

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



- Πτυχιακή Εργασία -

«Μηχανική Μάθηση για την Πρόγνωση Εγκεφαλικού Επεισοδίου»

ΓΕΩΡΓΙΟΣ – ΦΙΛΙΠΠΟΣ ΜΠΙΖΑΚΗΣ, Π2019135

ΕΠΙΒΛΕΠΩΝ: ΘΕΜΙΣΤΟΚΛΗΣ ΕΞΑΡΧΟΣ,
ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ

ΙΟΥΝΙΟΣ 2024

Τριμελής Εξεταστική Επιτροπή:

Πρώτος Εξεταστής
(επιβλέπων)

Θεμιστοκλής Έξαρχος
(Αναπληρωτής Καθηγητής)

Δεύτερος Εξεταστής

Παναγιώτης Βλάμος
(Καθηγητής)

Τρίτος Εξεταστής

Αριστείδης Βραχάτης
(Επίκουρος Καθηγητής)

Πίνακας Περιεχομένων

1	Εισαγωγή.....	1
2	Πληροφορίες για τα εγκεφαλικά επεισόδια.....	2
2.1	Τι είναι το εγκεφαλικό επεισόδιο ;	2
2.2	Παραδοσιακοί τρόποι διάγνωσης του εγκεφαλικού	2
2.3	Ο ρόλος της μηχανικής μάθησης στην ιατρική.....	4
2.4	Ανασκόπηση άρθρων πρόβλεψης εγκεφαλικών με μηχανική μάθηση	5
3	Μεθοδολογία.....	12
3.1	Δεδομένα.....	12
3.1.1	Καθαρισμός Δεδομένων.....	13
3.1.2	Σημαντικότητα Χαρακτηριστικών	16
3.1.3	Κωδικοποίηση Χαρακτηριστικών	16
3.1.4	Ακραίες Τιμές	18
3.2	Cross Validation.....	21
4	Αλγόριθμοι και Εκπαίδευση Μοντέλων	21
4.1	Διαχωρισμός Δεδομένων	21
4.2	Αξιολόγηση Μοντέλων και Μετρικές Απόδοσης	22
4.2.1	Πίνακας Σύγκρισης	22
4.2.2	Μετρήσεις Απόδοσης.....	23
4.3	Logistic Regression	24
4.4	Decision Tree	26
4.5	Random Forest.....	28
4.6	Boosting	30
4.6.1	XGBoost	31
4.6.2	RUSBoost	34
4.6.3	AdaBoost	36
5	Αποτελέσματα	38
5.1	Σύγκριση Απόδοσης των Μοντέλων	38
6	Ανάπτυξη Εφαρμογής	43
7	Συμπεράσματα	47
8	Βιβλιογραφία	65

Λίστα Πινάκων

Πίνακας 1: Χαρακτηριστικά δεδομένων.....	12
Πίνακας 2: Αποτελέσματα μετρήσεων του Logistic Regression.....	25
Πίνακας 3: Αποτελέσματα μετρήσεων του αλγορίθμου Decision Tree	28
Πίνακας 4: Αποτελέσματα μετρήσεων του αλγορίθμου Random Forest	30
Πίνακας 5: Αποτελέσματα μετρήσεων του αλγορίθμου XGBoost.....	33
Πίνακας 6: Αποτελέσματα μετρήσεων του αλγορίθμου RUSBoost.....	36
Πίνακας 7: Αποτελέσματα μετρήσεων του αλγορίθμου ADABOOST.....	38
Πίνακας 8: Αποτελέσματα μετρήσεων όλων των αλγορίθμων.....	38

Λίστα Εικόνων

Εικόνα 1: Ακρωνύμιο πρόληψης εγκεφαλικού	3
Εικόνα 2: Τα δεδομένα πριν την προεπεξεργασία	13
Εικόνα 3: Διαχωρισμός των χαρακτηριστικών από την έξοδο	13
Εικόνα 4: Εφαρμογή Random Undersampling στα δεδομένα.....	14
Εικόνα 5: Τα δεδομένα μετά την προεπεξεργασία.....	14
Εικόνα 6: Οι ελλείπουσες τιμές των δεδομένων.....	15
Εικόνα 7: Σημαντικότητα Χαρακτηριστικών (Feature Importance).....	16
Εικόνα 8: Ιστόγραμμα του χαρακτηριστικού 'bmi' πριν τη διαγραφή	18
Εικόνα 9: Boxplot του χαρακτηριστικού 'bmi' πριν τη διαγραφή.....	19
Εικόνα 10: Ιστόγραμμα του χαρακτηριστικού 'bmi' μετά τη διαγραφή.....	20
Εικόνα 11: Boxplot του χαρακτηριστικού 'bmi' μετά τη διαγραφή	20
Εικόνα 12: Παράμετροι του αλγορίθμου Logistic Regression	25
Εικόνα 13: Δομή του αλγορίθμου Δένδρων Απόφασης (Decision Tree).....	26
Εικόνα 14: Παράμετροι του αλγορίθμου Decision Tree.....	27
Εικόνα 15: Γραφική αναπαράσταση του αλγορίθμου Decision Tree	27
Εικόνα 16: Παράμετροι του αλγορίθμου Τυχαίου Δάσους (Random Forest)	29
Εικόνα 17: Γραφική αναπαράσταση του αλγορίθμου Random Forest.....	30
Εικόνα 18: Παράμετροι του αλγορίθμου XGBoost.....	33
Εικόνα 19: Παράμετροι του αλγορίθμου RUSBoost.....	35
Εικόνα 20: Παράμετροι του αλγορίθμου ADABOOST.....	38
Εικόνα 21: Πίνακας σύγκρισης του αλγορίθμου ADABOOST.....	40
Εικόνα 22: Πίνακας σύγκρισης του αλγορίθμου XGBoost	40
Εικόνα 23: Πίνακας σύγκρισης του αλγορίθμου Random Forest	41
Εικόνα 24: Πίνακας σύγκρισης του αλγορίθμου Decision Tree	41
Εικόνα 25: Πίνακας σύγκρισης του αλγορίθμου Logistic Regression.....	42
Εικόνα 26: Πίνακας σύγκρισης του αλγορίθμου RUSBoost	42
Εικόνα 27: Η εφαρμογή πρόβλεψης εγκεφαλικού επεισοδίου	43
Εικόνα 28: Επιλογή φύλλου.....	43
Εικόνα 29: Εισαγωγή ηλικίας	43
Εικόνα 30: Εισαγωγή μέσου επιπέδου σακχάρου.....	43
Εικόνα 31: Επιλογή καρδιακών παθήσεων.....	44
Εικόνα 32: Επιλογή κατάστασης της πίεσης	44
Εικόνα 33: Εισαγωγή Δείκτη Μάζας Σώματος.....	44

Εικόνα 34: Επιλογή τύπου δουλειάς.....	44
Εικόνα 35: Επιλογή κατάστασης καπνίσματος.....	44
Εικόνα 36: Επιλογή περιοχής κατοικίας.....	45
Εικόνα 37: Επιλογή αν ο χρήστης έχει παντρευτεί ποτέ	45
Εικόνα 38: Κείμενο αποποίησης ευθύνης.....	45
Εικόνα 39: Εμφάνιση αποτελέσματος πρόβλεψης με τη μορφή "pop-up" (χωρίς κίνδυνο εγκεφαλικού).....	46
Εικόνα 40: Εμφάνιση αποτελέσματος πρόβλεψης με τη μορφή "pop-up" (με κίνδυνο εγκεφαλικού).....	46

Περίληψη

Το εγκεφαλικό επεισόδιο είναι μία από τις κύριες αιτίες θανάτου και αναπηρίας παγκοσμίως, με σημαντικές επιπτώσεις στους ανθρώπους. Οι παραδοσιακές μέθοδοι διάγνωσης συχνά αποτυγχάνουν να λάβουν υπόψη όλους τους παράγοντες κινδύνου, καθιστώντας αναγκαία την ανάπτυξη ακριβέστερων εργαλείων πρόβλεψης. Η παρούσα πτυχιακή εργασία εξετάζει την εφαρμογή αλγορίθμων μηχανικής μάθησης για τη βελτίωση της πρόβλεψης εγκεφαλικών επεισοδίων, χρησιμοποιώντας δεδομένα ασθενών όπως η ηλικία, ο Δείκτης Μάζας Σώματος (bmi) και η υπέρταση. Μέσω της ανάλυσης αυτών των συνόλων δεδομένων και της δοκιμής διαφόρων μοντέλων, η έρευνα στοχεύει στην ανάπτυξη κατανοητών και ερμηνεύσιμων μοντέλων πρόβλεψης. Τα αποτελέσματα δείχνουν ότι η μηχανική μάθηση μπορεί να βελτιώσει σημαντικά την ακρίβεια των προβλέψεων εγκεφαλικού επεισοδίου, παρέχοντας πολύτιμη υποστήριξη για έγκαιρη παρέμβαση και εξατομικευμένη φροντίδα των ασθενών.

Λέξεις κλειδιά: μηχανική μάθηση, πρόβλεψη εγκεφαλικού επεισοδίου, ιατρικά δεδομένα, επιδόσεις αλγορίθμων.

Abstract

Stroke is one of the leading causes of death and disability worldwide, with a significant impact on humans. Traditional diagnostic methods often fail to account for all risk factors and the development of more accurate predictive tools is essential. This thesis examines the application of various machine learning algorithms to improve stroke prediction, using patient data such as age, body mass index (BMI) and hypertension. By analyzing these datasets and testing various models, the research aims to develop understandable and interpretable forecasting models. The results show that machine learning can significantly improve the accuracy of stroke predictions, providing valuable support for early intervention and personalized patient care.

Key words: machine learning, stroke prediction, medical data, algorithm performance

1 Εισαγωγή

Τα εγκεφαλικά επεισόδια είναι μια από τις πιο σοβαρές αιτίες θανάτου παγκοσμίως. Σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας (ΠΟΥ), το εγκεφαλικό είναι η δεύτερη κατά σειρά αιτία θανάτου παγκοσμίως και πρώτη αιτία αναπηρίας στους ενήλικες, με τα εκτιμώμενα περιστατικά να αγγίζουν τα 15 εκατομμύρια κάθε χρόνο εκ των οποίων 5 εκατομμύρια άνθρωποι χάνουν τη ζωή τους και άλλοι 5 εκατομμύρια αποκτούν μόνιμες αναπηρίες [1]. Στην Ευρώπη, σύμφωνα με την Stroke Alliance for Europe (SAFE), περισσότεροι από 750.000 άνθρωποι παθαίνουν εγκεφαλικό ετησίως και ο αριθμός τους τείνει να αυξάνεται κατά 35% την περίοδο 2017 με 2035 [2]. Υπάρχει μεγάλη πιθανότητα να προκαλέσουν μακροχρόνιες αναπηρίες, όπως προβλήματα στην όραση, στην επικοινωνία, και ψυχολογικά προβλήματα, έως και θάνατο [3]. Η πρόληψη είναι ένας από τους πιο σημαντικούς τρόπους αντιμετώπισης των εγκεφαλικών μαζί με τη σωστή πρόβλεψη από εξειδικευμένους ιατρούς.

Κάποιοι από τους πιο βασικούς παράγοντες που χρησιμοποιούνται για την διάγνωση ενός ασθενή για εγκεφαλικό επεισόδιο είναι η ηλικία, το κάπνισμα και η υπέρταση. Παρόλο που αυτές οι ενδείξεις δίνουν αρκετές πληροφορίες στην διάγνωση, πολλές φορές δεν καλύπτουν όλους τους ασθενείς και δεν λαμβάνονται άλλοι παράγοντες υπόψιν κατά τη διαδικασία αυτή.

Η πρόοδος της τεχνολογίας και πιο συγκεκριμένα της μηχανικής μάθησης στην ιατρική, έχει δώσει μια καινούργια πνοή στο κλάδο της ιατρικής με εργαλεία τα οποία βοηθούν τόσο στη δημιουργία μοντέλων πρόβλεψης όσο και στην υποστήριξη της λήψης απόφασης των ιατρών. Οι αλγόριθμοι μηχανικής μάθησης μπορούν να δέχονται τεράστια dataset μέσα από τα οποία βρίσκουν μοτίβα, συσχετίσεις και εξατομικευμένους κινδύνους ξεχωριστά για κάθε ασθενή.

