuracy του μοντέλου, κατά τη διαδικασία της εκπαίδευσης, για το Training Dataset.
- Accuracy του μοντέλου, κατά τη διαδικασία του ελέγχου, για το Testing Dataset.
- Specificity του μοντέλου, κατά τη διαδικασία του ελέγχου, για το Testing Dataset.

Όσον αφορά το Accuracy και το Specificity που προκύπτουν από τη διαδικασία Testing, παρατίθεται και το Standard Deviation, καθώς η διαδικασία περιλάμβανε Cross Validation με k-fold=5. Η αναλογία που χρησιμοποιήθηκε μεταξύ των δειγμάτων των Training και Testing Subsets αντίστοιχα, είναι 75% προς 25%. Επιπλέον έγινε εφαρμογή Stratified Sampling για τη διατήρηση της αναλογίας των κατηγοριών του Target Class στο εκάστοτε Subset.

Επίσης όσον αφορά το Metric Specificity, λόγω του ότι δεν υπήρχε έτοιμη μέθοδος στη βιβλιοθήκη του Scikit-learn, δημιουργήθηκε μια Custom μέθοδος για τον υπολογισμό του.

Στη συνέχεια η ίδια διαδικασία της αξιολόγησης επαναλήφθηκε για τα παρακάτω σενάρια.

- Μετά την εφαρμογή της μεθόδου SMOTE, στο Oversampled Dataset.
- Μετά την εφαρμογή της μεθόδου PCA, στο νέο Dataset με τα δύο Principal Components.
- Μετά την εφαρμογή τόσο της μεθόδου SMOTE αλλά και της PCA.
- Μετά την εφαρμογή των μεθόδων SMOTE, PCA αλλά και της βελτιστοποίησης των παραμέτρων των αλγορίθμων, βάση των αποτελεσμάτων της διαδικασίας HPO.

Ερμηνεύοντας τα συγκεντρωτικά αποτελέσματα στους πίνακες 7 και 8, παρατηρούμε ότι η πλειοψηφία των Baseline μοντέλων τείνει είτε να παρουσιάζει σημάδια Overfitting είτε σημάδια Underfitting.

Εφαρμόζοντας Oversampling με τον αλγόριθμο SMOTE παρατηρούμε σαφή βελτίωση στα μοντέλα που έχουν δημιουργηθεί με τους αλγόριθμους Logistic Regression, Multi-layer Perceptron για τα Manually επιλεγμένα Features ενώ για τα Features που επιλέχθηκαν μέσω της μεθόδου SelectKBest(chi2) παρατηρούμε βελτίωση στα ίδια αλλά και στο μοντέλο που δημιουργήθηκε με χρήση του αλγόριθμου SVC.

Εφαρμόζοντας Dimensionality Reduction με τον αλγόριθμο PCA, χωρίς κάποια άλλη τεχνική βελτιστοποίησης παρατηρούμε πολύ χαμηλές επιδόσεις σε όλα τα Metrics που καταγράφουμε και πιο συγκεκριμένα παρατηρούμε υψηλό Bias καθώς τα ποσοστά των Metrics είναι πολύ χαμηλά και για τα δύο διαφορετικά Set από Features.

Για το συνδυασμό των τεχνικών SMOTE και PCA όπου πλέον παρατηρούμε τις καλύτερες επιδόσεις που έχουμε καταγράψει για τα πειραματικά μοντέλα πλην αυτού που εκπαιδεύτηκε με τον αλγόριθμο Decision Tree, το ίδιο συμπέρασμα ισχύει και για τα δύο διαφορετικά Set από Features.

Για τα αποτελέσματα που προκύπτουν από τη συνδυαστική χρήση των τεχνικών SMOTE, PCA και HPO παρατηρούμε ότι η επιπλέον εξειδίκευση των παραμέτρων των αλγορίθμων δεν βελτιώνει τις επιδόσεις των παραγόμενων μοντέλων πλην των εξαιρέσεων του Multi-layer Perceptron για τα Manually επιλεγμένα Features όπου παρατηρούμε μείωση των επιδόσεων και του k-Nearest Neighbor για τα Features που επιλέχθηκαν μέσω της μεθόδου SelectKBest(chi2) όπου παρατηρούμε αύξηση των επιδόσεων στα Metrics.

Αυτό που προκύπτει ως συμπέρασμα για τα πειραματικά μοντέλα είναι ότι σε κάθε περίπτωση ο συνδυασμός της εφαρμογής Oversampling με τη μέθοδο SMOTE και το Dimensionality Reduction με τη μέθοδο PCA δείχνει να δημιουργεί μοντέλα με ικανοποιητικές επιδόσεις στα Metrics. Ενώ πιο συγκεκριμένα για κάθε ένα από τα Set των Features για τα μεν Manually επιλεγμένα δεν ξεχωρίζει ιδιαίτερα κάποιος αλγόριθμος μεταξύ των Logistic Regression, SVC, Multi-layer Perceptron, k-Nearest Neighbor και Custom Ensemble, για τα δε αυτόματα επιλεγμένα μέσω SelectKBest(chi2) Features ξεχωρίζει ο αλγόριθμος Random Forest τόσο για το υψηλό Accuracy όσο και για το υψηλό Specificity.

Manually Selected Features	**Baseline Models**									
	Training	Testing								
Algorithm	Accuracy	Accuracy	Acc. Std.	Specificity	Spec. Std.					
Logistic Regression	100	100	0	100	0					
Decision Tree	100	100	0	100	0					
Random Forest	100	100	0	100	0					
Support Vector Classifier	93.33	93.33	4.71	100	0					
Gaussian Naive Bayes	100	100	0	100	0					
Multi-layer Perceptron	90	90	8.16	91.67	11.79					
k-Nearest neighbors	76.67	76.67	9.43	75	10.21					
Custom Ensemble	100	100	0	100	0					
Manually Selected Features	SMOTE					PCA				
	Training	Testing				Training	Testing			
Algorithm	Accuracy	Accuracy	Acc. Std.	Specificity	Spec. Std.	Accuracy	Accuracy	Acc. Std.	Specificity	Spec. Std.
Logistic Regression	97.78	97.78	3.14	95.24	6.73	70	70	14.14	87.5	17.68
Decision Tree	100	100	0	100	0	76.67	76.67	12.47	83.33	15.59
Random Forest	100	100	0	100	0	80	80	8.16	91.67	5.89
Support Vector Classifier	97.78	97.78	3.14	95.24	6.73	80	80	0	100	0
Gaussian Naive Bayes	100	100	0	100	0	70	70	8.16	79.17	15.59
Multi-layer Perceptron	97.78	97.78	3.14	95.24	6.73	73.33	73.33	9.43	87.5	17.68
k-Nearest neighbors	85.14	85.14	12.8	69.64	25.55	76.67	76.67	4.71	87.5	10.21
Custom Ensemble	100	100	0	100	0	73.33	73.33	17	83.33	15.59
Manually Selected Features	SMOTE & PCA					SMOTE & PCA & Optimization				
	Training	Testing				Training	Testing			
Algorithm	Accuracy	Accuracy	Acc. Std.	Specificity	Spec. Std.	Accuracy	Accuracy	Acc. Std.	Specificity	Spec. Std.
Logistic Regression	95.56	95.56	6.29	90.48	13.47	95.56	93.33	8.89	92	16
Decision Tree	85.14	85.14	2.77	82.14	12.71	85.14	85.11	12.48	84	19.6
Random Forest	93.47	93.47	5.45	90.48	13.47	91.39	89.11	7.04	92	16
Support Vector Classifier	95.56	95.56	6.29	90.48	13.47	95.56	95.56	8.89	92	16
Gaussian Naive Bayes	93.47	93.47	5.45	90.48	13.47	93.47	93.56	8.79	92	16
Multi-layer Perceptron	95.56	95.56	6.29	90.48	13.47	91.39	89.11	7.04	88	16
k-Nearest neighbors	95.56	95.56	6.29	90.48	13.47	95.56	95.56	8.89	92	16
Custom Ensemble	95.56	95.56	6.29	90.48	13.47	95.56	95.56	8.89	92	16

Πίνακας 7 Τιμές των Metrics για Manually Selected Features

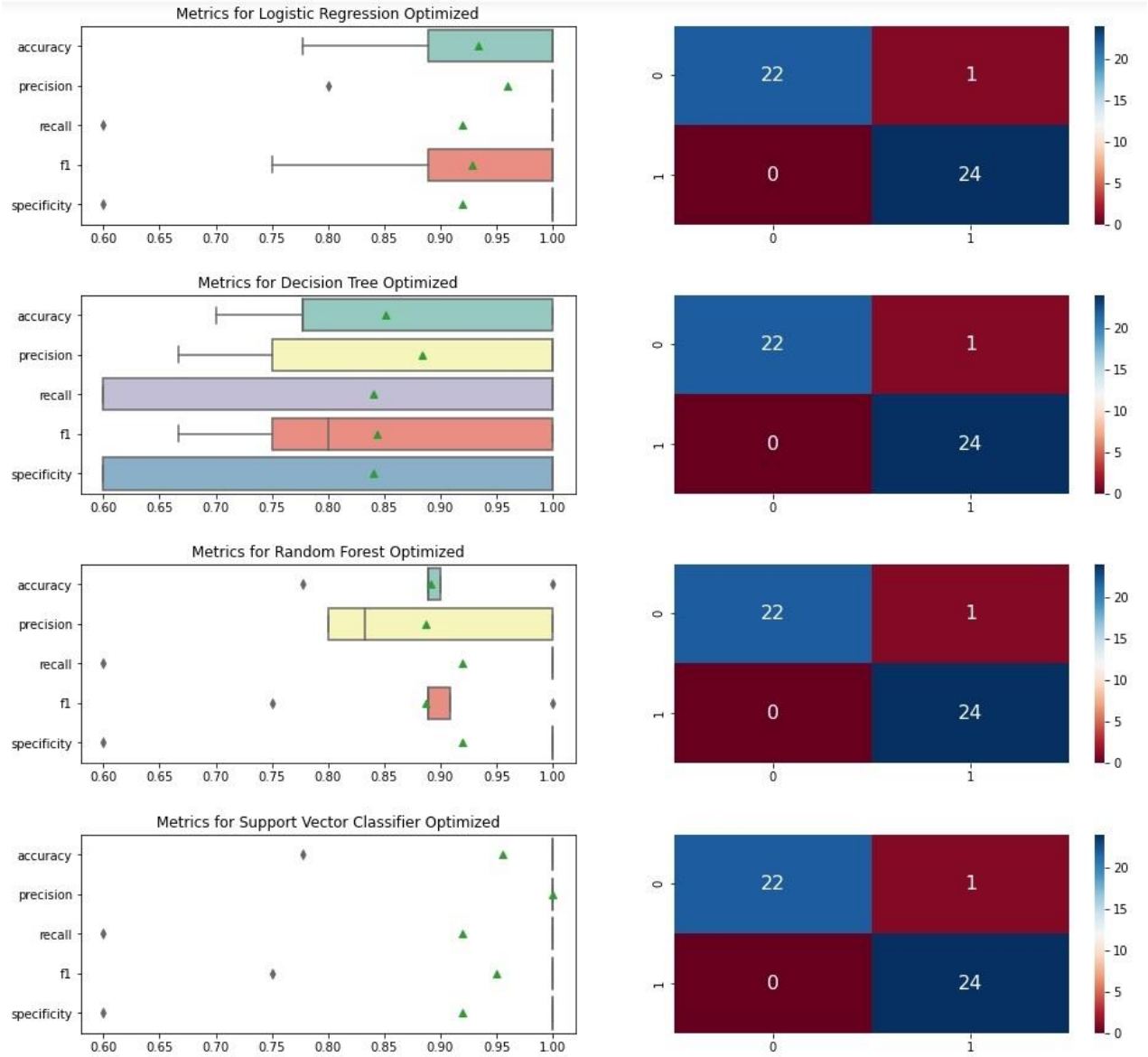
SelectKBest (Chi2) Selected Features	**Baseline Models**									
	Training	Testing								
Algorithm	Accuracy	Accuracy	Acc. Std.	Specificity	Spec. Std.					
Logistic Regression	93.33	93.33	13.33	95	10					
Decision Tree	96.67	96.67	6.67	100	0					
Random Forest	96.67	96.67	6.67	100	0					
Support Vector Classifier	93.33	93.33	8.16	96	8					
Gaussian Naive Bayes	100	100	0	100	0					
Multi-layer Perceptron	90	93.33	8.16	96	8					
k-Nearest neighbors	96.67	93.33	8.16	91	11.14					
Custom Ensemble	96.67	96.67	6.67	100	0					
SelectKBest (Chi2) Selected Features		SMOTE					PCA			
	Training	Testing				Training	Testing			
Algorithm	Accuracy	Accuracy	Acc. Std.	Specificity	Spec. Std.	Accuracy	Accuracy	Acc. Std.	Specificity	Spec. Std.
Logistic Regression	97.92	98	4	96	8	76.67	73.33	13.33	86	19.6
Decision Tree	100	100	0	100	0	80	73.33	8.16	79	12.81
Random Forest	100	100	0	100	0	86.67	90	13.33	95	10
Support Vector Classifier	100	97.778	4.44	100	0	86.67	83.33	0	96	8
Gaussian Naive Bayes	100	100	0	100	0	96.67	96.67	6.67	100	0
Multi-layer Perceptron	97.92	98	4	96	8	86.67	86.67	19.44	90	20
k-Nearest neighbors	85.28	91.56	7.62	83	15.36	83.33	76.67	22.61	74	24.98
Custom Ensemble	100	100	0	100	0	86.67	86.67	12.47	91	11.14
SelectKBest (Chi2) Selected Features		SMOTE & PCA					SMOTE & PCA & Optimization			
	Training	Testing				Training	Testing			
Algorithm	Accuracy	Accuracy	Acc. Std.	Specificity	Spec. Std.	Accuracy	Accuracy	Acc. Std.	Specificity	Spec. Std.
Logistic Regression	95.83	95.78	5.18	91	11.14	95.83	95.78	5.18	91	11.14
Decision Tree	93.61	89.78	10.96	84	23.32	93.61	89.78	10.96	84	23.32
Random Forest	**97.78**	**97.78**	**4.44**	**96**	**8**	**97.78**	**97.78**	**4.44**	**96**	**8**
Support Vector Classifier	95.83	98	4	96	8	100	100	0	100	0
Gaussian Naive Bayes	95.69	95.78	5.18	100	0	95.69	95.78	5.18	100	0
Multi-layer Perceptron	95.83	95.78	5.18	91	11.14	100	100	0	100	0
k-Nearest neighbors	91.67	91.33	8.27	82	18.33	95.83	95.78	5.18	92	9.8
Custom Ensemble	97.92	93.78	5.1	92	9.8	90	86.67	12.47	91	11.14