Ο στόχος αυτής της πτυχιακής είναι η εφαρμογή των μεθοδολογιών και αλγορίθμων μηχανικής μάθησης στην πρόβλεψη εγκεφαλικών επεισοδίων με απώτερο σκοπό τη δημιουργία μοντέλων πρόβλεψης τα οποία προσφέρουν σιγουριά και ακρίβεια στον ασθενή. Θα γίνει ανάλυση του dataset, των χαρακτηριστικών των ασθενών (ηλικία, bmi, υπέρταση) και δοκιμή διαφόρων αλγορίθμων. Μέσα από τις διάφορες δοκιμές, θα γίνει αναζήτηση των βέλτιστων μοντέλων που θα μπορούν να παρέχουν ακριβείς και ερμηνεύσιμες προβλέψεις για τον κίνδυνο εγκεφαλικού επεισοδίου.

Στα παρακάτω κεφάλαια, θα εμβαθύνουμε στις θεωρητικές βάσεις για την πρόβλεψη εγκεφαλικού επεισοδίου, θα γίνει βιβλιογραφική ανασκόπηση σχετικά με τη μηχανική μάθηση στη πρόβλεψη εγκεφαλικού επεισοδίου, θα γίνει περιγραφή της μεθοδολογίας για τη συλλογή και ανάλυση των δεδομένων, παρουσίαση των αποτελεσμάτων και τέλος, θα γίνει συζήτηση των αποτελεσμάτων στην πράξη και τις μελλοντικές έρευνες.

2 Πληροφορίες για τα εγκεφαλικά επεισόδια

2.1 Τι είναι το εγκεφαλικό επεισόδιο ;

Το εγκεφαλικό επεισόδιο, ή αλλιώς εγκεφαλική προσβολή, συμβαίνει όταν μια απόφραξη, π.χ. ένας θρόμβος, εμποδίζει τη φυσιολογική ροή του αίματος στον εγκέφαλο ή όταν ένα σπασμένο αιμοφόρο αγγείο αιμορραγεί στον εγκέφαλο. Υπάρχουν δύο τύποι εγκεφαλικών επεισοδίων: το Αγγειακό Αιμορραγικό Επεισόδιο και το Αγγειακό Ισχαιμικό Επεισόδιο. Το Αγγειακό Αιμορραγικό Επεισόδιο, υπάρχει στον εγκέφαλο αίμα μέσα ή γύρω του. Για το Αγγειακό Ισχαιμικό Επεισόδιο, τα αγγεία εμποδίζουν το αίμα να εισέλθει στον εγκέφαλο οπότε αυτά τα τμήματα στα οποία δεν μπορεί το αίμα να φτάσει βλάπτονται.

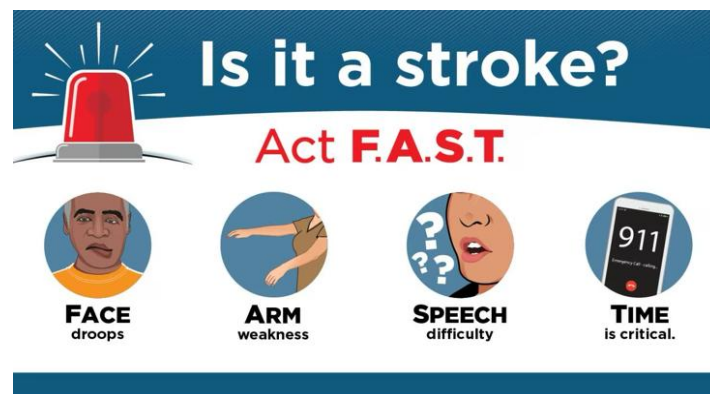
2.2 Παραδοσιακοί τρόποι διάγνωσης του εγκεφαλικού

Κάποιες ενδείξεις που μπορούν να βοηθήσουν στον εντοπισμό της εκδήλωσης του εγκεφαλικού είναι:

1. **Παράλυση του προσώπου:** Η πιο συχνή του εκδήλωση γίνεται με το «πέσιμο» του ενός μισού του προσώπου. Ο ασθενής μπορεί να έχει περιορισμένη κίνηση των μυών του προσώπου του οπότε η μια πλευρά να φαίνεται χαμηλότερη από την άλλη.
2. **Αδυναμία στα χέρια:** Κάποιοι από τους ασθενείς μπορεί να βιώσουν μούδιασμα ή αδυναμία στο ένα ή και στα δυο χεριά. Ένας απλός έλεγχος είναι να σηκώσει και τα δυο χεριά παράλληλα· αν το ένα χέρι πέσει ή δεν μπορεί να το κρατήσει ψηλά, υπάρχει πιθανότητα να αποτελεί μια ένδειξη εγκεφαλικού επεισοδίου.

3. **Δυσκολία στην ομιλία:** Συνήθως παρατηρείται με σύγχυση, δυσκολία στην κατανόηση του λόγου ή άρθρωση προτάσεων/ λέξεων. Ο ασθενής μπορεί να μην έχει την δυνατότητα να εκφραστεί ξεκάθαρα ή να μπερδεύει τις λέξεις που σκέφτεται ότι θέλει να πει.
4. **Ίλιγγος:** Ο ξαφνικός και έντονος ίλιγγος μπορεί να είναι μια ακόμα ένδειξη εγκεφαλικού, κυρίως όταν συνοδεύεται από άλλα συμπτώματα όπως απώλεια ισορροπίας ή δυσκολία στο περπάτημα.
5. **Θολή όραση:** Δυσκολία στην όραση σε ένα ή και στα δύο μάτια. Η θολή όραση μπορεί να είναι προσωρινή ή μόνιμη και να αποτελεί πιθανό σύμπτωμα εγκεφαλικού.

Ένα ακρωνύμιο που χρησιμοποιείται για την άμεση και εύκολη διάγνωση είναι το F.A.S.T [4]



Εικόνα 1: Ακρωνύμιο πρόληψης εγκεφαλικού

Τα παραπάνω χρησιμοποιούνται ως μια πρώτη και γρήγορη διάγνωση από ένα τρίτο άτομο και δεν χρειάζονται ιδιαίτερες γνώσεις ιατρικής. Βέβαια, η διάγνωση ενός εγκεφαλικού επεισοδίου από ένα νευρολόγο ή κάποιον εξειδικευμένο γιατρό είναι απαραίτητη και περιλαμβάνει [5]:

- **Φυσική Εξέταση:** Βασική κλινική αξιολόγηση του ασθενούς
- **Νευρολογική Εξέταση:** Ειδική εξέταση για την αξιολόγηση της νευρολογικής λειτουργίας
- **Αιματολογικές Εξετάσεις:** Ανάλυση του αίματος για βιοχημικούς και άλλους δείκτες υγείας
- **Εξετάσεις απεικόνισης του εγκεφάλου:** MPI ή CT για τον διεξοδικό έλεγχο της δομής του εγκεφάλου

- **Απεικόνιση αγγείων εγκεφάλου:** Αγγειογραφία για την αξιολόγηση της υγείας των αγγείων και τον έλεγχο ανωμαλιών
- **Δοκιμασίες καρδιακής λειτουργίας:** Διάφορες καρδιολογικές εξετάσεις για τον εντοπισμό καρδιακών παθήσεων που μπορεί να συσχετίζονται με το εγκεφαλικό
- **Ηλεκτροκαρδιογράφημα:** Εξέταση για την καταγραφή της καρδιακής δραστηριότητας
- **Υπερηχοκαρδιογράφημα:** Υπερηχογραφική εξέταση της καρδιάς για την αξιολόγηση της λειτουργίας της
- **Καταγραφή καρδιακού ρυθμού:** Μακροχρόνια παρακολούθηση του καρδιακού ρυθμού για τον εντοπισμό αρρυθμιών

Αυτό σημαίνει ότι η πρόληψη του εγκεφαλικού επεισοδίου είναι προτεραιότητα για να περιοριστούν οι θάνατοι που προκαλούνται από τα εγκεφαλικά επεισόδια.

2.3 Ο ρόλος της μηχανικής μάθησης στην ιατρική

Η αρχική χρήση των αλγορίθμων μηχανικής μάθησης, το 1950 με 1960, ήταν για την ανάλυση ιατρικών δεδομένων. Σήμερα, η χρήση τους όχι μόνο έχει επεκταθεί σε περισσότερους επιστημονικούς κλάδους, αλλά έχει γίνει πιο εύκολα προσβάσιμη και πιο οικονομική από εκείνη την εποχή. [6]

Η μηχανική μάθηση (Machine Learning – ML) βελτιώνει σε μεγάλο βαθμό την ακρίβεια των διαγνώσεων που κάνουν οι γιατροί, στην προσωποποίηση των φαρμάκων και των θεραπειών για κάθε ασθενή και στις περαιτέρω δυνατότητες έρευνας. Κάποιοι από τους πιο βασικούς τρόπους με τους οποίους η μηχανική μάθηση χρησιμοποιείται στην ιατρική είναι:

- **Διάγνωση και Απεικόνιση:** Οι αλγόριθμοι μηχανικής μάθησης χρησιμοποιούνται ολοένα και περισσότερο για την ανάλυση των ιατρικών εικόνων (X-rays). Με την βοήθεια τους εντοπίζουν ανωμαλίες και παθήσεις, όπως ο καρκίνος σε πρώιμα στάδια και πολλές φορές με μεγαλύτερη ακρίβεια από τον άνθρωπο.
- **Προσωποποιημένα Φάρμακα:** Η μηχανική μάθηση βοηθάει στην στοχευμένη φαρμακευτική αγωγή για κάθε ασθενή ξεχωριστά. Αυτό γίνεται με την ανάλυση μοτίβων από δεδομένα προηγούμενων ασθενών, που μέσα σ' αυτά περιλαμβάνονται και η αντίδραση σε θεραπείες και φάρμακα, έτσι οι αλγόριθμοι μηχανικής μάθησης

δίνουν μια πρόβλεψη για την αντίδραση που θα έχουν άλλοι ασθενείς στις ίδιες θεραπείες.

- **Ανάπτυξη και Ανακάλυψη Φαρμάκων:** Η μηχανική μάθηση μπορεί να διευκολύνει την ανάπτυξη καινούργιων φαρμάκων προβλέποντας την αποτελεσματικότητα και την ασφάλεια τους προτού δοκιμαστούν στον πραγματικό κόσμο. Αυτό έχει ως αποτέλεσμα την μείωση του χρόνου και του κόστους σε σχέση με τους παραδοσιακούς τρόπους πειραμάτων.
- **Επιδημιολογία:** Οι αλγόριθμοι μηχανικής μάθησης χρησιμοποιούνται για την παρακολούθηση και πρόβλεψη επιδημιών, αναλύοντας τις τάσεις των δεδομένων στον χρόνο σε διάφορες περιοχές παγκοσμίως.
- **Αποδοτικότητα:** Τα νοσοκομεία και οι κλινικές χρησιμοποιούν μηχανική μάθηση για τη βελτιστοποίηση της λειτουργίας τους, παράδειγμα: για το κλείσιμο των ραντεβού των ασθενών και με τον τρόπο αυτόν μειώνεται ο χρόνος αναμονής, όπως και για την πρόβλεψη των ασθενών που έχουν περισσότερη ανάγκη άμεσης περίθαλψης.
- **Ρομποτική:** Στο χειρουργείο, η μηχανική μάθηση σε συνδυασμό με την ρομποτική, προσφέρει στους χειρουργούς ακρίβεια και σταθερότητα που βελτιώνει το αποτέλεσμα του χειρουργείου. Θα μπορούσε να χρησιμοποιηθεί επίσης, ως εκπαίδευση των χειρουργών και στη βοήθεια σε πιο περίπλοκα χειρουργεία.

2.4 Ανασκόπηση άρθρων πρόβλεψης εγκεφαλικών με μηχανική μάθηση

Παρακάτω παρατίθενται σχετικά άρθρα/ δημοσιεύσεις με αντικείμενο την πρόβλεψη εγκεφαλικών επεισοδίων με τη βοήθεια της μηχανικής μάθησης. Συγκεκριμένα, αναφέρονται οι διάφορες μεθοδολογίες, οι αλγόριθμοι, ο πληθυσμός που αξιολογήθηκε, τα σημαντικότερα χαρακτηριστικά και τα αποτελέσματα.

Στην έρευνα των (Khosla, et al., 2010) [7] συγκρίνεται το μοντέλο αναλογικού κινδύνου του Cox με αλγορίθμους μηχανικής μάθησης, πιο συγκεκριμένα Support Vector Machines (SVMs), χρησιμοποιώντας τα δεδομένα του Cardiovascular Health Study (CHS). Οι μεθοδολογίες που χρησιμοποιούνται είναι η Αντικατάσταση Ελλειπόντων Δεδομένων, η επιλογή χαρακτηριστικών και μοντέλα πρόβλεψης, δηλαδή εφαρμόζει SVMs και ένα νέο αλγόριθμο παλινδρόμησης

που βασίζεται στο περιθώριο, με στόχο να ξεπεράσει το παραδοσιακό Cox μοντέλο πρόβλεψης. Ο πληθυσμός πάνω στον οποίο αξιολογήθηκε η έρευνα είναι 62.001 ασθενείς. Το dataset περιλαμβάνει αρκετές πληροφορίες από δημογραφικές και βιοϊατρικές πληροφορίες μέχρι και κλινικές. Η νέα αυτή μέθοδος της επιλογής χαρακτηριστικών ‘Conservative Mean’ ήταν η πιο αποτελεσματική στον εντοπισμό προγνωστικών παραγόντων κινδύνου εγκεφαλικού επεισοδίου μέσα από διάφορα πιθανά χαρακτηριστικά. Ο αλγόριθμος Margin-based Censored Regression, σε συνδυασμό με τη μέθοδο ‘Conservative Mean’ της επιλογής χαρακτηριστικών, ξεπερνούν αισθητά όλα τα άλλα μοντέλα στις μετρήσεις Area Under the Curve (AUC). Να σημειωθεί παράλληλα ότι η ηλικία ήταν το πρώτο κύριο χαρακτηριστικό στον εντοπισμό εγκεφαλικού επεισοδίου. Επίσης, μέσω αυτής της προσέγγισης εντοπιστήκαν πιθανές αίτιες που μέχρι πρότινος δεν είχαν αξιολογηθεί ως παράγοντες πρόκλησης εγκεφαλικού επεισοδίου που οι παραδοσιακοί μέθοδοι δεν τις αναγνώριζαν έως τώρα.

Η έρευνα των (Al-Zubaidi et al., 2022) [8] έχει ως στόχο τη δημιουργία ενός επιβλεπόμενου μοντέλου πρόβλεψης εγκεφαλικού με τη χρήση διαφόρων μεθόδων ταξινόμησης μηχανικής μάθησης. Το άρθρο επίσης επισημαίνει το πόσο σημαντικό είναι αυτού του είδους οι προβλέψεις στο να ληφθούν μετρά πρόληψης ή ακόμα και στο να σωθούν ανθρώπινες ζωές. Τα δεδομένα που χρησιμοποιήθηκαν ήταν από το Kaggle με 5110 δείγματα. Για την μεθοδολογία, χρησιμοποιήθηκαν διάφοροι μέθοδοι ταξινόμησης, όπως ο Random Forest, Decision Tree, Logistic Regression και Support Vector Machine. Για την ανισορροπία που είχε το dataset, 95% χωρίς εκδήλωση εγκεφαλικού και 5% με εγκεφαλικό, έγινε η χρήση της τεχνικής SMOTE (Synthetic Minority Over-sampling Technique). Όσο για τα χαρακτηριστικά που περιλαμβάνονται ήταν εξαρχής 12 (age, gender, hypertension, heart disease, residence type, smoking status, BMI κ.α.) και έπειτα με encoding συνολικά υπήρχαν 16. Αυτό οφείλεται στο γεγονός ότι κάποια χαρακτηριστικά είχαν περισσότερες από μια πιθανές απαντήσεις, όπως το work type και το smoking status, οπότε η μετατροπή τους σε δυαδικά χαρακτηριστικά, όπου 1 είναι το «ναι» και 0 είναι το «όχι» είναι σημαντική για την καλύτερη απόδοση των μεθόδων. Τα αποτελέσματα που προέκυψαν με βάση την ορθότητα, την ακρίβεια, την ανάκληση και το F1-Score έδειξαν ότι ο Random Forest ήταν ο πιο αποτελεσματικός με ορθότητα, ακρίβεια, ευαισθησία και F1-Score μεταξύ 94% και 95%. Τα σημαντικότερα χαρακτηριστικά που συσχετίζονται με την πρόβλεψη εγκεφαλικού επεισοδίου ήταν τα εξής: 1) age (ηλικία), 2) avg glucose levels (μέσο επίπεδο σακχάρου) και 3) ever married (αν έχει παντρευτεί).

Η έρευνα των (Wu et al., 2020) [9], εστιάζει περισσότερο σε μεγαλύτερης ηλικίας ανθρώπους και πιο συγκεκριμένα από 60 ετών και άνω, κινεζικής καταγωγής, για τον εντοπισμό εγκεφαλικού με τη χρήση μηχανικής μάθησης. Τα δεδομένα που χρησιμοποιήθηκαν ήταν από το Chinese Longitudinal Healthy Longevity Study (CLHLS), με 2439 συμμετέχοντες. Για την εξισορρόπηση των δεδομένων χρησιμοποιήθηκαν τρεις διαφορετικές μέθοδοι: Random Under-Sampling (RUS), Random Over-Sampling (ROS) και Smote (Synthetic Minority Over-sampling Technique). Όσο για τα μοντέλα μηχανικής μάθησης που χρησιμοποιήθηκαν ήταν τα εξής: Random Forest (RF), Regularized Logistic Regression (RLR) και Support Vector Machine (SVM). Κατά την διαδικασία της προεπεξεργασίας από τα 2439 δείγματα απέμειναν τα 1131, λόγω της διαγραφής των ελλειπόντων δεδομένων, εκ των οποίων τα 56 είχαν υποστεί εγκεφαλικό ενώ 1075 δεν είχαν. Συνολικά τα χαρακτηριστικά τα οποία χρησιμοποιήθηκαν ήταν 15 για την πρόβλεψη εγκεφαλικού επεισοδίου με κάποια από αυτά να είναι η ηλικία (age), το φύλο (sex), αν είναι καπνιστές (smoking), αν καταναλώνουν αλκοόλ (drinking), αν έχουν διαβήτη (diabetes), το ουρικό οξύ (Uric acid), η υπέρταση (hypertension) κ.α. Τα αποτελέσματα που προέκυψαν πριν την εξισορρόπηση των δεδομένων όπως είναι φυσικό δεν ήταν καθόλου ικανοποιητικά αφού η ευαισθησία ήταν πολύ χαμηλή και το AUC δεν ξεπερνούσε το 0.50, που σημαίνει ότι θα ήταν το ίδιο με το να επιλέγαμε τυχαία. Με τη χρήση του ROS, RUS και SMOTE υπήρξε σαφέστατη βελτίωση στα αποτελέσματα. Η ευαισθησία αυξήθηκε από σχεδόν 0.00 σε 0.78 στο υψηλότερο ενώ το AUC είχε βελτίωση σε όλα τα μοντέλα με τον RF στα 0.72 και 0.71 για SVM και RLR. Τα πιο σημαντικά χαρακτηριστικά ανάμεσα στα τρία αυτά μοντέλα είναι το φύλο, η υπέρταση και το ουρικό οξύ τα οποία εμφανίζονται σταθερά και στα τρία μοντέλα.

Η έρευνα των (Mridha et al., 2023) [10], έχει ως βασικό στόχο την ανάπτυξη κατανοητών και εξηγήσιμων μοντέλων πρόβλεψης τα οποία είναι και προσβάσιμα μέσω ιστοσελίδας. Τα δεδομένα που χρησιμοποιήθηκαν ήταν από το Kaggle με 5110 πληθυσμό και 11 χαρακτηριστικά (12 με την κλάση). Για την ανισορροπία που υπήρχε, 249 με εγκεφαλικό και 4861 χωρίς, χρησιμοποιήθηκε η τεχνική SMOTE. Χρησιμοποιήθηκαν επίσης, δυο μέθοδοι τεχνητής νοημοσύνης: η SHAP (Shapley Additive Explanations) και η LIME (Local Interpretable Model-agnostic Explanations) και οι δυο αυτοί μέθοδοι έχουν τον ίδιο στόχο, δηλαδή να προσφέρουν στο χρήστη μια πιο απλή και εύκολη στην κατανόηση διαδικασία. Τα αποτελέσματα των αλγορίθμων ήταν τα εξής: ο καλύτερος αλγόριθμος ήταν ο Random Forest με ορθότητα 90.36 % χωρίς data leakage, δηλαδή οι πληροφορίες του μοντέλου πάνω στο οποίο θέλουμε δοκιμάσουμε τους αλγορίθμους δεν αντλούν πληροφορίες από τα δεδομένα με το οποίο εκπαιδεύτηκε το

μοντέλο και 82.23 % με data leakage, δηλαδή δεδομένα τα οποία δεν θα έπρεπε να χρησιμοποιηθούν κατά τη διαδικασία της εκπαίδευσης χρησιμοποιούνται και αυτό πολλές φορές επηρεάζει τα αποτελέσματα. Αμέσως μετά τον Random Forest είναι ο XGB με 89.02% χωρίς data leakage και 83.42% με data leakage. Οι αλγόριθμοι Logistic Regression, SVM και KNN (K-Nearest Neighbors) είχαν επίσης αρκετά καλά αποτελέσματα αλλά δεν ξεπερνούσαν τον Random Forest και XGB. Ως προς τα πιο σημαντικά χαρακτηριστικά η ηλικία ήταν ξεκάθαρος παράγοντας πρόκλησης εγκεφαλικού επεισοδίου, με το μέσο επίπεδο σακχάρου, τον τύπο δουλειάς, τον τόπο κατοικίας, το φύλο και αν έχει παντρευτεί να ακολουθούν. Στο άρθρο επισημαίνεται η αποτελεσματικότητα της μηχανικής μάθησης που χρησιμοποιείται και η δυνατότητα συγχώνευσης στοιχείων τεχνητής νοημοσύνης στη βοήθεια κλινικών αποφάσεων.

Το βασικό θέμα του άρθρου των (Dev et al., 2022) [11], είναι η ανάγκη της πρόβλεψης του εγκεφαλικού λόγω της επίδρασης των συνεπειών που έχει στα άτομα που το έχουν υποστεί και κατά επέκταση στη κοινωνία. Επίσης, με σκοπό τη βελτίωση των ιατρικών διαγνώσεων μέσω της τεχνολογίας που εξελίσσεται συνεχώς. Το άρθρο στηρίχτηκε σε δεδομένα που χρησιμοποιήθηκαν από το Kaggle, με πληθυσμό 29,072. Τα χαρακτηριστικά είναι 11 (μοναδικός αριθμός ασθενή, ηλικία, φύλο, υπέρταση, παθήσεις καρδιάς, οικογενειακή κατάσταση, είδος επαγγέλματος, είδος κατοικίας, μέσο επίπεδο σακχάρου, BMI και αν είναι καπνιστές) και 1 τιμή εξόδου η οποία προσδιορίζει αν ο ασθενής έχει υποστεί ή όχι εγκεφαλικό. Με την χρήση sub-sampling μεθόδων έγινε εξισορρόπηση των δεδομένων ώστε να εφαρμοστούν πάνω οι αλγόριθμοι μηχανικής μάθησης. Ο αλγόριθμος Νευρονικών Δικτύων (Neural Network), με ακρίβεια 77%, είχε τις καλύτερες επιδόσεις συνολικά και ήταν πιο αποδοτικός όταν χρησιμοποιήθηκε με λιγότερα χαρακτηριστικά (ηλικία, μέσο επίπεδο σακχάρου, παθήσεις καρδιάς και υπέρταση) που είναι και τα τέσσερα πιο σημαντικά χαρακτηριστικά για την πρόβλεψη εγκεφαλικού, σύμφωνα με το άρθρο. Ακολουθούν οι αλγόριθμοι Decision Tree και Random Forest με 74% ορθότητα και οι δυο, με τον decision tree να έχει λίγο καλύτερα συνολικά αποτελέσματα όταν τα χαρακτηριστικά που χρησιμοποιήθηκαν μειωθήκαν στα τέσσερα πιο σημαντικά. Συμπερασματικά, τα μοντέλα μηχανικής μάθησης μπορούν να βοηθήσουν στην έγκαιρη πρόβλεψη εγκεφαλικού επεισοδίου και πιο συγκεκριμένα η χρήση υποσυνόλων των πιο σημαντικών χαρακτηριστικών μπορεί να βοηθήσει σε πιο εύστοχες και ακριβείς προβλέψεις.