Πίνακας 8 Τιμές των Metrics για SelectKBest (Chi2) Selected Features

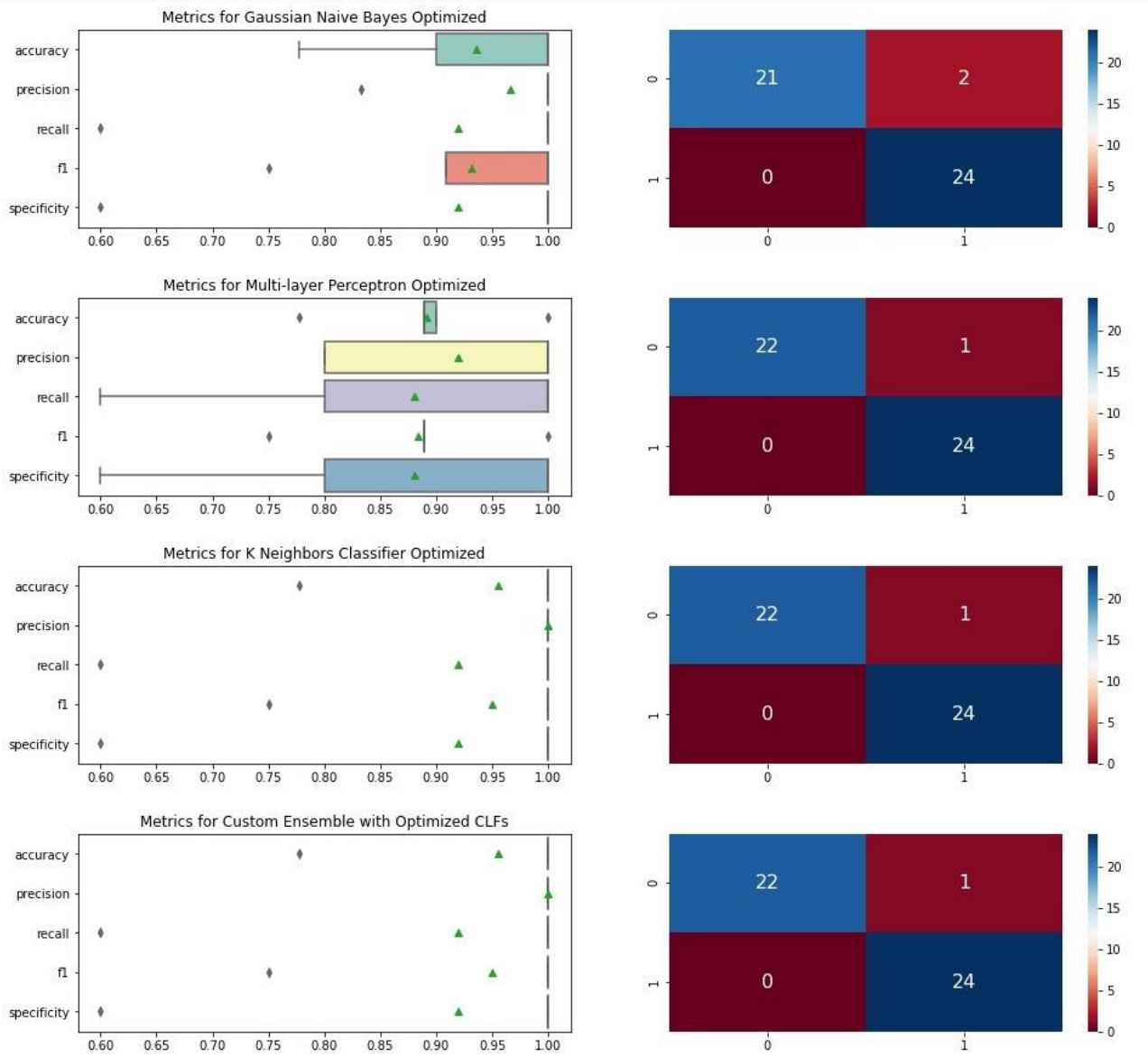
Στις εικόνες που ακολουθούν παρατηρούμε σε Custom διαγράμματα τα αποτελέσματα της Wrapper μεθόδου “generate_metrics” η οποία για κάθε έναν αλγόριθμο εκτυπώνει στα αριστερά ένα Boxplot με το αποτέλεσμα της εφαρμογής της μεθόδου “cross_val_score” για 5-fold Cross-Validation, ενώ στα δεξιά εκτυπώνει ένα Confusion Matrix για μια μεμονωμένη πρόβλεψη, με άλλα λόγια μια μεμονωμένη εκτέλεση της “predict” μεθόδου, του μοντέλου.

Επιπλέον, μετά από τα Boxplot και τα Confusion Matrix, η μέθοδος εκτυπώνει δύο συγκεντρωτικά διαγράμματα τύπου Area Under Curve και Precision-Recall για όλους τους αλγόριθμους που χρησιμοποιήθηκαν.

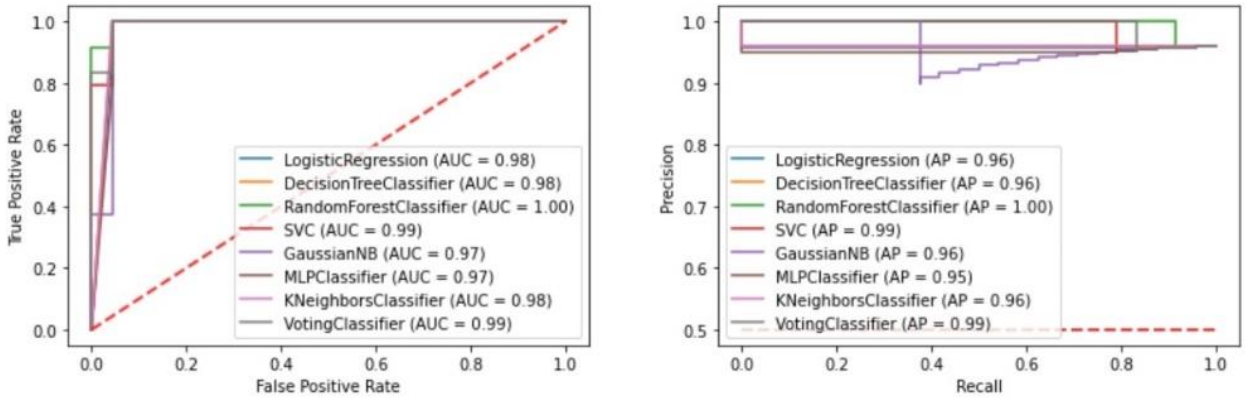
Τέλος η εκτέλεση της “generate_metrics” επαναλαμβάνεται για κάθε ένα από τα δύο Feature Set που έχουμε δημιουργήσει, όπως αυτά περιγράφονται στην ενότητα §4.2.3.3.



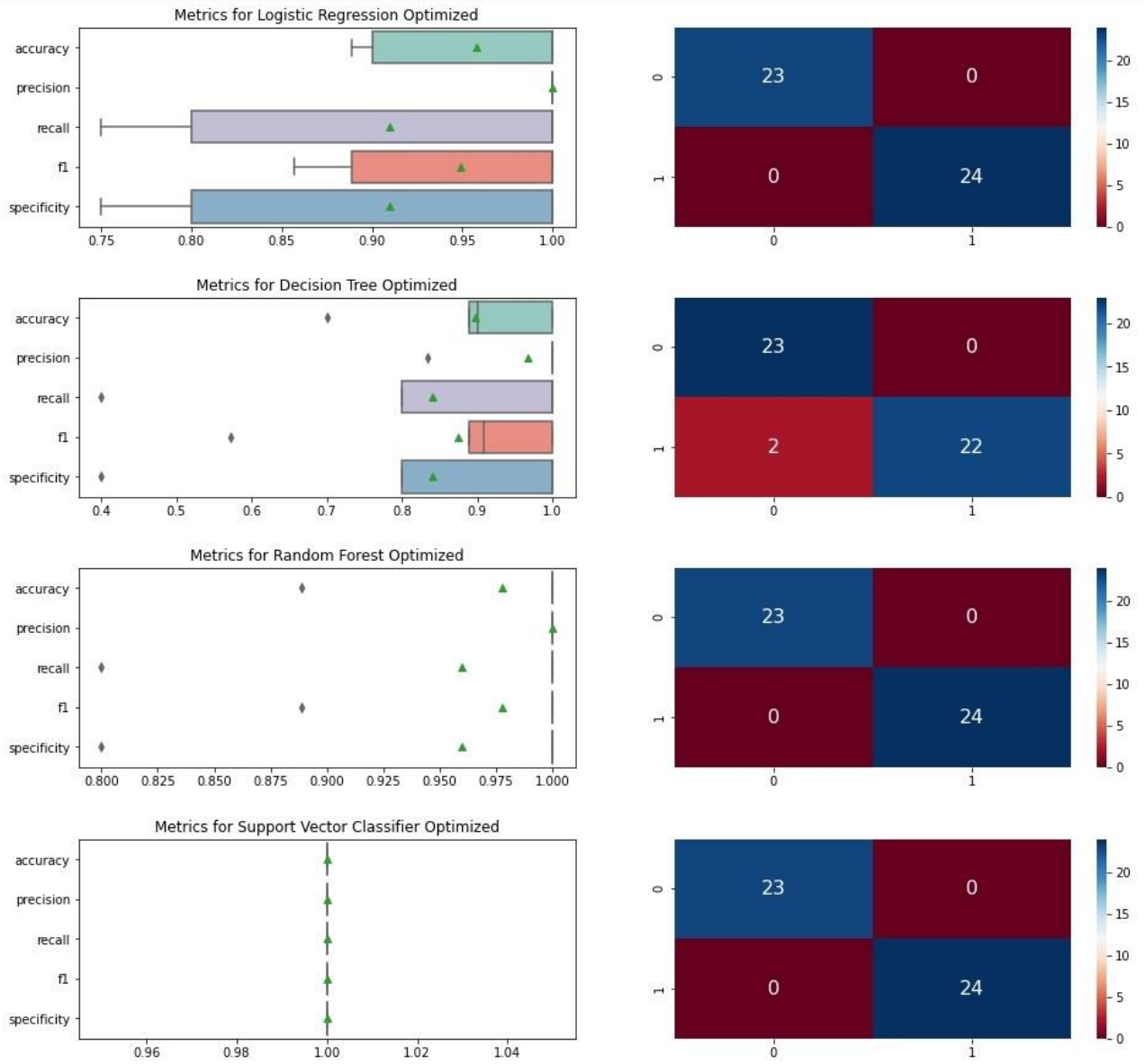
Εικόνα 48 Αριστερή στήλη: Boxplot με τιμές για Accuracy, Precision, Recall, F-Score, Specificity Δεξιά στήλη: Confusion Matrix για τα μοντέλα Logistic Regression, Decision Tree, Random Forest, SVC, τα οποία εκπαιδεύτηκαν με τα Manually επιλεγμένα Features και μετά από τις βελτιστοποιήσεις SMOTE, PCA και HPO