Η έρευνα των (Kokkotis et al., 2022) [12], έχει ως στόχο οι αποφάσεις που λαμβάνει το μοντέλο πρόβλεψης να είναι κατανοητές για να μπορούν να είναι κλινικά και ιατρικά αποδεκτές. Τα δεδομένα που χρησιμοποιήθηκαν για την έρευνα ήταν από το Kaggle με 43,400 πληθυσμό

και 10 χαρακτηριστικά (ηλικία, φύλο, υπέρταση, παθήσεις καρδιάς, οικογενειακή κατάσταση, είδος επαγγέλματος, είδος κατοικίας, μέσο επίπεδο σακχάρου, BMI και αν είναι καπνιστές) με 1 τιμή εξόδου. Για την εξισορρόπηση των δεδομένων έγινε χρήση Random Under-Sampling και 10-fold-cross-validation για κάθε fold ξεχωριστά. Όσο για τους καλύτερους σε απόδοση αλγόριθμους, ο Logistic Regression (LR) είναι ο πρώτος σε ορθότητα με 73.52% και με ευαισθησία στο 78,12%. Ακολουθούν οι αλγόριθμοι SVM και Multi-Layer Perceptron (MLP) με ελάχιστα χαμηλότερη ορθότητα αλλά λίγο υψηλότερη ευαισθησία στα 80.48% και 81.40% αντίστοιχα. Επίσης, με τη βοήθεια της SHAP μεθόδου η κατανόηση από τον χρήστη και η διαφάνεια που προσφέρει στις προβλέψεις παίζουν σημαντικό ρόλο στη χρήση τους σε κλινικά σενάρια.

Η επόμενη έρευνα των (Sailasya et al., 2021) [13], αναφέρεται στην χρησιμότητα των αλγορίθμων μηχανικής μάθησης για την πρόβλεψη εγκεφαλικών επεισοδίων και στην έλλειψη της εφαρμογής τους. Τα δεδομένα, στα οποία θα χρησιμοποιηθούν αυτοί οι αλγόριθμοι, είναι από το Kaggle με 5110 πληθυσμό και 11 χαρακτηριστικά (μοναδικός αριθμός ασθενή, ηλικία, φύλο, υπέρταση, παθήσεις καρδιάς, οικογενειακή κατάσταση, είδος επαγγέλματος, είδος κατοικίας, μέσο επίπεδο σακχάρου, BMI και αν είναι καπνιστές) και 1 τιμή εξόδου η οποία προσδιορίζει αν ο ασθενής κινδυνεύει ή όχι από εγκεφαλικό επεισόδιο. Για την εξισορρόπηση των δεδομένων, έγινε χρήση undersampling. Στο σύνολο εφαρμοστήκαν έξι αλγόριθμοι (Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, SVM και Naïve Bayes). Ο Naïve Bayes είχε τα καλύτερα αποτελέσματα με ορθότητα στο 82% , ευαισθησία στο 85.7% και ακρίβεια στο 79.2%. Ακολουθούν οι αλγόριθμοι K-Nearest Neighbors και SVM με ορθότητα στο 80% και οι δυο, ευαισθησία στο 83.7% και 83.8% και ακρίβεια στο 77.4% και 78.6% αντίστοιχα. Τέλος, τονίζεται ότι η έρευνα αυτή μπορεί να επεκταθεί ακόμα περισσότερο με φωτογραφικό υλικό και το πόσο σημαντική είναι η πρόληψη και η έγκαιρη πρόβλεψη του εγκεφαλικού.

Η έρευνα των (Bandi et al., 2020) [14], επικεντρώνεται στην ανάπτυξη ενός καινούργιου αλγορίθμου που ονομάζεται Stroke Prediction (SPN) και βασίζεται σε μια καλύτερη εκδοχή του Random Forest. Διάφοροι αλγόριθμοι μηχανικής μάθησης χρησιμοποιούνται για την πρόβλεψη εγκεφαλικού επεισοδίου, όπως ο Gaussian Naïve Bayes, Linear Regression, Logistic Regression, K-Means Clustering, SVM, Decision Trees, Random Forest και AdaBoost. Ο πληθυσμός πάνω στον οποίο διεκπαιρώθηκε η έρευνα είναι 4,799 με 10 χαρακτηριστικά (ηλικία,

NIHSS¹, mRS², συστολική αρτηριακή πίεση, διαστολική πίεση, επίπεδα σακχάρου, επίπεδα χοληστερίνης, παράλυση, αν είναι καπνιστές και BMI. Συγκριτικά τα αποτελέσματα μεταξύ του καινούργιου αλγορίθμου, Stroke Prediction, και του Random Forest παρουσιάζουν ιδιαίτερο ενδιαφέρον. Συγκεκριμένα, ο Stroke Prediction αλγόριθμος είχε ορθότητα 96.97% σε σύγκριση με τον Random Forest που είχε 94.23%. Όπως και στην ορθότητα έτσι και στις υπόλοιπες μετρήσεις ο SPN ήταν αισθητά καλύτερος. Ακριβώς μετά, ακολουθεί ο αλγόριθμος Decision Tree με 93.12% ορθότητα με τους υπολοίπους αλγορίθμους να βρίσκονται στο 80% με 85%. Τα τέσσερα πιο σημαντικά χαρακτηριστικά σύμφωνα με τον καλύτερο σε επιδόσεις αλγόριθμο είναι τα εξής: 1) επίπεδα σακχάρου, 2) επίπεδα χοληστερίνης, 3) συστολική πίεση και 4) ηλικία.

Η έρευνα των (Islam et al. 2021) [15] αποσκοπεί στην έγκαιρη πρόληψη των εγκεφαλικών επεισοδίων. Ο πληθυσμός πάνω στο οποίον αξιολογήθηκε η έρευνα είναι 5110 με 12 χαρακτηριστικά (ηλικία, φύλο, υπέρταση κ.α.) εκ των οποίων ένα από αυτά είναι και η τιμή εξόδου. Για την εξισορρόπηση των δεδομένων έγινε χρήση της SMOTE μεθόδου. Ως προς τους αλγορίθμους που χρησιμοποιήθηκαν ήταν ο Random Forest, Decision Tree, K-Nearest Neighbors, Naïve Bayes, Logistic Regression και AdaBoost. Ο Random Forest ήταν ο καλύτερος σε επιδόσεις αλγόριθμος σε όλες τις μετρήσεις (ακρίβεια, ευαισθησία και F1-Score) με 96% και στις τρεις. Ακολουθούν οι αλγόριθμοι, Decision Tree με 93% σε όλες τις μετρήσεις, K-Nearest Neighbors με 90% ακρίβεια και F1-Score και 91% ευαισθησία και τέλος ο Logistic Regression με 87% σε όλες τις μετρήσεις.

Για την έρευνα των (Zhu et al., 2023) [16], ο συνολικός πληθυσμός πριν την προεπεξεργασία ήταν 8,276 και αφού αφαιρέθηκαν 757 ασθενείς, λόγο έλλειψης δεδομένων, απέμειναν 7,483 ασθενείς. Τα δεδομένα προέρχονται από MIMIC-IV (Medical Information Mart for Intensive Care) database. Κάποια από τα χαρακτηριστικά που περιλαμβάνονται είναι η ηλικία, η εθνικότητα, το BMI, ο κορεσμός του οξυγόνου, ο αριθμός ερυθρών αιμοσφαιρίων και άλλες παθήσεις του νευρικού συστήματος. Οι αλγόριθμοι που χρησιμοποιήθηκαν για την πρόβλεψη εγκεφαλικού επεισοδίου είναι ο NODE (Neural Oblivious Decision Ensemble), ο CatBoost, ο XGBoost, ο LightGBM, ο FCNN (Fully Connected Neural Network) και ο Logistic Regression. Ο CatBoost ήταν ο καλύτερος σε επιδόσεις αλγόριθμος με ορθότητα 89.93%, ακρίβεια 81.55, AUC

¹ National Institutes of Health Stroke Scale

² Modified Rankin Scale

92.17% και F1-Score 67.35%. Ο δεύτερος καλύτερος αλγόριθμος ήταν ο XGBoost με ορθότητα 89.69%, ακρίβεια 77.83%, AUC 91.75% και F1-Score 68.90%. Τα τρία πιο κρίσιμα χαρακτηριστικά που είχαν την μεγαλύτερη επιρροή στο μοντέλο για τον εντοπισμό εγκεφαλικού ήταν ο κορεσμός του οξυγόνου, ο αριθμός ερυθρών αιμοσφαιρίων και άλλες παθήσεις του νευρικού συστήματος.

3 Μεθοδολογία

3.1 Δεδομένα

Τα δεδομένα τα οποία χρησιμοποιήθηκαν, είναι από το Kaggle[17]. Σε αυτά περιλαμβάνονται 5110 γραμμές, δείγματα και 12 στήλες, χαρακτηριστικά. Πιο συγκεκριμένα τα χαρακτηριστικά περιλαμβάνουν: “id”, “gender”, “age”, “hypertension”, “heart_disease”, “ever_married”, “work_type”, “residence_type”, “avg_glucose_level”, “bmi”, “smoking_status” και “stroke”. Παρακάτω ακολουθεί ο πίνακας με την ανάλυση του κάθε χαρακτηριστικού και τις τιμές που λαμβάνει.

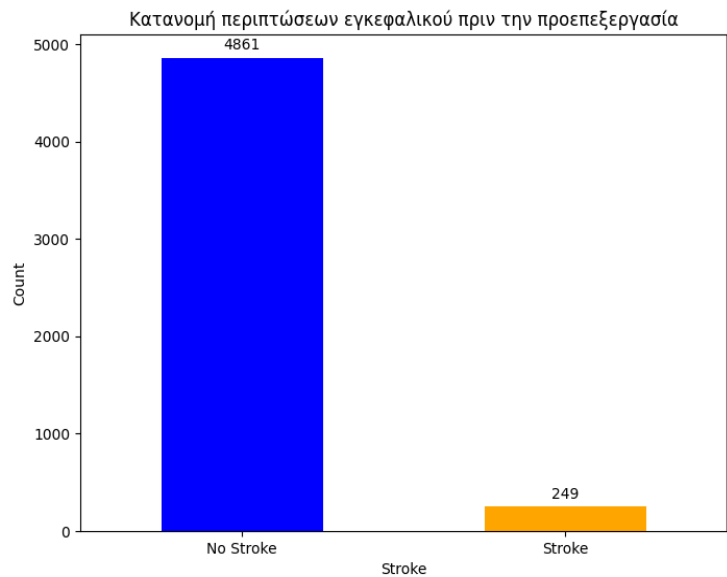
Πίνακας 1:

Χαρακτηριστικά δεδομένων

Όνομα Χαρακτηριστικού	Τύπος μεταβλητής	Περιγραφή
id	Διακριτή	Ο μοναδικός αριθμός του ασθενή
gender	Κατηγορική (“male”, “female” ή “other”)	Το φύλο του ασθενή
age	Συνεχής (24)	Η ηλικία του ασθενή
hypertension	Δυαδική (0 ή 1)	Αν ο ασθενής έχει ή όχι υπέρταση
heart_disease	Δυαδική (0 ή 1)	Αν ο ασθενής έχει ή όχι κάποια πάθηση στη καρδιά
ever_married	Δυαδική (Yes ή No)	Αν ο ασθενής έχει παντρευτεί ή όχι
work_type	Κατηγορική (“private”, “Self-employed”, “children”, “Govt_job” ή “Never_worked”)	Το είδος της δουλειάς που κάνει ο ασθενής
residence_type	Κατηγορική (“Urban” ή “Rural”)	Ο τύπος κατοικίας του ασθενή
avg_glucose_level	Συνεχής (205.77)	Τα μέσα επίπεδα σακχάρου του ασθενή
bmi	Συνεχής (31.1)	Ο Δείκτης Μάζας Σώματος του ασθενή
smoking_status	Κατηγορική (“never smoked”, “Unknown”, “formerly smoked” ή “smokes”)	Αν ο ασθενής καπνίζει, δεν καπνίζει, κάπνιζε ή δεν προσδιορίζετε
stroke	Δυαδική (0 ή 1)	Η έξοδος που έχει ως τιμή 0, αν δεν έχει εγκεφαλικό και 1 στην περίπτωση που έχει

3.1.1 Καθαρισμός Δεδομένων

Σημαντικό ρολό, στην αξιοπιστία και ακρίβεια ενός μοντέλου πρόβλεψης, έχει το κατά πόσο ισορροπημένα είναι τα δεδομένα. Στην Εικόνα 2, φαίνεται ξεκάθαρα ότι στην περίπτωση αυτή τα δεδομένα είναι μη ισορροπημένα, κάτι το οποίο θα στρεβλώσει τα αποτελέσματα τα οποία θα προκύψουν αν δεν γίνει η προεργασία των δεδομένων. Υπάρχουν διάφορες τεχνικές εξισορρόπησης των δεδομένων, όπως η random oversampling, SMOTE και η random undersampling.



Εικόνα 2: Τα δεδομένα πριν την προεπεξεργασία

Σε αυτή την περίπτωση χρησιμοποιείται η τεχνική random undersampling διότι κρίθηκε ως καταλληλότερη. Η επιλογή στηρίχθηκε στο γεγονός ότι τα δεδομένα πάνω στα οποία θα εφαρμοστούν αργότερα οι αλγόριθμοι, πρέπει να είναι όσο το δυνατόν πιο αληθινά και όχι τεχνητά δημιουργημένα έτσι ώστε το μοντέλο πρόβλεψης να επεξεργάζεται πραγματικές συνθήκες.

Για τη τεχνική Random Undersampling που πραγματοποιήθηκε, η διαδικασία είναι η εξής:

- Διαχωρισμός των χαρακτηριστικών και της τιμής εξόδου (“stroke”).

```
# Split the data into features and target
X = data.drop('stroke', axis=1)
y = data['stroke']
```

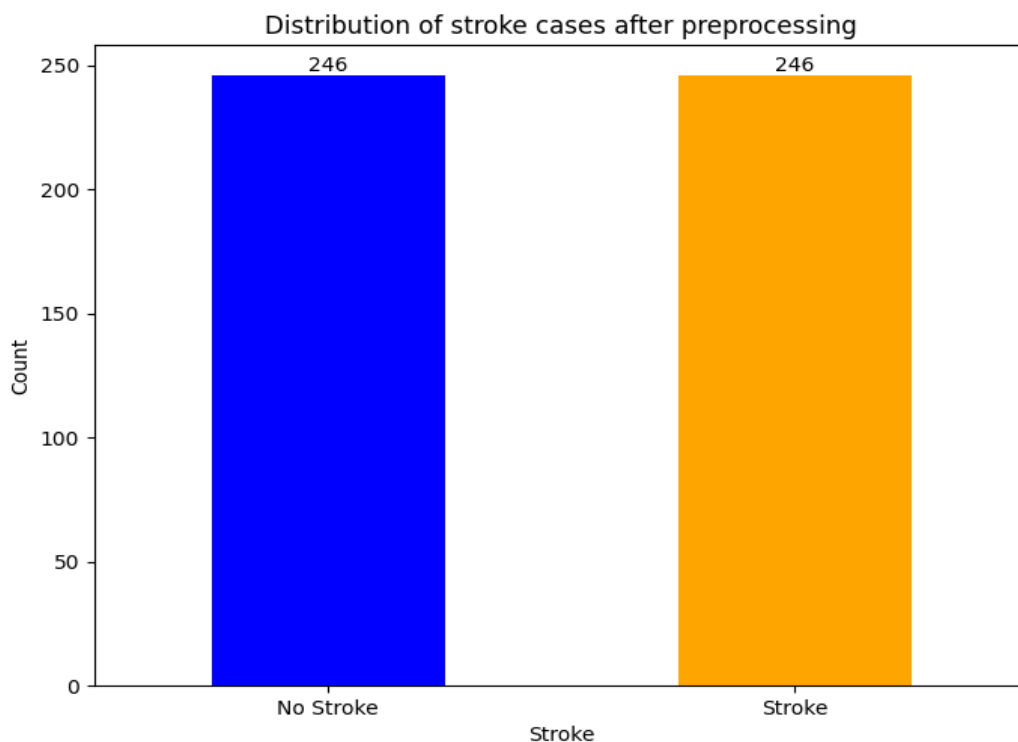
Εικόνα 3: Διαχωρισμός των χαρακτηριστικών από την έξοδο

- Εντοπισμός της μείζονα κλάση (majority class) και της ελάσσονα κλάση (minority class) για το ποια κλάση (“no_stroke” και “stroke”) έχει περισσότερα δείγματα και ποια λιγότερα.

- Τυχαία επιλογή δειγμάτων από την μείζονα κλάση, δηλαδή γίνεται τυχαία επιλογή από τη κλάση με τα περισσότερα δείγματα ώστε να έχει τον ίδιο αριθμό δειγμάτων με την ελάσσονα κλάση.
- Τελικά, συγχωνεύονται τα δείγματα που επιλέχθηκαν από την μείζονα κλάση με όλα τα δείγματα από τη ελάσσονα κλάση και έτσι δημιουργήθηκε ένα ισορροπημένο σύνολο δεδομένων, όπως φαίνεται και στην *Εικόνα 5*.

```
# Undersampling
rus = RandomUnderSampler(random_state=1)
X_res, y_res = rus.fit_resample(X, y)
```

Εικόνα 4: Εφαρμογή Random Undersampling στα δεδομένα



Εικόνα 5: Τα δεδομένα μετά την προεπεξεργασία

Τα αποτελέσματα της random undersampling τεχνικής φαίνονται στην *Εικόνα 5*. Τώρα τα δεδομένα παρά το γεγονός ότι είναι σαφώς λιγότερα, είναι ισορροπημένα οπότε τα αποτελέσματα δεν θα είναι επηρεασμένα προς τη μια ή την άλλη κλάση.

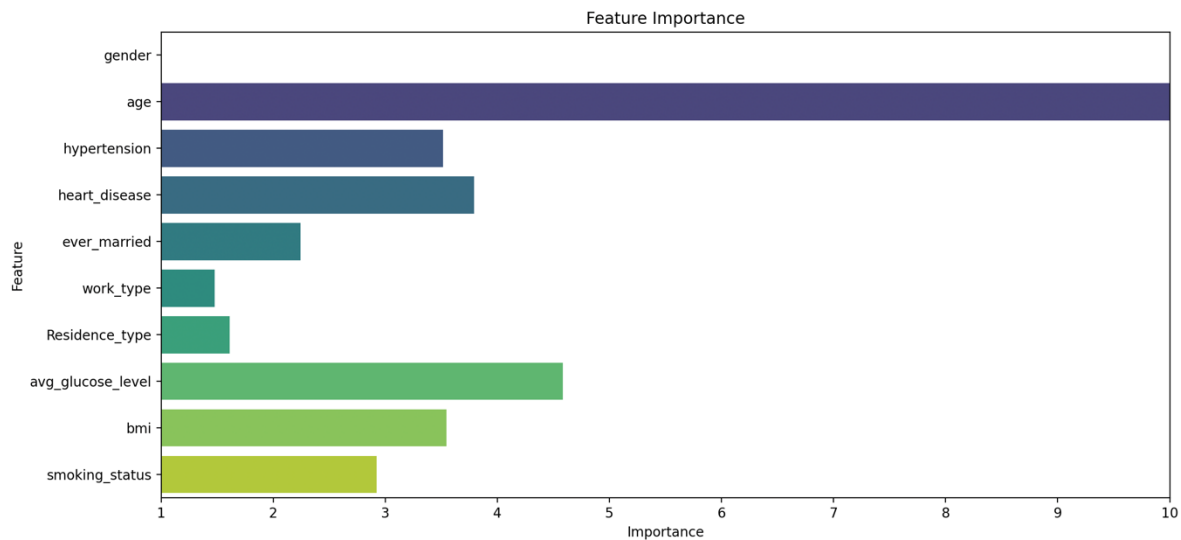
Συνεχίζοντας την προεπεξεργασία είναι σημαντικό να γίνει έλεγχος όλων των χαρακτηριστικών που περιγράφουν τα δεδομένα. Όπως αναφέρεται και στον *Πίνακα 1*, τα χαρακτηριστικά είναι 11 εκ των οποίων 1 είναι η τιμή εξόδου. Το πρώτο χαρακτηριστικό με ονομασία “id” δεν είναι χρήσιμο για την πρόβλεψη εγκεφαλικού αφού η μόνη χρήση του είναι η τοποθέτηση ενός μοναδικού αριθμού σε κάθε ασθενή, οπότε αφαιρείται τελείως αυτό το χαρακτηριστικό. Τα υπόλοιπα 10 χαρακτηριστικά παραμένουν στην ανάλυση διότι το κάθε ένα δίνει κάποιο δεδομένο το οποίο μπορεί να επηρεάσει τα αποτελέσματα.

Η στήλη “bmi” είναι η μόνη με ελλείπουσες τιμές, όπως φαίνεται και στην *Εικόνα 6*. Για την διαχείριση αυτών τιμών έγινε χρήση της μέσης τιμής των δεδομένων, δηλαδή οι 201 ελλείπουσες τιμές αντικαταστάθηκαν με τη μέση τιμή της στήλης “bmi”.

```
Missing values in each column:
id                                0
gender                            0
age                               0
hypertension                      0
heart_disease                    0
ever_married                     0
work_type                        0
Residence_type                   0
avg_glucose_level                 0
bmi                              201
smoking_status                   0
stroke                           0
```

Εικόνα 6: Οι ελλείπουσες τιμές των δεδομένων

3.1.2 Σημαντικότητα Χαρακτηριστικών



Εικόνα 7: Σημαντικότητα Χαρακτηριστικών (Feature Importance)

Από το παραπάνω γράφημα βγαίνει το συμπέρασμα ότι το πιο σημαντικό χαρακτηριστικό που επηρεάζει το αποτέλεσμα των αλγορίθμων είναι η ηλικία. Το δεύτερο πιο σημαντικό χαρακτηριστικό είναι τα μέσα επίπεδα σακχάρου και το τρίτο οι παθήσεις καρδιάς.

3.1.3 Κωδικοποίηση Χαρακτηριστικών

Σε πολλές περιπτώσεις τα χαρακτηριστικά, τα οποία περιλαμβάνονται σε ένα σύνολο δεδομένων, δεν ανήκουν στην ίδια κατηγορία. Δηλαδή, άλλα είναι αριθμητικά και άλλα είναι κατηγορικά (1, 2, 3... και «μπλε», «κόκκινο», «κίτρινο»...). Για μερικούς αλγορίθμους αυτό δεν είναι αποτελεί πρόβλημα στην εκμάθηση τους αλλά σε αλγορίθμους, όπως είναι ο AdaBoost και ο K-Nearest Neighbors, απαιτείται οι τιμές που λαμβάνουν τα χαρακτηριστικά να είναι αριθμητικές.

Ως προς τις μεθόδους που μπορούν να χρησιμοποιηθούν για την μετατροπή κατηγορικών μεταβλητών σε αριθμητικές μεταβλητές περιλαμβάνονται:

- **Label Encoding:** δηλαδή, γίνεται μετατροπή των κατηγορικών τιμών σε μοναδικούς ακέραιους αριθμούς. Για παράδειγμα το χαρακτηριστικό “φύλο” οι τιμές που έπαιρνε πριν τη μέθοδο ήταν “άνδρας”, “γυναίκα” ή “άλλο”, ενώ μετά μετατράπηκαν σε 0, 1 και 2 αντίστοιχα.