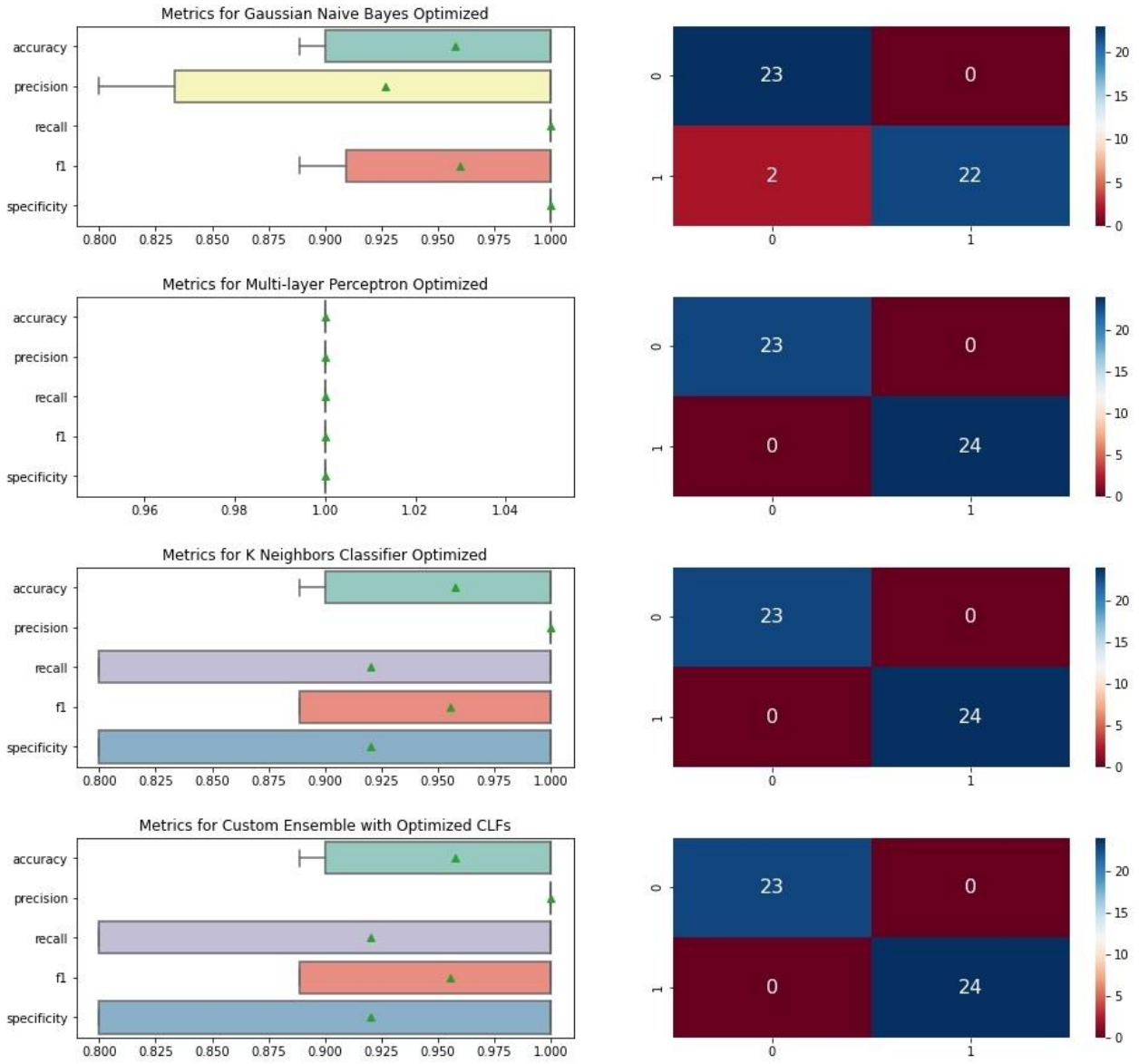
Εικόνα 49 Αριστερή στήλη: Boxplot με τιμές για Accuracy, Precision, Recall, F-Score, Specificity Δεξιά στήλη: Confusion Matrix για τα μοντέλα Gaussian Naive Bayes, K Neighbors Classifier, Custom Ensemble τα οποία εκπαιδεύτηκαν με τα Manually επιλεγμένα Features και μετά από τις βελτιστοποιήσεις SMOTE, PCA και HPO



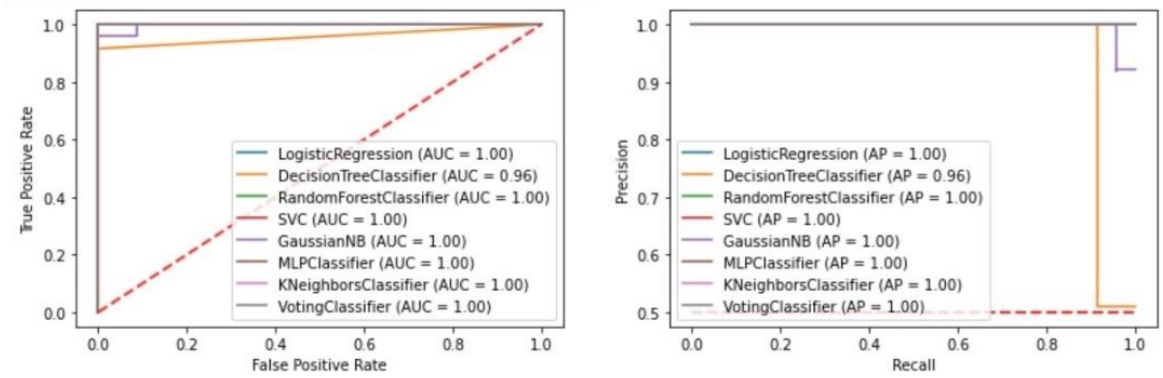
Εικόνα 50 Area Under Curve και Precision-Recall διαγράμματα, αριστερά και δεξιά αντίστοιχα, για όλα τα μοντέλα, εκπαιδευμένα με τα Manually επιλεγμένα Features και μετά από τις βελτιστοποιήσεις SMOTE, PCA και HPO



Εικόνα 51 Αριστερή στήλη: Boxplot με τιμές για Accuracy, Precision, Recall, F-Score, Specificity Δεξιά στήλη: Confusion Matrix για τα μοντέλα Logistic Regression, Decision Tree, Random Forest, SVC εκπαιδευμένα με τα Automatically επιλεγμένα Features μέσω SelectKBest (chi2) και μετά από τις βελτιστοποιήσεις SMOTE, PCA και HPO

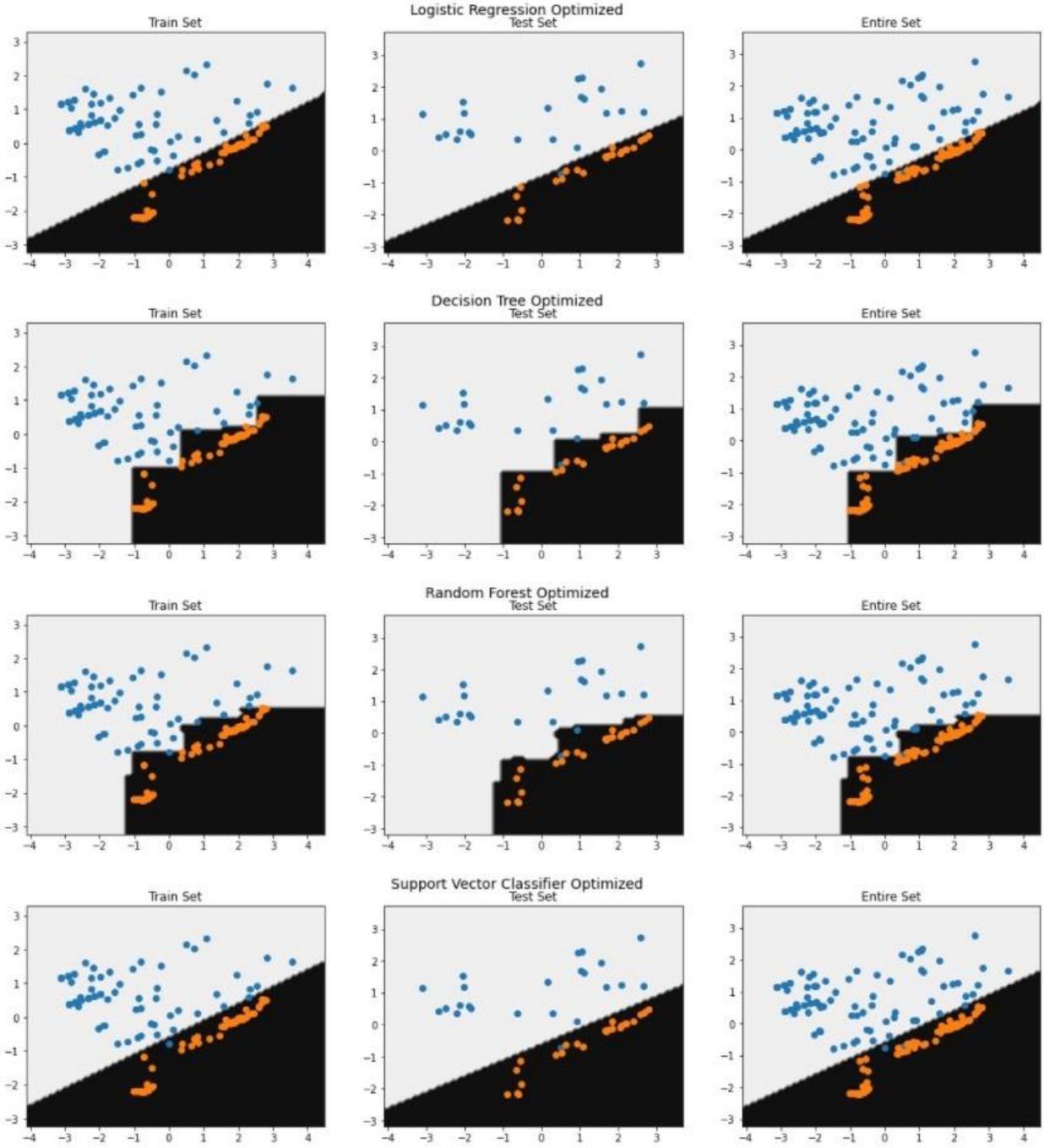


Εικόνα 52 Αριστερή στήλη: Boxplot με τιμές για Accuracy, Precision, Recall, F-Score, Specificity Δεξιά στήλη: Confusion Matrix για τα μοντέλα Gaussian Naive Bayes, K Neighbors Classifier, Custom Ensemble εκπαιδευμένα με τα Automatically επιλεγμένα Features μέσω SelectKBest (chi2) και μετά από τις βελτιστοποιήσεις SMOTE, PCA και HPO

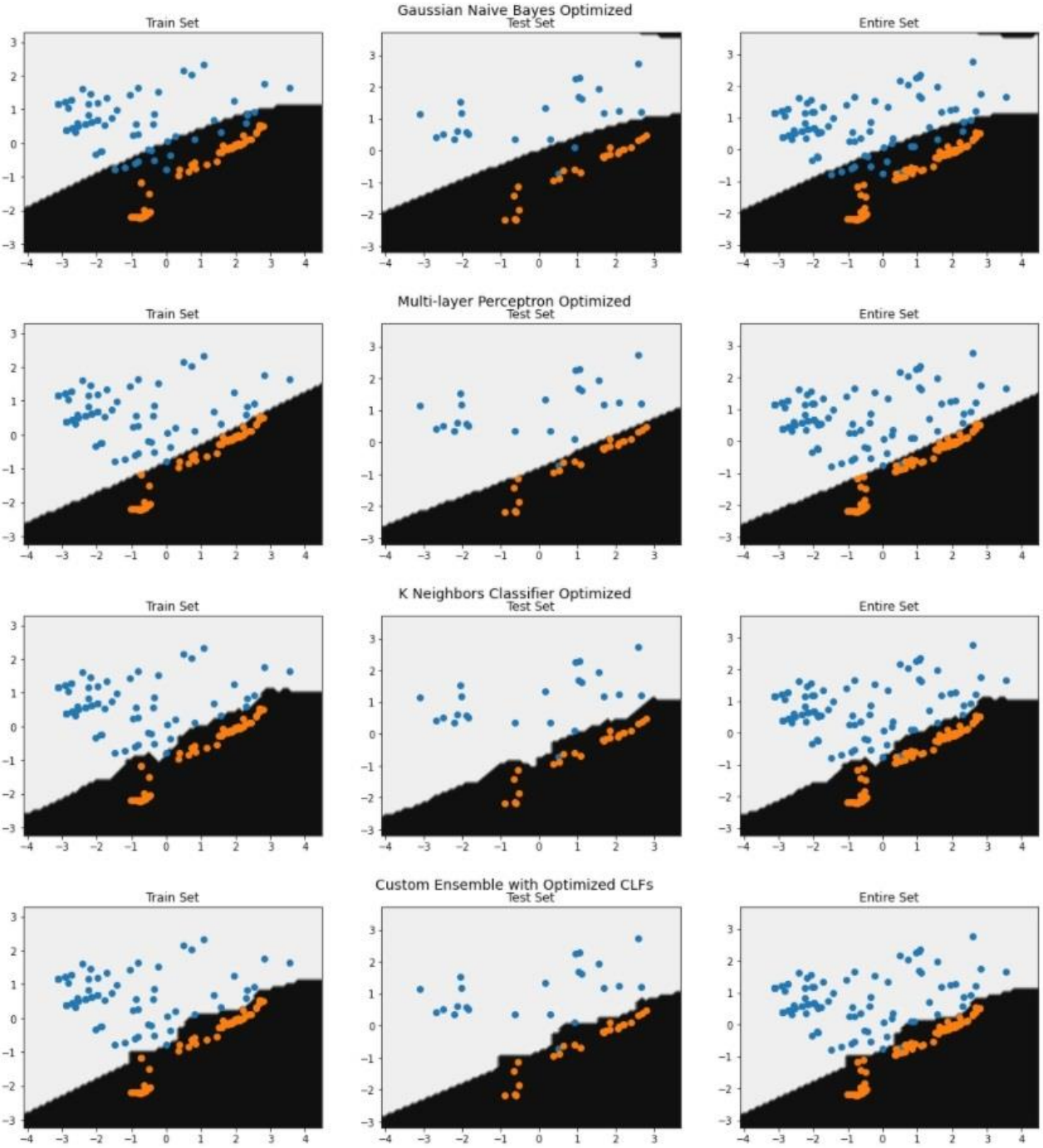


Εικόνα 53 Area Under Curve και Precision-Recall διαγράμματα, αριστερά και δεξιά αντίστοιχα, για όλα τα μοντέλα, εκπαιδευμένα με τα SelectKBest (chi2) Automatically επιλεγμένα Features και μετά από τις βελτιστοποιήσεις SMOTE, PCA και HPO

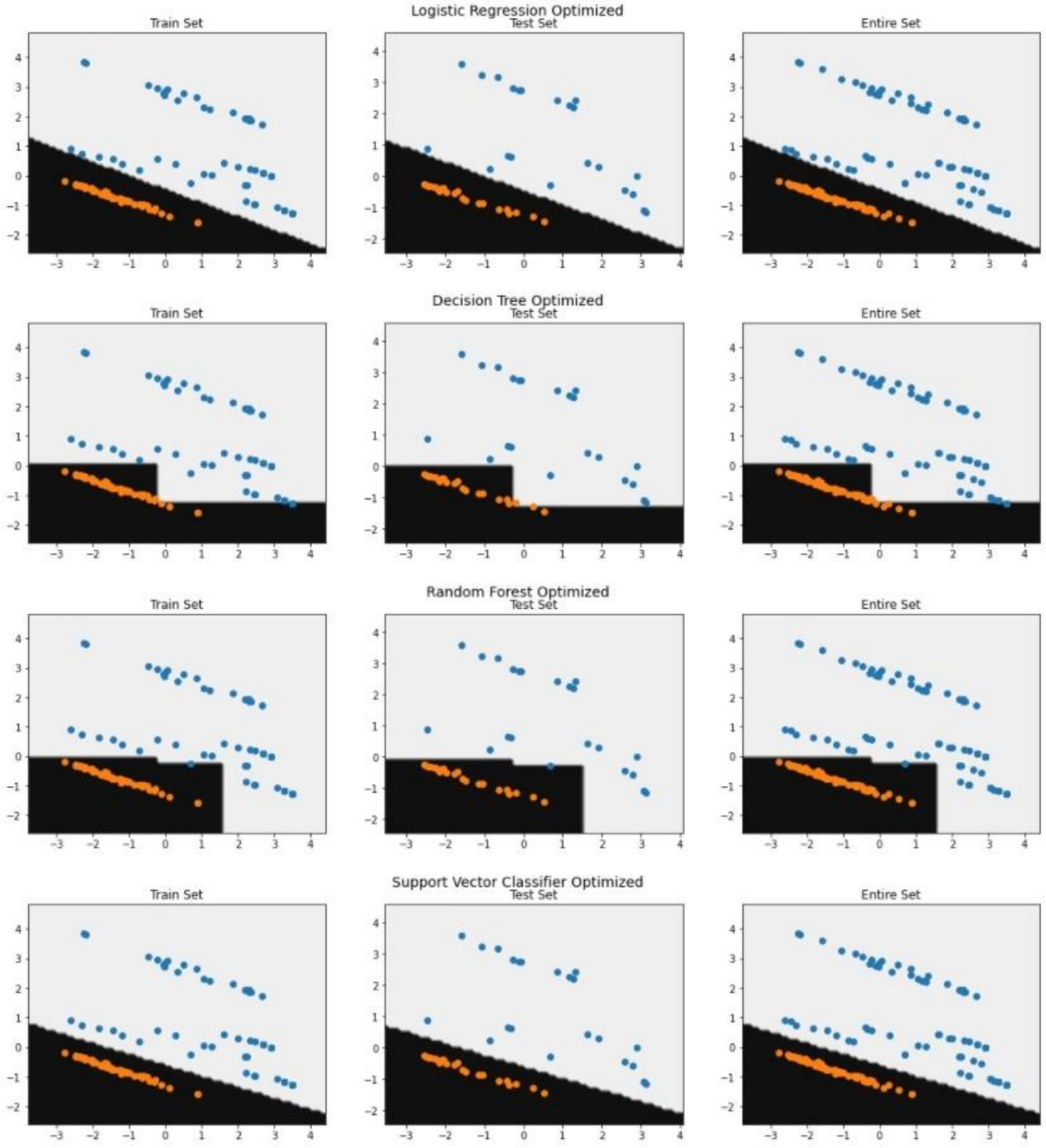
Οι εικόνες που ακολουθούν είναι το αποτέλεσμα της εκτέλεσης της Wrapper μεθόδου "print_decision_surface" που έχουμε δημιουργήσει. Η μέθοδος εκτυπώνει τρία διαγράμματα για κάθε αλγόριθμο που εξετάζουμε, στα αριστερά για το Training Dataset, στο κέντρο για το Testing Dataset και τέλος στα δεξιά για το ολόκληρο το Dataset. Η μέθοδος έχει εκτελεστεί κατόπιν της εφαρμογής της μεθόδου PCA. Αυτό που απεικονίζεται ουσιαστικά είναι το Decision Surface το οποίο αποτελείται από το Decision Boundary μεταξύ των δύο τμημάτων που αντιπροσωπεύουν τις δύο κατηγορίες [46]. Τα μεμονωμένα σημεία απεικονίζουν εγγραφές του Dataset βάση των τιμών που έχουν στα δύο Principal Components.



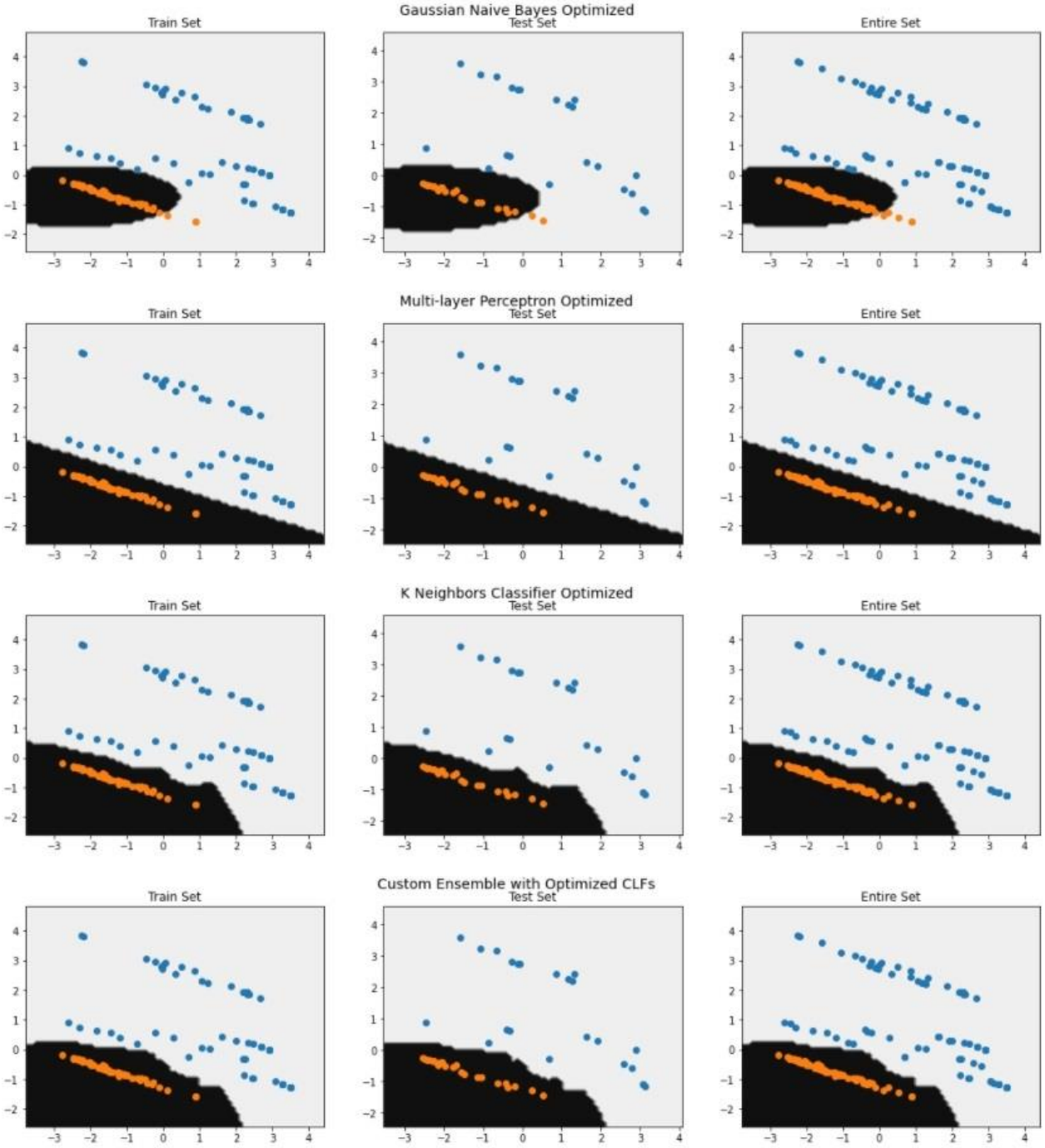
Εικόνα 54 Decision Surface για τα πειραματικά μοντέλα που έχουν δημιουργηθεί με τους αλγόριθμους Logistic Regression, Decision Tree, Random Forest, Support Vector Classifier, για τα Manually επιλεγμένα Features και κατόπιν των μεθόδων βελτιστοποίησης SMOTE, PCA και HPO



Εικόνα 55 Decision Surface για τα πειραματικά μοντέλα που έχουν δημιουργηθεί με τους αλγόριθμους Gausian Naive Bayes, Multi-layer Perceptron, K Neighbors Classifier και Custom Ensemble, για τα Manually επιλεγμένα Features και κατόπιν των μεθόδων βελτιστοποίησης SMOTE, PCA και HPO



Εικόνα 56 Decision Surface για τα πειραματικά μοντέλα που έχουν δημιουργηθεί με τους αλγόριθμους Logistic Regression, Decision Tree, Random Forest, Support Vector Classifier, για τα Automatically επιλεγμένα Features μέσω SelectKBest (chi2) και κατόπιν των μεθόδων βελτιστοποίησης SMOTE, PCA και HPO



Εικόνα 57 Decision Surface για τα πειραματικά μοντέλα που έχουν δημιουργηθεί με τους αλγόριθμους Gausian Naive Bayes, Multi-layer Perceptron, K Neighbors Classifier και Custom Ensemble, για τα Automatically επιλεγμένα Features μέσω SelectKBest (chi2) και κατόπιν των μεθόδων βελτιστοποίησης SMOTE, PCA και HPO.

5.3 *Αξιολόγηση Παραγωγικών Μοντέλων*

Στους Πίνακες 9 και 10, βρίσκονται αναλυτικά οι τιμές των Metrics βάση των οποίων αξιολογήθηκαν συνολικά 7 αλγόριθμοι μηχανικής μάθησης για τη δημιουργία των παραγωγικών μοντέλων.

Η διαφορά με την αξιολόγηση των πειραματικών μοντέλων είναι ότι δεν έχουμε συμπεριλάβει τον αλγόριθμό Custom Ensemble.

Επίσης μια δεύτερη διαφορά είναι ότι έχει πραγματοποιηθεί μια μόνο εκπαίδευση σε ένα Dataset για κάθε αλγόριθμο, ανά Feature Set, συγκεκριμένα στη μορφή που αυτό έχει μετά την εφαρμογή των μεθόδων βελτιστοποίησης SMOTE, PCA και HPO.

Επιπλέον μια ακόμη διαφορά με την αξιολόγηση των πειραματικών μοντέλων είναι ότι στους συγκεντρωτικούς πίνακες καταγράφεται και το Metric Sensitivity με την αντίστοιχη τυπική του απόκλιση.

Τέλος θα πρέπει να ληφθεί υπόψη ότι ο λόγος μεταξύ των Training και Testing Subsets για τα παραγωγικά μοντέλα ήταν στο 70% και 30% αντίστοιχα. Η διαφοροποίηση αυτή έγινε λόγω του ότι με τη χρήση των Pipelines ο διαχωρισμός μεταξύ Training και Testing Subset πραγματοποιείται πριν την εφαρμογή του Oversampling. Οπότε, αυξάνοντας το ποσοστό του Testing Subset, μειώνουμε τη πιθανότητα να μην υπάρχουν δείγματα της μειοψηφικής κατηγορίας στο Testing Subset, για κάποια από τα Folds του Cross-Validation.