Φύλο	Φύλο
Ανδρας	0
Γυναίκα	1
Άλλο	2

- One-hot Encoding:** είναι ένας ακόμη τρόπος μετατροπής κατηγορικών μεταβλητών με τη διαφορά ότι από τη στήλη του μοναδικού χαρακτηριστικού, πχ κατοικίδιο, δημιουργούνται περισσότερες-καινούργιες στήλες για κάθε κατηγορία αυτού του χαρακτηριστικού και δείχνει το αν υπάρχει ή όχι αυτή η εκάστοτε κατηγορία για κάθε δείγμα. Με αυτή την μέθοδο δεν λαμβάνεται υπόψιν αν υπάρχει κάποια συσχέτιση, ιεραρχική σειρά μεταξύ των κατηγοριών το οποίο είναι ωφέλιμο όταν υπάρχουν ονομαστικά/κατηγορικά δεδομένα. Παρακάτω αποτυπώνεται η μετατροπή:

Κατοικίδιο	Κατ_γατα	Κατ_σκυ- λος	Κατ_παπαγαλος
Σκύλος	0	1	0
Γάτα	1	0	0
Παπαγάλος	0	0	1
Γάτα	1	0	0
Σκύλος	0	1	0
Γάτα	1	0	0

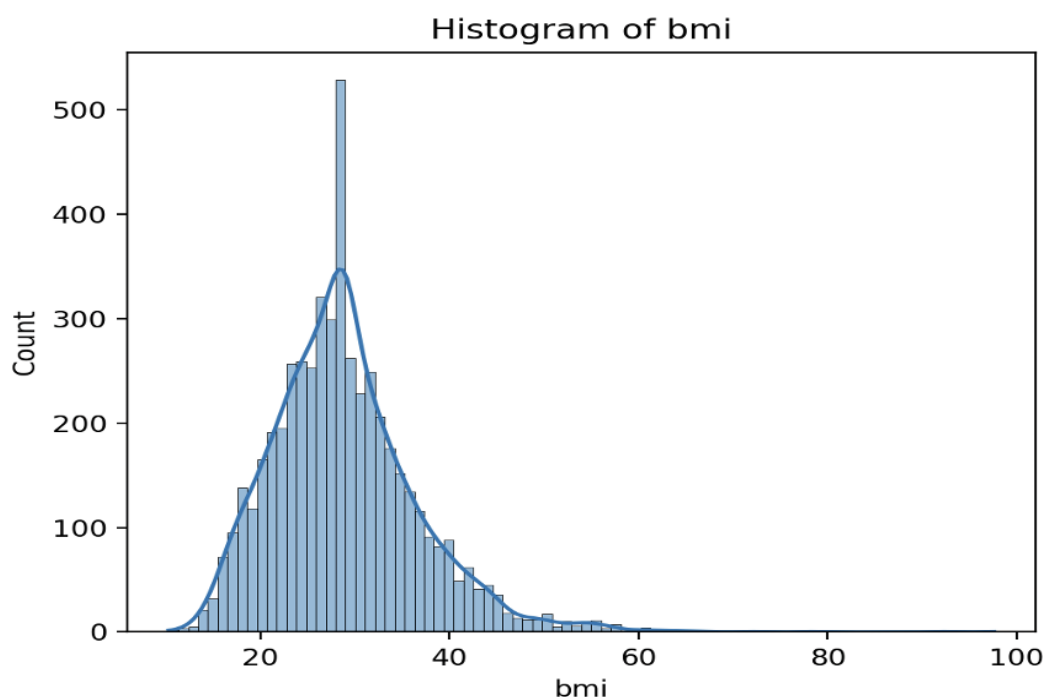
- Διαγραφή κατηγορικών χαρακτηριστικών:** Με αυτό τον τρόπο διαγράφεται εξολοκλήρου η στήλη του χαρακτηριστικού. Αν και είναι πιο εύκολος στην εκτέλεση από τους άλλους τρόπους, δεν εφαρμόζεται συχνά. Αυτό οφείλεται στο γεγονός ότι με τη διαγραφή ενός χαρακτηριστικού από τα δεδομένα, χάνεται πολύ πιθανό χρήσιμη για την ανάλυση πληροφορία.

Ανάμεσα στις δυο αυτές μεθόδους, Label Encoding και One-hot Encoding, δεν υπήρχαν μεγάλες διαφορές, εκτός από τους αλγορίθμους Logistic Regression και Decision Tree, που η διαφορά αγγίζει το 5%. Οπότε η επιλογή ανάμεσα στις δυο αυτές μεθόδους δεν είναι κρίσιμη. Για κάθε χαρακτηριστικό που έχει κατηγορική τιμή (“gender”, “ever_married”, “work_type”, “Residence_type” και “smoking_status”) χρησιμοποιήθηκε η μέθοδος Label Encoding. Οι

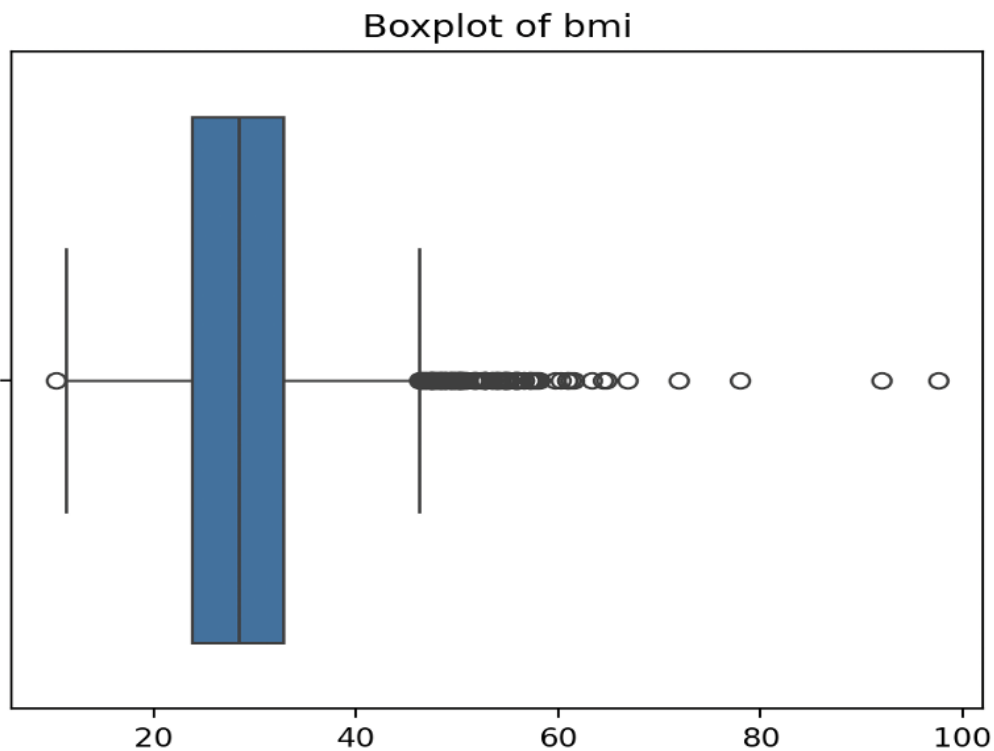
λόγοι για τους οποίους χρησιμοποιείται αυτή μέθοδος είναι διότι πολλοί από τους αλγορίθμους μηχανικής μάθησης που βασίζονται σε μαθηματικούς υπολογισμούς, όπως ο Linear Regression, ο Logistic Regression και ο SVM, χρειάζονται αριθμητικές τιμές ως είσοδο. Οι αλγόριθμοι οι οποίοι βασίζονται σε μετρικές απόστασης, όπως ο K-Nearest Neighbors και K-Means Clustering, χρειάζονται επίσης αριθμητικές τιμές εισόδου διότι με τις κατηγορικές τιμές είναι αδύνατον να υπολογιστεί η απόσταση μεταξύ των σημείων. Ένας ακόμη σημαντικός λόγος είναι η αυξημένη απόδοση και ακρίβεια των μοντέλων μηχανικής μάθησης μέσω της μεθόδου Label Encoding αφού διευκολύνεται η επεξεργασία των δεδομένων από αυτό.

3.1.4 Ακραίες Τιμές

Οι ακραίες τιμές (outliers), είναι τιμές οι οποίες διαφέρουν σημαντικά από την πλειονότητα των δεδομένων ενός χαρακτηριστικού. Όταν υπάρχουν πολλές ακραίες τιμές στα δεδομένα, κατά την διαδικασία της εκπαίδευσης του μοντέλου προκύπτουν σφάλματα και λανθασμένα αποτελέσματα. Αυτό μπορεί να οφείλεται σε διάφορους παράγοντες όπως λόγω κάποιου σφάλματος κατά την μέτρηση ή κατά την διαδικασία συλλογής των δεδομένων. Όπως φαίνεται και παρακάτω στην *Εικόνα 8* και *Εικόνα 9*, φαίνονται ξεκάθαρα οι τιμές οι οποίες αποκλίνουν από όλες τις υπόλοιπες.