Ερμηνεύοντας τα αποτελέσματα των παραγωγικών μοντέλων για το Set των Manually επιλεγμένων Features ότι οι τα μοντέλα που δημιουργήθηκαν με τους αλγόριθμους Logistic Regression, Decision Tree και Multi-layer Perceptron παρουσιάζουν υψηλό Variance κατά το Training και άρα τα αντίστοιχα παραγόμενα μοντέλα τους είναι κατά πάσα πιθανότητα Overfitted. Στον αντίποδα, ο αλγόριθμος Gaussian Naive Bayes παράγει ένα μοντέλο με υψηλό Bias. Ενώ τα μοντέλα που δείχνουν ιδανικά για αυτό το Feature Set είναι αυτά που προκύπτουν από τους αλγόριθμους k-Nearest Neighbor και SVC.

Manually Selected Features	**Production Models**						
	Training	Testing					
Algorithm	Accuracy	Accuracy	Acc. Std.	Sensitivity	Sens. Std.	Specificity	Spec. Std.
Logistic Regression	100	91.79	6.74	96.6	6.8	70	40
Decision Tree	100	86.07	9.06	96.6	6.8	50	44.72
Random Forest	98.79	88.93	10.62	96.6	6.8	70	40
Support Vector Classifier	**98.79**	**91.79**	**6.74**	**93.20**	**8.33**	**90**	**20**
Gaussian Naive Bayes	84.33	83.57	10	93.20	8.33	60	37.42
Multi-layer Perceptron	100	94.64	6.59	96.60	6.8	90	20
k-Nearest neighbors	**98.79**	**89.29**	**9.58**	**90**	**13.25**	**90**	**20**

Πίνακας 9 Τιμές των Metrics για τα παραγωγικά μοντέλα για τα Manually Selected Features

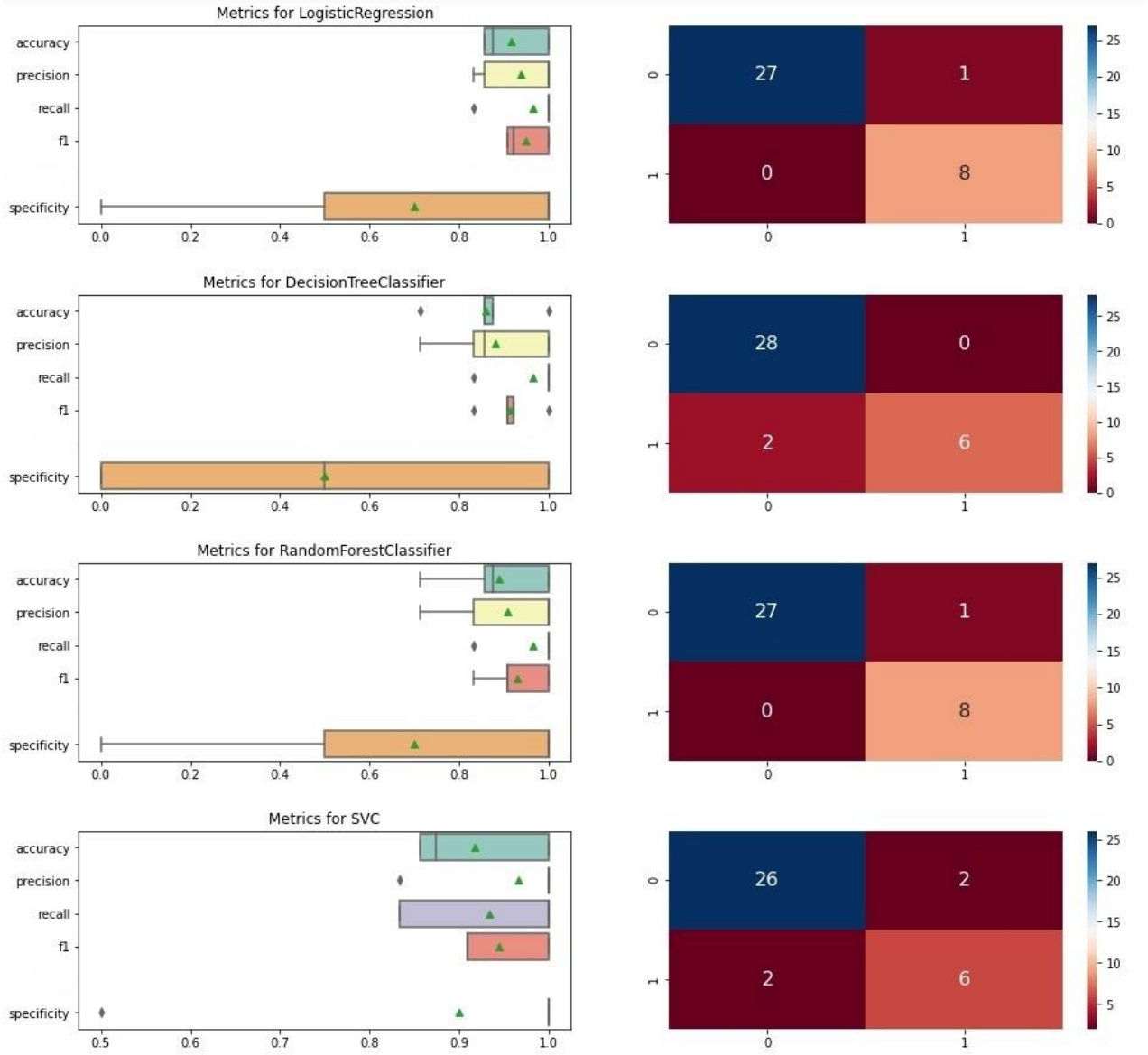
Όσον αφορά τα αποτελέσματα για το SelectKBest (Chi2) Feature Set, παρατηρούμε ότι οι αλγόριθμοι Decision Tree, Random Forest, k-Nearest Neighbors και Multi-layer Perceptron παρουσιάζουν πολύ υψηλό Variance κατά την εκπαίδευση και πολύ χαμηλές επιδόσεις κατά το Testing συνεπώς δεν παράγουν αξιόπιστα μοντέλα. Τα μοντέλα που παράγουν οι αλγόριθμοι Logistic Regression και SVC δεν παρουσιάζουν υψηλό Bias ή Variance κατά την εκπαίδευση ωστόσο έχουν χαμηλές επιδόσεις κατά τη διαδικασία του Testing. Τέλος το μοντέλο που παράγει ο αλγόριθμος Gaussian Naive Bayes δείχνει να είναι το ιδανικό καθώς χωρίς να παρουσιάζει υψηλό Variance έχει μεγάλο Accuracy τόσο κατά το Training όσο και στο Testing, ενώ παράλληλα παρουσιάζει υψηλές επιδόσεις στα Metrics Sensitivity και Specificity.

SelectKBest (Chi2) Selected Features	**Production Models**						
	Training	Testing					
Algorithm	Accuracy	Accuracy	Acc. Std.	Sensitivity	Sens. Std.	Specificity	Spec. Std.
Logistic Regression	96.38	89.64	14.66	89.4	13.62	90	20
Decision Tree	100	86.79	13.72	90	13.52	80	24.49
Random Forest	100	83.93	15.07	89.4	13.62	70	24.49
Support Vector Classifier	96.38	89.64	14.66	89.4	13.62	90	20
Gaussian Naive Bayes	**98.79**	**92.14**	**10.2**	**93.4**	**13.2**	**90**	**20**
Multi-layer Perceptron	100	89.64	14.66	89.4	13.62	90	20
k-Nearest neighbors	100	89.64	14.66	89.4	13.62	90	20

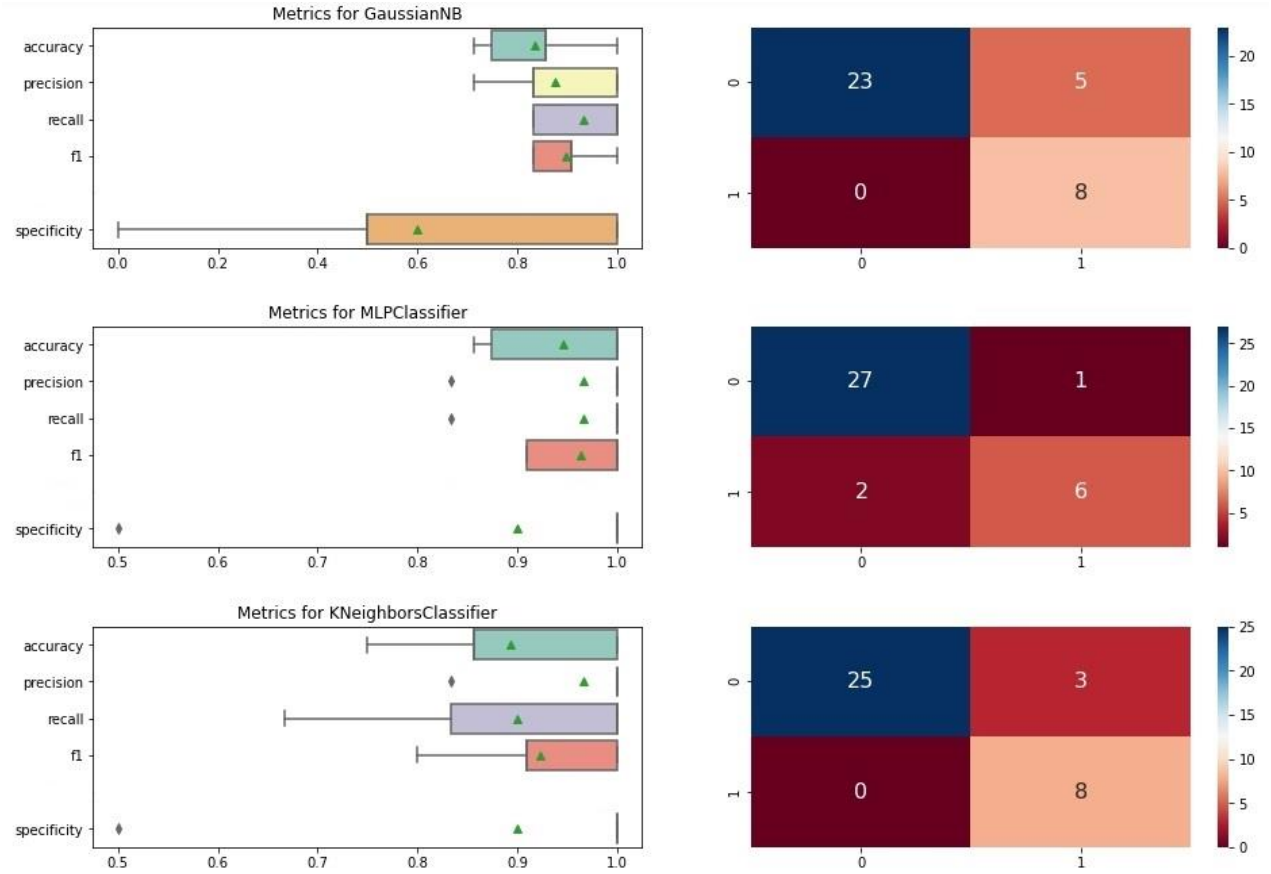
Πίνακας 10 Τιμές των Metrics για τα παραγωγικά μοντέλα για τα SelectKBest (Chi2) Selected Features

Όπως ακριβώς και με τα πειραματικά μοντέλα, έτσι και για τα αντίστοιχα που θεωρούμε παραγωγικά, θα εκτελέσουμε τη μέθοδο “generate_metrics” η οποία θα μας δώσει ως αποτέλεσμα την εκτύπωση των αντίστοιχων Boxplot, Confusion Matrix, Area Under Curve και Precision-Recall διαγραμμάτων.

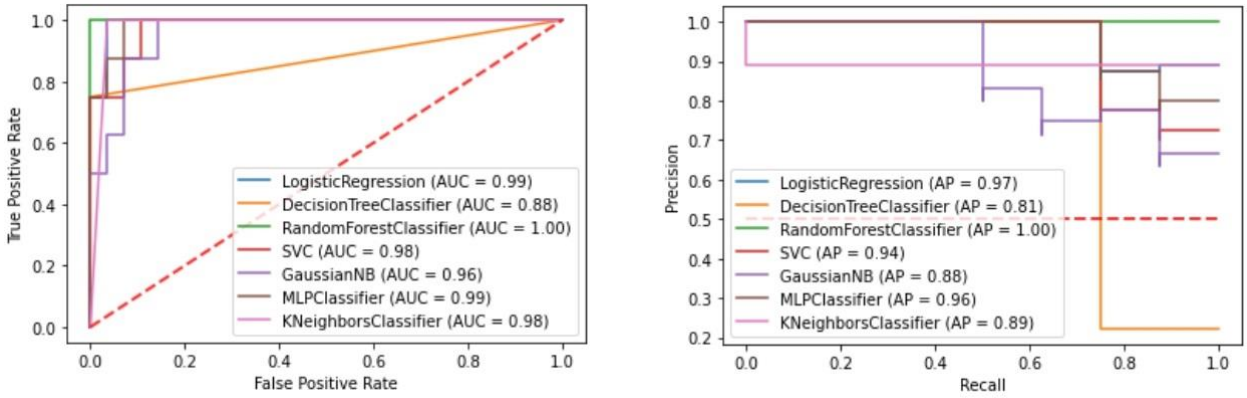
Επίσης και στη περίπτωση των παραγωγικών μοντέλων η μέθοδος “generate_metrics” έχει εκτελεστεί για κάθε ένα από τα δύο Feature Set που έχουμε δημιουργήσει, όπως αυτά περιγράφονται στην ενότητα §4.2.3.3.



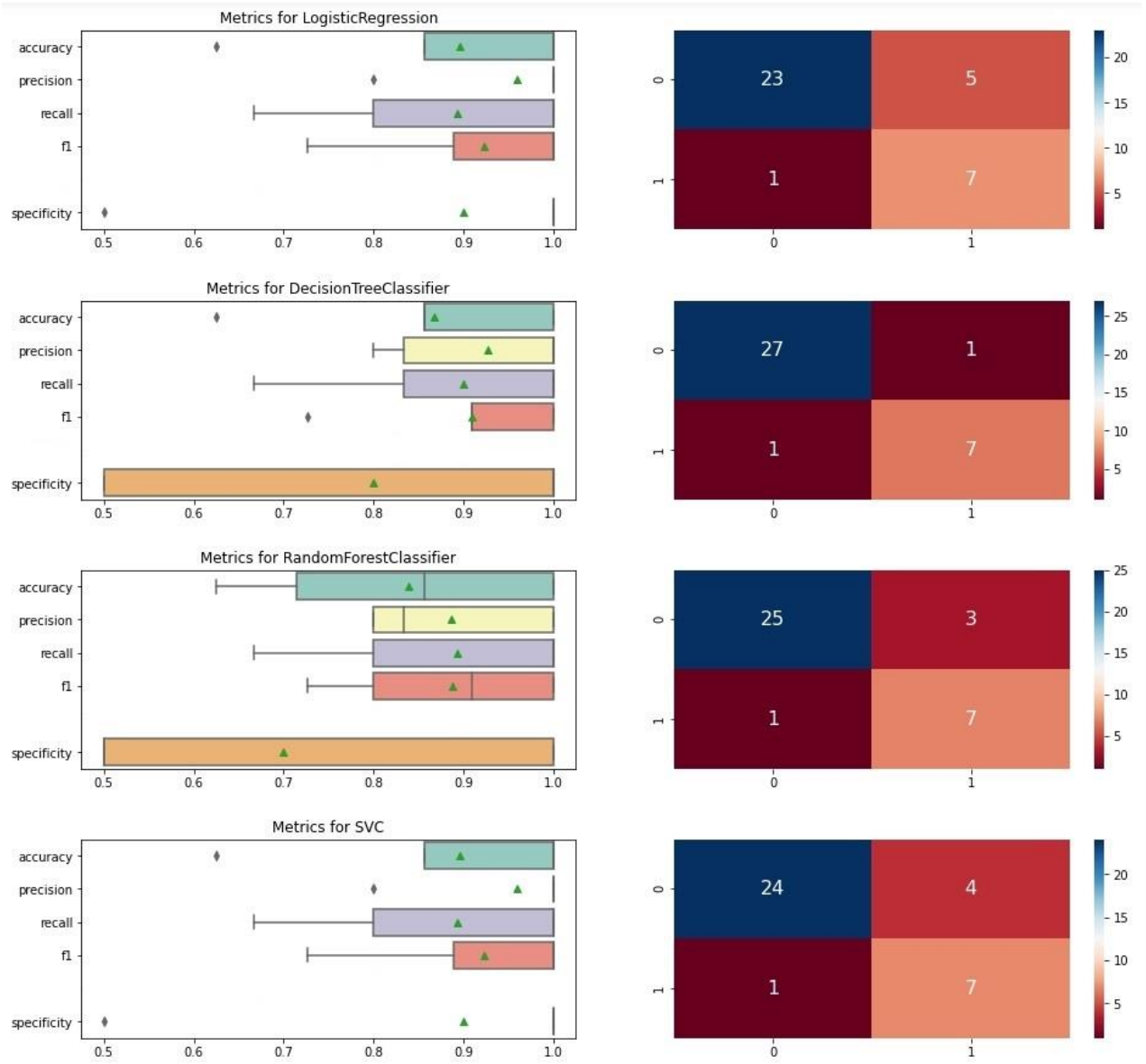
Εικόνα 58 Αριστερή στήλη: Διαγράμματα Boxplot για τα Metrics Accuracy, Precision, Recall, F-Score, Specificity, Δεξιά στήλη: Confusion Matrix για τα μοντέλα όπως προκύπτουν από τα Manually επιλεγμένα Features και μετά από την διαδικασία Production Model Creation, για τους αλγόριθμους Logistic Regression, Decision Tress, Random Forest, SVC



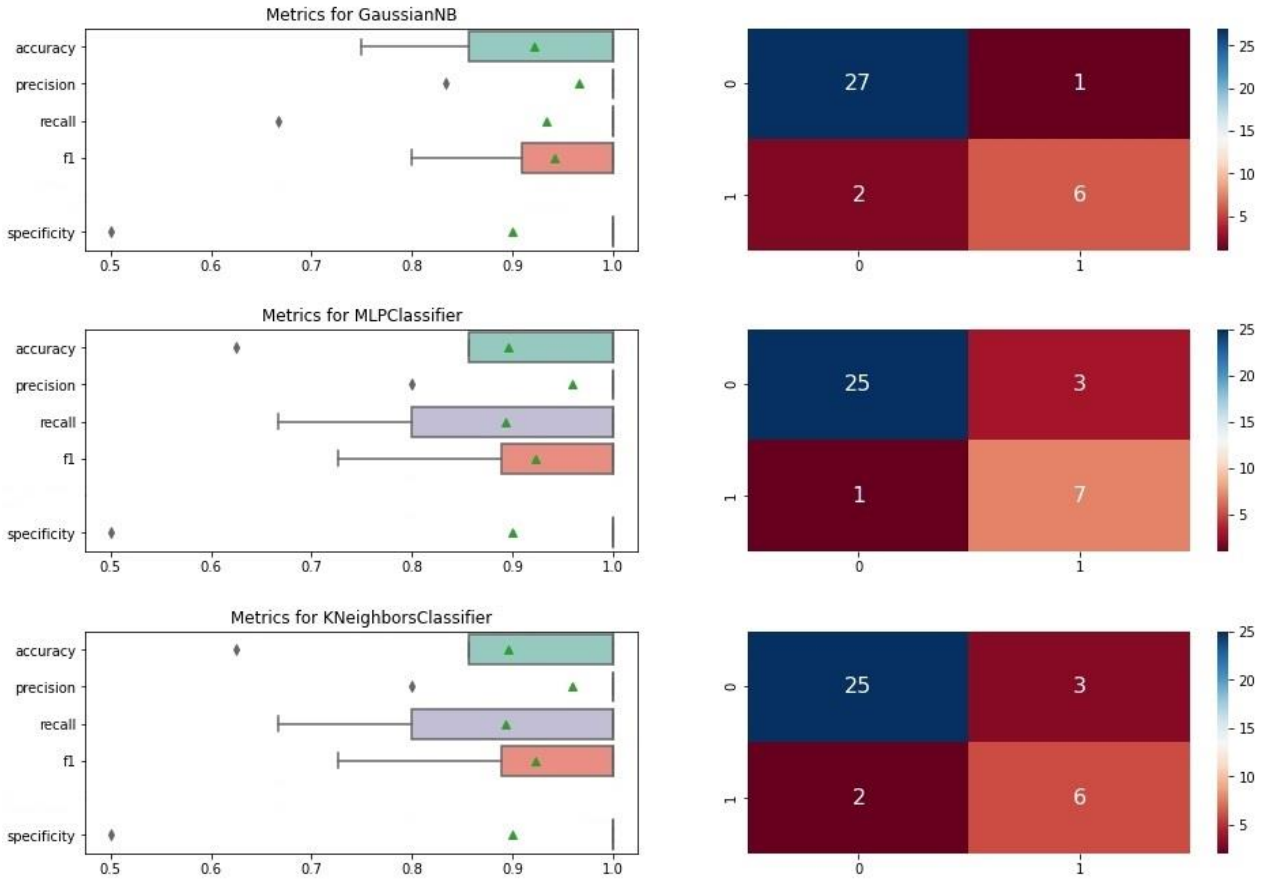
Εικόνα 59 Αριστερή στήλη: Διαγράμματα Boxplot για τα Metrics Accuracy, Precision, Recall, F-Score, Specificity, Δεξιά στήλη: Confusion Matrix για τα μοντέλα όπως προκύπτουν από τα Manually επιλεγμένα Features και μετά από την διαδικασία Production Model Creation, για τους αλγόριθμους Gaussian Naive Bayes, Multi-layer Perceptron, K Neighbors Classifier



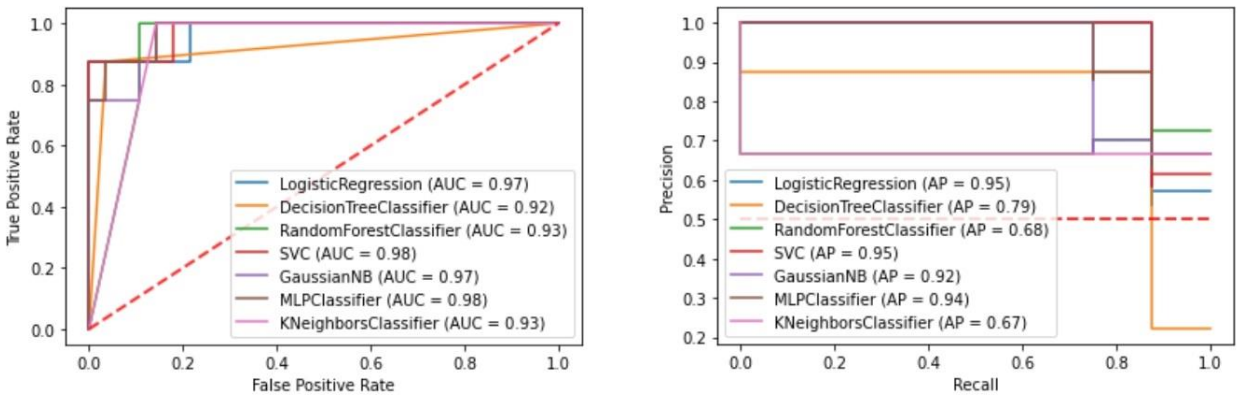
Εικόνα 60 Area Under Curve και Precision-Recall διαγράμματα, αριστερά και δεξιά αντίστοιχα, για όλα τα μοντέλα πλην του Custom Ensemble Classifier, εκπαιδευμένα με τα Manually επιλεγμένα Features μετά από τη διαδικασία Production Model Creation



Εικόνα 61 Αριστερή στήλη: Διαγράμματα Boxplot για τα Metrics Accuracy, Precision, Recall, F-Score, Specificity, Δεξιά στήλη: Confusion Matrix για τα Production μοντέλα όπως προκύπτουν από την εκπαίδευση με τους αλγόριθμους Logistic Regression, Decision Tree, Random Forest και SVC για τα Automatically επιλεγμένα Features μέσω SelectKBest (chi2) και κατόπιν των μεθόδων βελτιστοποίησης SMOTE, PCA και HPO



Εικόνα 62 Αριστερή στήλη: Διαγράμματα Boxplot για τα Metrics Accuracy, Precision, Recall, F-Score, Specificity, Δεξιά στήλη: Confusion Matrix για τα Production μοντέλα όπως προκύπτουν από την εκπαίδευση με τους αλγόριθμους Gaussian Naive Bayes, Multi-Layer Perceptron και K Nearest Neighbor για τα Automatically επιλεγμένα Features μέσω SelectKBest (chi2) και κατόπιν των μεθόδων βελτιστοποίησης SMOTE, PCA και HPO



Εικόνα 63 Area Under Curve και Precision-Recall διαγράμματα, αριστερά και δεξιά αντίστοιχα, για όλα τα παραγωγικά μοντέλα πλην του Custom Ensemble Classifier, εκπαιδευμένα με τα Automatically επιλεγμένα Features μέσω SelectKBest (chi2) και κατόπιν των μεθόδων βελτιστοποίησης SMOTE, PCA και HPO

6

Προκλήσεις

Σε αυτό το κεφάλαιο θα εξετάσουμε μερικές από τις σημαντικότερες προκλήσεις που εμφανίστηκαν κατά τη διάρκεια της εκπόνησης της διπλωματικής και αφορούν στο σύνολό τους την υλοποίηση.

6.1 Dataset

Όσον αφορά τις προκλήσεις γύρω από τα δεδομένα που υπήρχαν διαθέσιμα για την εκπαίδευση των μοντέλων, θα πρέπει να αναφερθούν οι παρακάτω περιορισμοί.