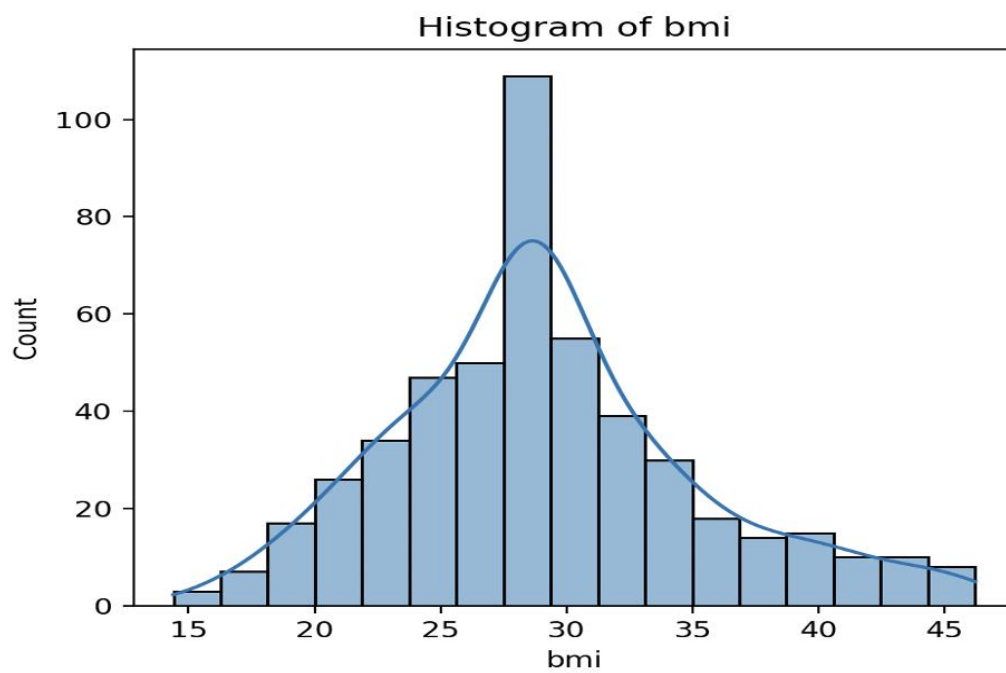


Εικόνα 8: Ιστόγραμμα του χαρακτηριστικού 'bmi' πριν τη διαγραφή

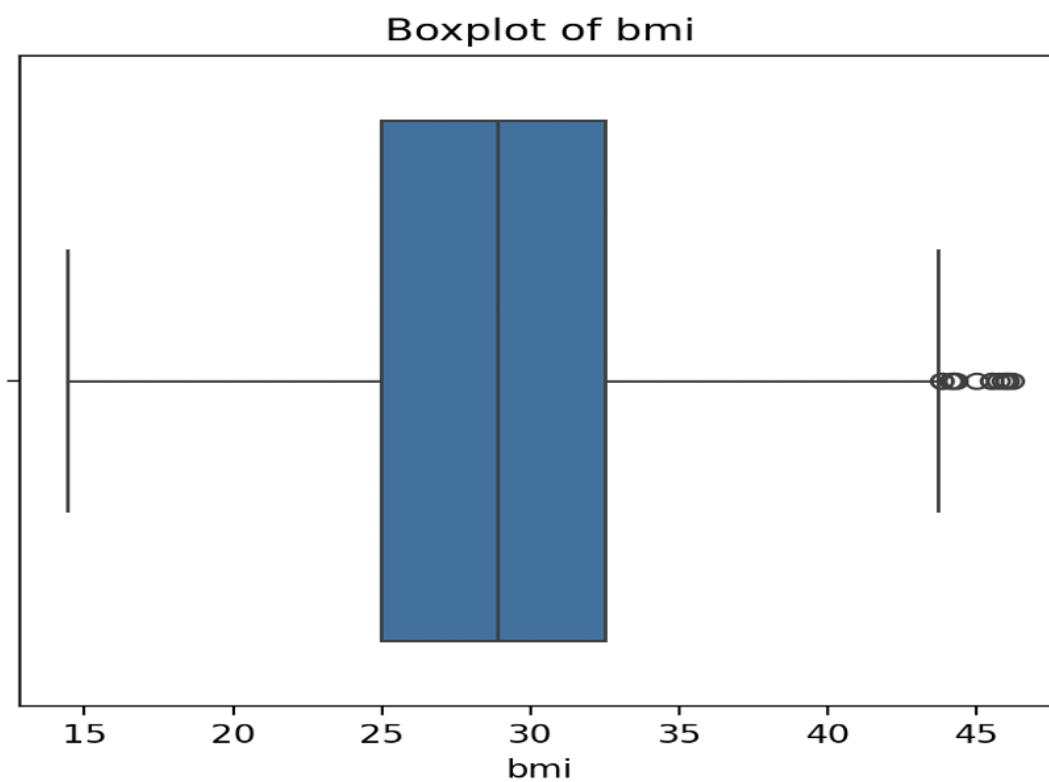


Εικόνα 9: Boxplot του χαρακτηριστικού 'bmi' πριν τη διαγραφή

Επειδή οι τιμές οι οποίες ξεπερνούν τον μέσο όρο είναι λίγες, μπορούμε απλώς να τις διαγράψουμε. Στο ιστόγραμμα φαίνεται ότι οι ακραίες τιμές ξεκινούν από το 60 και φτάνουν μέχρι και το 100, οπότε θα διαγραφούν όλες τιμές του χαρακτηριστικού “bmi” είναι από 60 και πάνω και το αποτέλεσμα φαίνεται στην *Εικόνα 10* και *Εικόνα 11*. Συγκεκριμένα, μειώνεται ο θόρυβος και σε πολλές περιπτώσεις αυξάνεται η ορθότητα των αποτελεσμάτων.



Εικόνα 10: Ιστόγραμμα του χαρακτηριστικού 'bmi' μετά τη διαγραφή



Εικόνα 11: Boxplot του χαρακτηριστικού 'bmi' μετά τη διαγραφή

3.2 Cross Validation

Η τεχνική cross validation χρησιμοποιείται στη μηχανική μάθηση για την αξιολόγηση της ικανότητας ενός μοντέλου να γενικεύσει ένα ανεξάρτητο σύνολο δεδομένων. Αυτή η διαδικασία βοηθάει στην εκτίμηση της απόδοσης του μοντέλου και διασφαλίζει ότι δεν υπάρχει υπερπροσαρμογή στα δεδομένα που χρησιμοποιούνται για εκπαίδευση. Συγκεκριμένα περιλαμβάνει την διαίρεση του αρχικού συνόλου δεδομένων σε υποσύνολα, την εκπαίδευση του μοντέλου σε ορισμένα από αυτά (σύνολο εκπαίδευσης) και τη δοκιμή του στα υπόλοιπα υποσύνολα (σύνολο επικύρωσης). Αυτή η διαδικασία επαναλαμβάνεται πολλές φορές για να διασφαλιστεί ότι κάθε σημείο δεδομένων έχει την δυνατότητα να βρίσκεται στο σύνολο εκπαίδευσης και στο σύνολο επικύρωσης. Έπειτα, από κάθε μετρική απόδοσης της κάθε επανάληψης βγαίνει ο μέσος όρος και έτσι δίνεται μια πιο αξιόπιστη εκτίμηση του μοντέλου.

Ένας από τους πιο γνωστούς τύπους cross validation είναι ο K-fold cross validation. Η διαδικασία που ακολουθεί αυτός ο τύπος είναι η εξής:

- Τα δεδομένα χωρίζονται σε K ίσων μεγεθών υποσύνολα (π.χ. $K=10$)
- Η εκπαίδευση του μοντέλου γίνεται σε $K-1$ υποσύνολα και επικυρώνεται στο υπολειπόμενο υποσύνολο. Αυτή η διαδικασία επαναλαμβάνεται K φορές, κάθε φορά με διαφορετικό υποσύνολο ως σύνολο επικύρωσης
- Τέλος, για κάθε υποσύνολο υπολογίζονται οι μετρικές απόδοσης (ορθότητα, ακρίβεια...) και βγαίνει ο μέσος όρος απόδοσης για όλες τις K επαναλήψεις

4 Αλγόριθμοι και Εκπαίδευση Μοντέλων

4.1 Διαχωρισμός Δεδομένων

Πριν γίνει η εκπαίδευση του κάθε μοντέλου, είναι σημαντικό να διαχωριστούν τα δεδομένα σε σύνολα εκπαίδευσης και δοκιμών. Στόχος είναι τα μοντέλα αυτά να μπορούν να κάνουν προβλέψεις σε δείγματα τα οποία δεν υπήρχαν ήδη στα δεδομένα. Τα σύνολα εκπαίδευσης είναι τα δείγματα, τα οποία επιλέχθηκαν τυχαία, για την εκπαίδευση των μοντέλων τόσο στον εντοπισμό μοτίβων όσο και στην συσχέτιση των χαρακτηριστικών των δεδομένων. Τα σύνολα δοκιμών είναι τα δείγματα πάνω στα οποία τα μοντέλα αξιολογούνται για την απόδοσή τους μετά την εκπαίδευση. Συνήθως αυτό το σύνολο είναι μικρότερο σε μέγεθος. Σε αυτά τα μοντέλα εφαρμόστηκε η αναλογία 80-20, δηλαδή 80% για το σύνολο εκπαίδευσης και 20% για το

σύνολο δοκιμών. Αυτό επιτεύχθηκε με τη μέθοδο ‘train_test_split’ της βιβλιοθήκης ‘scikit-learn’ [18], εξασφαλίζοντας την τυχαία αλλά επαναλαμβανομένη κατανομή των δεδομένων.

4.2 Αξιολόγηση Μοντέλων και Μετρικές Απόδοσης

4.2.1 Πίνακας Σύγχυσης

Ο πίνακας σύγχυσης (confusion matrix) αποτελεί πολύτιμο εργαλείο στην αξιολόγηση των αποδόσεων των ταξινόμησης μοντέλων. Ουσιαστικά συγκρίνει τις πραγματικές τιμές με τις προβλέψεις που έχει κάνει το μοντέλο και η δομή του έχει ως εξής:

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

- True Positive (TP): Τα παραδείγματα τα οποία ανήκουν στη θετική κλάση και ταξινομήθηκαν στη θετική κλάση
- True Negative (TN): Τα παραδείγματα τα οποία ανήκουν στην αρνητική κλάση και ταξινομήθηκαν στην αρνητική κλάση
- False Positive (FP): Τα παραδείγματα τα οποία ανήκουν στην θετική κλάση αλλά ταξινομήθηκαν στην αρνητική κλάση
- False Negative (FN): Τα παραδείγματα τα οποία ανήκουν στην αρνητική κλάση αλλά ταξινομήθηκαν στην θετική κλάση

4.2.2 Μετρήσεις Απόδοσης

Με τη βοήθεια των τιμών του πίνακα σύγχυσης, μπορούμε να υπολογίσουμε αρκετές σημαντικές μετρήσεις απόδοσης. Συγκεκριμένα:

- Ορθότητα (Accuracy): $Ορθότητα = \frac{\text{αριθμός σωστών προβλέψεων}}{\text{συνολικός αριθμός προβλέψεων}} = \frac{TP+TN}{TP+TN+FP+FN}$
- Ακρίβεια (Precision): $Ακρίβεια = \frac{TP}{TP+FP}$
- Ευαισθησία (Recall/ Sensitivity): $Ευαισθησία = \frac{TP}{TP+FN}$
- F1-Score: $F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$
- Εξειδίκευση (Specificity): $Εξειδίκευση = \frac{TN}{TN+FP}$

4.3 Logistic Regression

Η λογιστική παλινδρόμηση (logistic regression) είναι μια ευρέως χρησιμοποιούμενη στατιστική μέθοδος για προβλήματα δυαδικής ταξινόμησης. Είναι ιδιαίτερα χρήσιμη σε τομείς όπως η ιατρική, τα οικονομικά και οι κοινωνικές επιστήμες, όπου η πρόβλεψη ενός δυαδικού αποτελέσματος βάσει διαφόρων προβλεπτικών μεταβλητών είναι κρίσιμη. Η λογιστική παλινδρόμηση έχει σχεδιαστεί για να προβλέπει την πιθανότητα ενός δυαδικού αποτελέσματος, κωδικοποιημένου ως 0 ή 1. Ο πυρήνας της λογιστικής παλινδρόμησης είναι η λογιστική συνάρτηση, γνωστή και ως συνάρτηση sigmoid, η οποία αντιστοιχεί οποιονδήποτε πραγματικό αριθμό σε μια τιμή μεταξύ 0 και 1.

$$\sigma(z) = \frac{1}{1 + e^z}$$

Αυτή η συνάρτηση εξασφαλίζει ότι οι προβλεπόμενες πιθανότητες βρίσκονται εντός της περιοχής $[0, 1]$.

Το μοντέλο λογιστικής παλινδρόμησης συνδυάζει τις προβλεπτικές μεταβλητές γραμμικά:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

- Σταθερά β_0 : οι βασικές λογαριθμικές πιθανότητες όταν όλες οι προβλεπτικές μεταβλητές είναι μηδενικές
- Συντελεστές κλίσης β_i : υποδεικνύουν την αλλαγή στις λογαριθμικές πιθανότητες για μια μονάδα αλλαγής στη προβλεπτικές μεταβλητή
- Λόγος πιθανοτήτων e^{β_i} : παρέχει μια διαισθητική ερμηνεία της επίδρασης κάθε προβλεπτικής μεταβλητής στην πιθανότητα του αποτελέσματος

Η λογιστική συνάρτηση στη συνέχεια μετατρέπει αυτόν τον γραμμικό συνδυασμό σε πιθανότητα:

$$P(Y = 1|X) = \sigma(z)$$

Όσο για τις παραμέτρους που χρησιμοποιήθηκαν ήταν οι εξής:

```
# Create the classifier  
lr = LogisticRegression(random_state=42, C=1, solver='liblinear')
```

Εικόνα 12: Παράμετροι του αλγορίθμου Logistic Regression

- Random State (int, default = None): Η παράμετρος αυτή ελέγχει την τυχειότητα του αλγορίθμου. Με τον ορισμό μιας συγκεκριμένης τιμής, (π.χ. random_state = 42), εξασφαλίζεται ότι τα αποτελέσματα μπορούν να επανεπιβεβαιωθούν επειδή οι τυχαίες διαδικασίες ελέγχονται και ξεκινούν από το ίδιο σημείο.
- C (float, default = 1.0): Η παράμετρος αυτή είναι το αντίστροφο της κανονικοποίησης, δηλαδή οι μικρότερες τιμές υποδηλώνουν ισχυρότερη κανονικοποίηση. Η κανονικοποίηση χρησιμοποιείται για την αποφυγή υπερπροσαρμογής.
- Solver (str, default = 'lbfgs'): Αυτή η παράμετρος χρησιμοποιείται για την συγκεκριμενοποίηση του αλγορίθμου για βελτιστοποίηση. Σε σύγκριση με τους υπόλοιπους αλγορίθμους ο 'liblinear' είχε την καλύτερη απόδοση και ταιριάζει περισσότερο σε σύνολα δεδομένων με μικρότερο αριθμό δειγμάτων.