- Λόγω του σχετικά μικρού όγκου εγγραφών, υπάρχει ο περιορισμός στη εφαρμογή πιο σύνθετων αλγορίθμων μηχανικής μάθησης όπως είναι οι αλγόριθμοι τύπου Deep Learning. Το συγκεκριμένο πρόβλημα θα μπορούσε να αντιμετωπιστεί μόνο σε πιθανή επανάληψη της έρευνας. Ο πλέον σύνθετος αλγόριθμος που χρησιμοποιήθηκε, είναι ο Multi-layer Perceptron της βιβλιοθήκης Scikit-learn ο οποίος πρόκειται ουσιαστικά για ένα νευρωνικό δίκτυο.
- Ένα εξίσου σημαντικό ζήτημα αναφορικά με τα δεδομένα, έχει να κάνει με την άνιση κατανομή μεταξύ των δύο κατηγοριών για το Target Class που ορίσαμε. Συγκεκριμένα τα Game Sessions που αντιστοιχούν σε χρήστες της κατηγορίας MCI-AD είναι 94, σε αντίθεση με αυτά των χρηστών της κατηγορίας NC τα οποία είναι μόλις 25. Το συγκεκριμένο ζήτημα προκαλεί προβλήματα σε πολλά σημεία της διαδικασίας. Για παράδειγμα μας περιορίζει στον αριθμό k-fold που μπορούμε να χρησιμοποιήσουμε στις μεθόδους Cross Validation, ενώ ευνοεί τη δημιουργία Biased μοντέλων προς τη πλειοψηφική κατηγορία. Για την επίλυση του επιλέχθηκε η μέθοδος Oversampling με χρήση του αλγορίθμου SMOTE, όπως περιγράφεται και στην ενότητα του Optimization.
- Λόγω της ελευθερίας του χρήστη της εφαρμογής MCI Rehab στην επιλογή παιχνιδιών, επιπέδου και επαναλήψεων, τα δεδομένα μεταξύ των Game Sessions παρουσιάζουν μια σχετική ανομοιομορφία. Για παράδειγμα υπάρχουν Game Sessions στα οποίο ένας χρήστης δεν έχει παίξει ένα συγκεκριμένο παιχνίδι καθόλου, ή Game Sessions στα οποία ένας χρήστης έχει παίξει παιχνίδια σε όλα τα επίπεδα δυσκολίας ενώ σε άλλα Game Sessions μόνο σε ένα επίπεδο. Συμπερασματικά, τα δεδομένα πιθανώς να μην καλύπτουν το βαθμό ομοιομορφίας που χρειάζεται ένας αλγόριθμος μηχανικής μάθησης τύπου Time Series Analysis. Η συγκεκριμένη τεχνική, εάν

εφαρμόζονταν, θα μπορούσε να μας προσφέρει επιπλέον Features στο πλαίσιο του Feature Engineering.

- Επίσης όσον αφορά τα δεδομένα και συγκεκριμένα την ομάδα εστίασης, υπάρχουν χαρακτηριστικά με πολύ χαμηλό Variance όπως είναι το φύλο όπου έχουμε μόλις 2 γυναίκες προς 7 άντρες, ή ακόμη και μηδενικό όπως είναι το Feature της υπέρτασης και του καπνίσματος όπου υπάρχει η ίδια τιμή για όλα τα υποκείμενα. Τα Features αυτά, αποκλείστηκαν από την εκπαίδευση του μοντέλου, με τη διαδικασία Low Variance Feature Removal.

6.2 Data Leakage

Όσον αφορά τις προκλήσεις στο επίπεδο των διαδικασιών EDA και PMC, αυτές έχουν να κάνουν κυρίως με δύο ζητήματα, πρώτον το πρόβλημα που ονομάζεται Data Leakage και δεύτερον με την δυνατότητα ανάκτησης δεδομένων σε επιλεγμένα βήματα ενός Pipeline.

Με τον όρο Data Leakage αναφερόμαστε σε οποιαδήποτε μεταφορά πληροφορίας από το Testing προς το Training Dataset. Οι επιπτώσεις του Data Leakage είναι ουσιαστικά η πιθανή αλλοίωση των αποτελεσμάτων των Metrics καθώς τα δεδομένα του Testing Dataset εμπλέκονται στην διαδικασία της δημιουργίας (fit) των Transformers. Ο τρόπος αποφυγής αυτού του προβλήματος είναι η αντικατάσταση της μεθόδου fit_transform με τη χρήση της μεθόδου fit μόνο για το Training Dataset και η χρήση της μεθόδου transform στη συνέχεια και για τα δύο Dataset. Ωστόσο, αυτή η διαδικασία προσθέτει αρκετά μεγάλη πολυπλοκότητα καθώς θα έπρεπε να αποθηκεύουμε ξεχωριστά το κάθε Instance του κάθε Transformer για να το χρησιμοποιήσουμε στη δημιουργία του παραγωγικού μοντέλου. Η λύση σε αυτό το πρόβλημα είναι η χρήση των Pipelines, όπου πλέον όλοι οι μετασχηματισμοί γίνονται εντός μιας κλειστής διαδικασίας που δεν περιέχει κανένα στοιχείο του Testing Dataset.

Η παραπάνω λύση, η οποία έχει εφαρμοστεί για τη δημιουργία του παραγωγικού μοντέλου στην υλοποίηση της παρούσας διπλωματικής, πέρα από το πλεονέκτημα της αποφυγής του Data Leakage, έχει ένα μειονέκτημα, το οποίο είναι η δυσκολία στο να ανακτήσουμε τα δεδομένα από ένα συγκεκριμένο βήμα ενός Pipeline. Ο λόγος που θα θέλαμε τα δεδομένα μετά από συγκεκριμένους μετασχηματισμούς είναι για να προβάλουμε τις εκτυπώσεις όπως ακριβώς κάνουμε κατά τη διαδικασία του Preprocessing και του Classifier Selection. Η λύση που δόθηκε σε αυτό το ζήτημα είναι η μεθοδολογία που ακολουθήθηκε. Όπου πλέον κάνουμε δύο ξεχωριστά βήματα για να καταλήξουμε στο επιθυμητό αποτέλεσμα που είναι η δημιουργία του παραγωγικού μοντέλου. Πρώτα την αναλυτική δημιουργία όλων των διαδικασιών της EDA και έπειτα τη χρήση των δεδομένων, των Features, των αλγορίθμων και των παραμέτρων μέσω ενός Pipeline.

6.3 Optimization

Όσον αφορά τις προκλήσεις που παρουσιάστηκαν στις διαδικασίες του Optimization θα πρέπει να αναφερθούν τα εξής.

6.3.1 Oversampling και Pipelines

Για το κύριο όγκο της υλοποίησης χρησιμοποιήθηκε η βιβλιοθήκη Scikit-learn, ωστόσο για ένα από τα σημαντικότερα σημεία που είναι η δημιουργία ενός παραγωγικού μοντέλου, αντί της κλάσης Pipeline του Scikit-learn, έγινε χρήση της κλάσης Pipeline της βιβλιοθήκης imblearn. Αυτό διότι η πρώτη δεν υποστηρίζει

τη μέθοδο SMOTE, η οποία όμως κρίθηκε απαραίτητο να χρησιμοποιηθεί, γιατί όπως φαίνεται και από τα αποτελέσματα προσφέρει αρκετά στη βελτίωση των μοντέλων.

6.3.2 Dimensionality Reduction

Μια από τις τεχνικές Optimization που εξετάσαμε ήταν το Dimensionality Reduction. Πιο συγκεκριμένα εφαρμόσαμε τη τεχνική PCA καταφέρνοντας έτσι να μετασχηματίσουμε τις ανεξάρτητες μεταβλητές του Dataset, δηλαδή τα Features, σε δύο Principal Components, τα οποία περιείχαν ένα ποσοστό του αρχικού Variance. Η μέθοδος PCA, όπως αναλύεται και στο κεφάλαιο της μεθοδολογίας, είναι μια από τις πολλές τεχνικές για την υλοποίηση του Dimensionality Reduction.

Μια από τις εναλλακτικές μεθόδους είναι η Linear Discriminant Analysis (LDA), η οποία σε αντίθεση με την μέθοδο PCA, αποτελεί μια Supervised Learning τεχνική, καθώς για τη δημιουργία των νέων Components λαμβάνει υπόψη και το Target Class. Η πρόκληση που σχετίζεται με την μέθοδο LDA, είναι ότι ο αριθμός των νέων Components που προκύπτουν, σε αντίθεση με την τεχνική PCA, είναι συγκεκριμένος και για την ακρίβεια είναι πάντοτε η μικρότερη τιμή μεταξύ του αριθμού των Features και του αριθμό των κατηγοριών του Target Class.

Όσον αφορά τη διπλωματική, λόγω του ότι έχουμε επιλέξει ως Target Class την Binary έκφραση των αποτελεσμάτων της δοκιμασίας MOCA, έχουμε αυτομάτως τον περιορισμό του μέγιστου αριθμού Components που μπορούν να εξαχθούν με εφαρμογή της μεθόδου LDA, σε μόλις ένα Component. Συνεπώς στη προσπάθεια που έγινε για την εφαρμογή της μεθόδου LDA, συμπεριλαμβάνονταν η ομαδοποίηση των Features ώστε για κάθε ομάδα από Features να προκύπτει και ένα LDA Component. Ωστόσο αυτό σημαίνει πως στις περιπτώσεις όπου επιλέγαμε πάνω από 4 Features, μετά από την εφαρμογή της LDA για τα Features, θα έπρεπε να προχωρήσουμε σε επανάληψη της εφαρμογής της μεθόδου μεταξύ των Components, εάν υποθέσουμε πως θα έπρεπε να καταλήξουμε σε δύο Components. Λόγω αυτής της πολυπλοκότητας της μεθόδου η εφαρμογή της εγκαταλείφθηκε.

6.4 Evaluation Repeatability

Ένα από τα ζητήματα που παρουσιάστηκαν κατά τη διάρκεια των πειραμάτων ήταν η λήψη διαφορετικών τιμών μεταξύ των επαναλήψεων. Αυτό εν μέρη μπορεί να δικαιολογηθεί πρώτον, λόγω της στοχαστικής φύσης για κάποιους από τους αλγόριθμους και δεύτερον, λόγω της διαφοροποίησης των εγγραφών που περιέχονται στα Training και Testing Subsets αντίστοιχα ανά επανάληψη της διαδικασίας [37].

Όσον αφορά τον όποιο βαθμό Randomness περιέχεται στους αλγόριθμους μηχανικής μάθησης, αυτός αντιμετωπίστηκε ορίζοντας μια συγκεκριμένη τιμή για τη παράμετρο “random_state”. Η συγκεκριμένη παράμετρος αποτελεί ουσιαστικά ένα Seed το οποίο, όταν έχει οριστεί, αναγκάζει την όποια γεννήτρια ψευδοτυχαίων αριθμών χρησιμοποιεί ο αλγόριθμος, να επαναλάβει το ίδιο αποτέλεσμα σε κάθε επανάληψη. Όλοι οι αλγορίθμων που χρησιμοποιήθηκαν δέχονται ως παράμετρο ένα “random_state”, με δύο εξαιρέσεις. Πρώτη εξαίρεση αποτελεί ο αλγόριθμος KNeighborsClassifier του οποίου το Randomness καθορίζεται από το πλήθος των “n_neighbors” που θα ορίσουμε. Ενώ δεύτερη εξαίρεση αποτελεί ο αλγόριθμος GaussianNB ο οποίος ανήκει στη κατηγορία των Probabilistic Classifiers.

Αντίστοιχα το πρόβλημα του Randomness από τη σκοπιά των δεδομένων έχει να κάνει με τις διαφορετικές εγγραφές που μπορεί να έχουν τα επιμέρους Subsets σε κάθε επανάληψη του πειράματος. Αυτό το πρόβλημα αντιμετωπίστηκε ορίζοντας συγκεκριμένες τιμές για τις παρακάτω παραμέτρους για τη μέθοδο "train_test_split".

- Random State. Η παράμετρος "random_state", όταν οριστεί με μια συγκεκριμένη τιμή, διασφαλίζει ότι ο διαχωρισμός του Dataset σε Training και Testing παράγει πανομοιότυπα υποσύνολα σε κάθε επανάληψη.
- Training, Testing Ratio. Συμπληρωματικά, θα πρέπει να ορίσουμε τη παράμετρο "test_size" ώστε να καθορίσουμε έτσι το λόγο μεταξύ του πλήθους των εγγραφών του Training και του Testing Subset.
- Stratified Sampling. Επιπλέον, μπορούμε προαιρετικά να διασφαλίσουμε την ίση κατανομή μεταξύ των κατηγοριών του Target Class για κάθε Subset ορίζοντας τη παράμετρο "stratify" ως True.

7

Συμπεράσματα

Η παρούσα διπλωματική απέδειξε πως είναι εφικτή η δημιουργία μοντέλων μηχανικής μάθησης, βασισμένα σε δεδομένα που έχουν συλλεχθεί από παιχνίδια σοβαρού σκοπού, τα οποία μπορούν να αναγνωρίσουν με ικανοποιητική ακρίβεια εάν ένα άτομο ανήκει στη κατηγορία MCI-AD ή στη κατηγορία NC, συγκριτικά με την ακρίβεια που παρουσιάζει η νευροψυχολογική δοκιμασία MOCA, βάση της βιβλιογραφίας όπως καταγράφεται στην ενότητα §2.6.

Πέρα όμως από τα ποσοστά ακρίβειας που καταγράψαμε μέσω των Metrics και το ύψος των Bias και Variance του κάθε μοντέλου, το οποίο μπορούμε να συμπεράνουμε είτε από το ποσοστό του Accuracy κατά το Training, είτε από διάγραμμα Decision Boundary του κάθε μοντέλου, θα πρέπει να λάβουμε υπόψη μας και τα αντίστοιχα ποσοστά ακρίβειας που μας παρέχουν τα ήδη υπάρχοντα εργαλεία αναγνώρισης της MCI.

Αυτό σημαίνει, για παράδειγμα, πως εάν το ποσοστό ακρίβειας στην αναγνώριση της MCI, για τη νευροψυχολογική δοκιμασία MOCA, κυμαίνεται σε τιμές κοντά στο 99%, τότε ένα μοντέλο με ακρίβεια γύρω στο 90% θα έπρεπε να το κρίνουμε ανεπαρκές.

Τα ποσοστά που καταγράφονται στη βιβλιογραφία, για το Sensitivity κυμαίνονται μεταξύ 89% και 96%, ενώ για το Specificity κυμαίνονται μεταξύ 54% και 84%.

Ενώ τα ποσοστά των μοντέλων που έχουμε δημιουργήσει, συγκεκριμένα για δύο επιλεγμένα μοντέλα που πληρούν όλες τις προϋποθέσεις που θέσαμε για να χαρακτηρίσουμε ένα μοντέλο ιδανικό, είναι τα εξής.

Για το παραγωγικό μοντέλο που εκπαιδεύτηκε με τα Manually επιλεγμένα Features, τον αλγόριθμο Support Vector Classifier και όλες τις βελτιστοποιήσεις της HPO διαδικασίας, έχουμε για το Sensitivity **93.2% με 8,33% Standard Deviation**, ενώ για το Specificity έχουμε **90% με 20% Standard Deviation**.

Για το παραγωγικό μοντέλο που εκπαιδεύτηκε με τα Features που προκύπτουν μέσω SelectKBest και chi2 Scorer Function, τον αλγόριθμο Gaussian Naive Bayes και όλες τις βελτιστοποιήσεις της HPO διαδικασίας, έχουμε για το Sensitivity **93.4% με 13.2% Standard Deviation** και για το Specificity **90% με 20% Standard Deviation**.

Οι τιμές για το Metric Specificity παρουσιάζουν υψηλό Standard Deviation λόγω του ότι για τη δημιουργία του παραγωγικού μοντέλου, το Testing Dataset δεν υφίσταται τη διαδικασία Oversampling η

οποία είναι μέρος του Pipeline, οπότε οι True Negative τιμές είναι λίγες και κατά συνέπεια μικρά λάθη του μοντέλου καταγράφουν μεγάλη διακύμανση.

7.1 Προτάσεις

Σε αυτή την ενότητα παρατίθενται μερικές προτάσεις για βελτίωση της υλοποίησης και εξερεύνηση νέων ερευνητικών ερωτημάτων που πιθανώς να παρουσιάζουν ενδιαφέρον.

7.1.1 Έρευνα

Όσον αφορά στην έρευνα, οι προτάσεις είναι οι εξής.

- Επανάληψη της έρευνας από το σημείο της EDA διαδικασίας με στόχο αυτή τη φορά ένα μοντέλο το οποίο θα μπορεί να διακρίνει και στις επιμέρους κατηγορίες, έναν χρήστη, μεταξύ τριών κατηγοριών AD, MCI και NC.
- Επανάληψη της έρευνας με επίκεντρο άτομα με διαγνωσμένη MCI και στόχο την ομαδοποίηση τους ως προς τις υποκατηγορίες MCI. Δεδομένου ότι έχουν αναγνωριστεί πολλές υποκατηγορίες της MCI όπως είναι για παράδειγμα οι aMCI, Single Domain MCI, Multiple Domain MCI, Dysnomic MCI, Dysexecutive MCI, mx MCI και συνδυασμοί τους, οι οποίες συνδέονται άμεσα με τη κατάσταση της γνωστικής επάρκειας του ατόμου σε συγκεκριμένες γνωστικές λειτουργίες [5], θα είχε ενδιαφέρον ένα ερευνητικό ερώτημα όπου θα εξετάζεται η συσχέτιση των χαμηλών επιδόσεων σε ένα συγκεκριμένα παιχνίδια με συγκεκριμένες υποκατηγορίες της MCI.
- Η πραγματοποίηση μιας αντίστοιχης έρευνας, σε μια ομάδα εστίασης με πιο σοβαρά συμπτώματα νευροεκφυλιστικών παθήσεων ή ακόμη και σε άτομα με διαγνωσμένη άνοια. Οι νευροψυχολογικές δοκιμασίες MMSE και MOCA θα μπορούσαν να αντικατασταθούν με αντίστοιχες που χρησιμοποιούνται συνήθως για αυτές τις περιπτώσεις όπως είναι η δοκιμασία “The Cognitive Abilities Screening Instrument” (CASI) και η δοκιμασία Hasegawa's Dementia Scale [31].

7.1.2 Τεχνικές προδιαγραφές

Όσον αφορά βελτιώσεις που θα μπορούσαν να γίνουν σε τεχνικό επίπεδο, προτείνονται τα εξής.

- Μια προσθήκη που μπορούσε πιθανώς να φανεί χρήσιμη στον τελικό χρήστη, θα ήταν η χρήση αλγορίθμων μηχανικής μάθησης που είναι ικανοί να παράγουν μοντέλα τύπου Explainable Models. Τα συγκεκριμένα μοντέλα πέραν της κατηγοριοποίησης και τη επιστροφής ενός ποσοστού βεβαιότητας (Confidence), για όσους αλγόριθμους το υποστηρίζουν, μπορούν να επιστρέφουν και τον λόγο για τον οποίο μια είσοδος δεδομένων χαρακτηρίστηκε με μια συγκεκριμένη κατηγορία. Συνεπώς έτσι θα μπορούσε ο χρήστης να πληροφορηθεί για το ποια από τα Features είναι που καθόρισαν την κατηγορία του και ποιες ήταν συγκεκριμένα οι τιμές τους. Επιπλέον, θα μπορούσε η εφαρμογή να θέτει στόχους στον χρήστη για τη βελτίωση των συγκεκριμένων τιμών.

7.1.3 Δεδομένα

Όσο για πιθανές βελτιώσεις που θα μπορούσαν να γίνουν αναφορικά με τα δεδομένα που θα συλλεχθούν σε μια μελλοντική επανάληψη της έρευνας, οι προτάσεις είναι οι εξής.

- Επιλογή ατόμων για την ομάδα εστίασης με πιο ευρύ φάσμα όσον αφορά τις βαθμολογίες στις νευροψυχολογικές δοκιμασίες.
- Δημιουργία μιας επιπλέον λειτουργίας της εφαρμογής η οποία θα στοχεύει αποκλειστικά στην αξιολόγηση, για παράδειγμα ένα Assessment Mode. Κατά τη διάρκεια μιας τέτοιας λειτουργίας, η εμπειρία του χρήστη θα ήταν προκαθορισμένη σε αντίθεση με τη κανονική λειτουργία όπου θα μπορεί να εξασκηθεί σε όποια δοκιμασία επιθυμεί. Ο λόγος είναι για να υπάρχει μια κανονικοποίηση όσον αφορά τα δεδομένα που θα συλλέγονται για αξιολόγηση, για παράδειγμα ίσος αριθμός Game Round, σε προκαθορισμένα επίπεδα δυσκολίας, για κάθε παιχνίδι σε ένα Game Session, για όλους τους χρήστες. Η εφαρμογή τέτοιων περιορισμών θα έκανε τα δεδομένα μεταξύ των χρηστών πιο εύκολα συγκρίσιμα και πιθανώς θα μας έδινε τη δυνατότητα για εφαρμογή αλγορίθμων τύπου Time Series Analysis.

Βιβλιογραφία

1. Γεώργιος Σκίκος (2019), Σχεδίαση και υλοποίηση παιχνιδιών σοβαρού σκοπού για την υποστήριξη ασθενών με ήπια γνωστική εξασθένηση
2. O L Lopez (2005), Neuropsychological characteristics of mild cognitive impairment subgroups, σελ.1
3. Usama Fayyad (1996), From Data Mining to Knowledge Discovery in Databases, σελ.4
4. Jeffrey Saltz (2018), Exploring Project Management Methodologies Used Within Data Science Teams
5. María del Carmen Díaz-Mardomingo (2017), Problems in Classifying Mild Cognitive Impairment (MCI): One or Multiple Syndromes?
6. Usama Fayyad (1996), The KDD process for extracting useful knowledge from volumes of data, σελ.29
7. Mc Leod RD (2019), A Framework for Utilizing Serious Games and Machine Learning to Classifying Game Play Towards Detecting Cognitive Impairments
8. Kyle Leduc-McNiven (2018), Serious games to assess mild cognitive impairment: 'The game is the assessment'
9. V. Joshi (2016), Metrics to Monitor Performance of Patients with Mild Cognitive Impairment using Computer Based Games
10. Russell Binaco (2019), Machine Learning Analysis of Digital Clock Drawing Test Performance for Differential Classification of Mild Cognitive Impairment Subtypes Versus Alzheimer's Disease, σελ.6
11. Rok Blagus (2013), SMOTE for high-dimensional class-imbalanced data, σελ.2
12. J.S. Roberts (2010), Mild cognitive impairment in clinical care: A survey of American Academy of Neurology members
13. Fadi Massoud (2010), Update on the Pharmacological Treatment of Alzheimers Disease
14. John E. Morley (2004), A Brief History of Geriatrics
15. A. Bifet (2011), Handling Concept Drift: Importance, Challenges & Solutions
16. Kamini Krishnan (2016), Changes in Montreal Cognitive Assessment Scores Over Time
17. World Health Organization and Alzheimer's Disease International (2012), Dementia: a public health priority
18. Ziad S. Nasreddine (2005), The Montreal Cognitive Assessment, MoCA: A Brief Screening Tool For Mild Cognitive Impairment
19. Υπουργείο Υγείας Διεύθυνση Ψυχικής Υγείας (2014), Κλινικές Κατευθυντήριες Οδηγίες (Κ.Ο.) Κ.Ο.-1: Άνοια. Προτάσεις Για Τη Διάγνωση Και Αντιμετώπιση Της Νόσου Alzheimer Και Άλλων Διαταραχών Σχετιζόμενων Με Άνοια
20. University of Iowa (2012), MMSE for Dementia
21. IHPA (2014), Standardised Mini-Mental State Examination (SMMSE)

22. Michael Galarnyk (2018), https://towardsdatascience.com/understanding-the-68-95-99-7-rule-for-a-normal-distribution-b7b7cbf760c2
23. Pedregosa et al. (2011), JMLR 12, Scikit-learn: Machine Learning in Python, pp. 2825-2830.
24. Jason Brownlee (2020), https://machinelearningmastery.com/gentle-introduction-concept-drift-machine-learning/
25. "What is "unit" standard deviation?", https://stats.stackexchange.com/questions/305672/what-is-unit-standard-deviation
26. Daniel Shiffman (2012), The Nature of Code, https://github.com/nature-of-code/NOC-S17-2-Intelligence-Learning/wiki/Glossary:-Statistics
27. Fatemeh Nargesian (2017), Learning Feature Engineering for Classification, σελ.1
28. Lei Tang (2005), Bias Analysis in Text Classification for Highly Skewed Data σελ.2
29. Giles Hooker (2019), Please Stop Permuting Features An Explanation and Alternatives
30. Deborah E. Barnes (2020), Development and Validation of eRADAR: A Tool Using HER Data to Detect Unrecognized Dementia
31. Evelyn L. Teng (2018), The Cognitive Abilities Screening Instrument (CASI): a practical test for cross-cultural epidemiological studies of dementia
32. Afshine Amidi (2018), VIP Cheatsheet: Machine Learning Tips, Stanford CS229-Machine Learning
33. Suzanne Ekelund (2017), Precision-recall curves – what are they and how are they used?
34. U.Roessner (2011), Comprehensive Biotechnology (Third Edition) Volume 1, Κεφάλαιο 1.31.7.2.1 Principal Component Analysis
35. Andrew L. Beam (2017), Big Data and Machine Learning in Health Care
36. Griffin M. Weber (2014), Finding the Missing Link for Big Biomedical Data
37. Jason Brownlee (2020), https://machinelearningmastery.com/different-results-each-time-in-machine-learning/
38. Cody Sider (2017), Comparative Efficacy of the Montreal Cognitive Assessment (MOCA) and the Rowland Universal Dementia Assessment Scale (RUDAS) as Brief Screening Tools for Cognitive Impairment and Dementia, CGS 37th Annual Scientific Meeting Integrating Care, Making an Impact
39. Felicia C. Goldstein (2015), Validity of the Montreal Cognitive Assessment as a Screen for Mild Cognitive Impairment and Dementia in African Americans
40. Ron´ an O'Caoimh (2016), Screening for Mild Cognitive Impairment: Comparison of MCI Specific Screening Instruments
41. Jun-Young Lee (2015), Brief Screening for Mild Cognitive Impairment in Elderly Outpatient Clinic: Validation of the Korean Version of the Montreal Cognitive Assessment
42. Tiffany Tong (2016), A Serious Game for Clinical Assessment of Cognitive Status: Validation Study, Table 1, σελ.2
43. Sonia Valladares-Rodriguez (2018), Learning to Detect Cognitive Impairment through Digital Games and Machine Learning Techniques*, σελ.6
44. P. Chapman (2000), CRISP-DM 1.0: Step-by-step data mining guide
45. Team Data Science Process Documentation (2020), https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/overview
46. Jason Brownlee (2020), Plot a Decision Surface for Machine Learning Algorithms in Python, https://machinelearningmastery.com/plot-a-decision-surface-for-machine-learning/