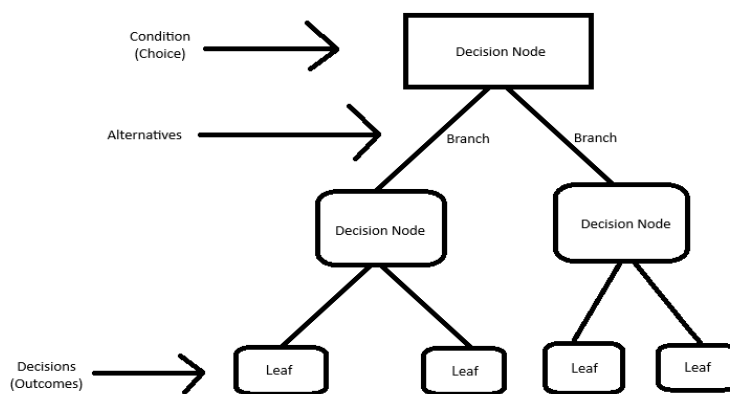
Τα αποτελέσματα των μετρήσεων έχουν ως εξής:

Πίνακας 2:
Αποτελέσματα μετρήσεων του Logistic Regression

Accuracy	Recall	F1-Score	Precision	Mean cross validation score
81%	85%	82%	79%	72%

4.4 Decision Tree

Τα δένδρα απόφασης (decision tree) είναι μια μέθοδος για ταξινομήσεις και παλινδρομήσεις σε μηχανική μάθηση και στατιστική. Χρησιμοποιούνται για την πρόβλεψη ενός τιμής μιας μεταβλητής μαθαίνοντας απλούς κανόνες απόφασης που προκύπτουν από τα χαρακτηριστικά των δεδομένων.



Εικόνα 13: Δομή του αλγορίθμου Δένδρων Απόφασης (Decision Tree)

Ενός φαίνεται και παραπάνω ένα δένδρο απόφασης περιλαμβάνει:

- Ρίζα (root node): Η ρίζα αντιπροσωπεύει το σύνολο των δεδομένων το οποίο στη συνέχεια χωρίζεται σε δυο ή περισσότερα ομοιογενή σύνολα. Στο σχήμα η ρίζα θα ήταν η συνθήκη-επιλογή (condition-choice)
- Κόμβους απόφασης (decision nodes): Είναι σημεία στα δένδρα απόφασης όπου γίνεται μια επιλογή βάσει μιας συγκεκριμένης τιμής χαρακτηριστικού. Αυτοί οι κόμβοι χωρίζουν τα δεδομένα σε δυο ή περισσότερα υποσύνολα, καθορίζοντας την πορεία που θα ακολουθήσει το δεδομένο μέσα στο δένδρο
- Κλαδιά (branches): Υποδιαίρέσεις του δένδρου που αντιπροσωπεύουν το αποτέλεσμα ενός διαχωρισμού σε έναν κόμβο
- Φύλλα (leaves): Κόμβοι που αντιπροσωπεύουν μια ταξινόμηση ή απόφαση

Όσο για ενός παραμέτρους που χρησιμοποιήθηκαν ήταν οι εξής:

Εικόνα 14: Παράμετροι του αλγορίθμου Decision Tree

-
- Figure 1: A complex decision tree for predicting the probability of a patient being a responder to treatment. The tree starts with a root node 'Test' and branches into 'Yes' and 'No' paths. Each node contains a box with patient characteristics (age, gpi, gpi2, gpi3, gpi4, gpi5, gpi6, gpi7, gpi8, gpi9, gpi10, gpi11, gpi12, gpi13, gpi14, gpi15, gpi16, gpi17, gpi18, gpi19, gpi20, gpi21, gpi22, gpi23, gpi24, gpi25, gpi26, gpi27, gpi28, gpi29, gpi30, gpi31, gpi32, gpi33, gpi34, gpi35, gpi36, gpi37, gpi38, gpi39, gpi40, gpi41, gpi42, gpi43, gpi44, gpi45, gpi46, gpi47, gpi48, gpi49, gpi50, gpi51, gpi52, gpi53, gpi54, gpi55, gpi56, gpi57, gpi58, gpi59, gpi60, gpi61, gpi62, gpi63, gpi64, gpi65, gpi66, gpi67, gpi68, gpi69, gpi70, gpi71, gpi72, gpi73, gpi74, gpi75, gpi76, gpi77, gpi78, gpi79, gpi80, gpi81, gpi82, gpi83, gpi84, gpi85, gpi86, gpi87, gpi88, gpi89, gpi90, gpi91, gpi92, gpi93, gpi94, gpi95, gpi96, gpi97, gpi98, gpi99, gpi100) and a box with the predicted probability of being a responder. The tree is highly detailed, with many branches and nodes, and is color-coded with orange and blue boxes.

Εικόνα 15: Γραφική αναπαράσταση του αλγορίθμου Decision Tree

Τα αποτελέσματα των μετρήσεων έχουν ως εξής:

Πίνακας 3:
Αποτελέσματα μετρήσεων του αλγορίθμου *Decision Tree*

Accuracy	Recall	F1-Score	Precision	Mean cross validation score
70%	75%	71%	68%	66%

4.5 Random Forest

Τα τυχαία δάση (random forest) είναι μια μέθοδος συλλογικής μάθησης που χρησιμοποιείται κυρίως σε προβλήματα ταξινόμησης και παλινδρόμησης. Ο τρόπος με τον οποίο λειτουργεί είναι κατασκευάζοντας μια πληθώρα δένδρων απόφασης κατά τη διάρκεια ενός εκπαίδευσης και εξάγοντας τον επικρατέστερο των κατηγοριών (ταξινόμηση) ή τον μέσο όρο ενός πρόβλεψης (παλινδρόμηση) των επιμέρους δένδρων.

Τα βήματα που ακολουθεί ο αλγόριθμος είναι τα εξής:

1. Από το αρχικό σύνολο εκπαίδευσης, λαμβάνονται τυχαία N δείγματα με αντικατάσταση για τη δημιουργία πολλαπλών υποσυνόλων. Κάθε υποσύνολο έχει το ίδιο μέγεθος με το αρχικό σύνολο εκπαίδευσης.
2. Για κάθε υποσύνολο, αναπτύσσεται ένα δέντρο απόφασης. Ωστόσο, σε κάθε κόμβο, αντί να εξετάζονται όλα τα χαρακτηριστικά, επιλέγεται ένα τυχαίο υποσύνολο m χαρακτηριστικών (όπου το m είναι συνήθως $\sqrt{p}$ για την ταξινόμηση ή $p/3$ για την παλινδρόμηση, και το p είναι ο συνολικός αριθμός χαρακτηριστικών). Το δέντρο αναπτύσσεται μέχρι να φτάσει στο μέγιστο βάθος του ή τα φύλλα να περιέχουν λιγότερα από τον ελάχιστο αριθμό δειγμάτων που απαιτούνται για τη διαίρεση.
3. Για την ταξινόμηση, κάθε δέντρο στο δάσος “ψηφίζει” για μια κατηγορία, και η κατηγορία με ενός περισσότερες ψήφους επιλέγεται ως η τελική πρόβλεψη. Για την παλινδρόμηση, ο μέσος όρος όλων των προβλέψεων των δέντρων λαμβάνεται ως το τελικό αποτέλεσμα.

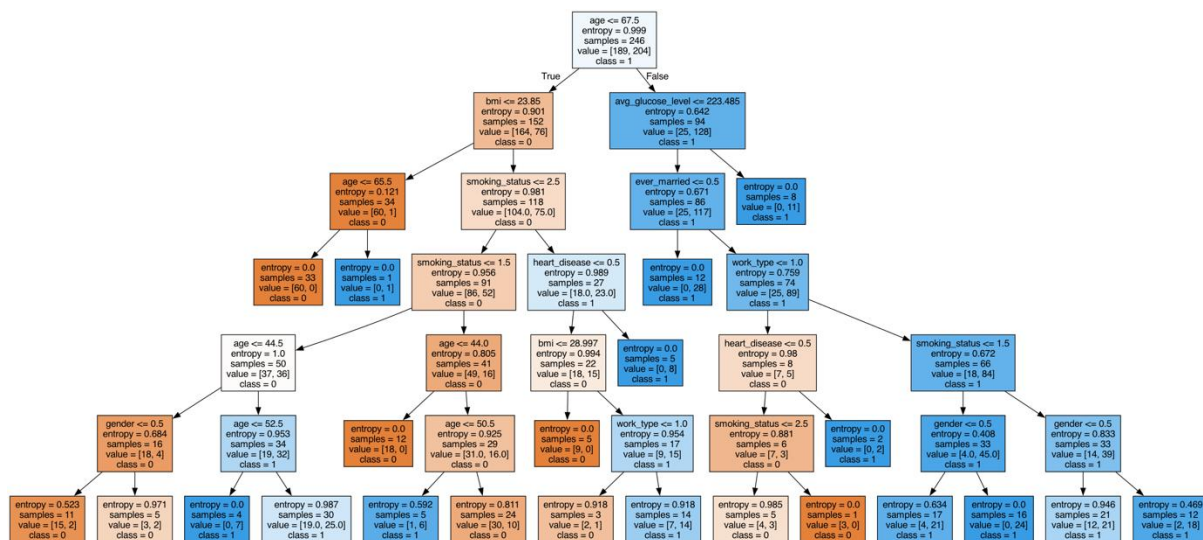
Κάποια από τα πλεονεκτήματα των τυχαίων δασών περιλαμβάνουν αυξημένη ορθότητα λόγω ενός συλλογικής φύσης ενός, διαχείριση μεγάλων συνόλων δεδομένων επειδή τα δένδρα κατασκευάζονται ανεξάρτητα το ένα από το άλλο, μπορούν να εκτελούνται παράλληλα και μπορούν να παρέχουν μια εκτίμηση ενός σημαντικότητας των χαρακτηριστικών, η οποία είναι αναγκαία για την καλύτερη κατανόηση των δεδομένων. Βασικό μειονεκτήματα ενός είναι το πόσο πολύπλοκα και υπολογιστικά απαιτητικά μπορούν να γίνουν με μεγάλα σύνολα δεδομένων και πολλά δένδρα.

Όσο για ενός παραμέτρους που χρησιμοποιήθηκαν ήταν οι εξής:

```
# Create a random forest classifier  
rf = RandomForestClassifier(random_state=42, n_estimators= 120, criterion='entropy', max_depth= 6)
```

Εικόνα 16: Παράμετροι του αλγορίθμου Τυχαίου Δάσους (Random Forest)

- `n_estimators` (int, default = 100): Το σύνολο των δένδρων. Σε αυτήν την περίπτωση υπάρχουν 120 δένδρα.
- `Criterion` (str, default = 'gini'): Η συνάρτηση για τη μέτρηση ενός ποιότητας ενός διαχωρισμού. Τα υποστηριζόμενα κριτήρια είναι το 'gini' και το 'entropy' για το κέρδος πληροφοριών. Ουσιαστικά αυτές οι συναρτήσεις καθορίζουν το πως διαχωρίζονται οι κόμβοι των δένδρων αποφάσεων. Το κριτήριο 'entropy' επιλέγηκε διότι είχε καλύτερες αποδόσεις.



Εικόνα 17: Γραφική αναπαράσταση του αλγορίθμου Random Forest

Τα αποτελέσματα των μετρήσεων έχουν ως εξής:

Πίνακας 4:

Αποτελέσματα μετρήσεων του αλγορίθμου Random Forest

Accuracy	Recall	F1-Score	Precision	Mean cross validation score
77%	91%	80%	71%	74%

4.6 Boosting

Το boosting είναι μια τεχνική συνόλου (ensemble learning) στη μηχανική μάθηση που έχει ως βασικό στόχο τη δημιουργία ενός ισχυρού ταξινομητή από μια σειρά ασθενών ταξινομητών

- Ασθενής ταξινομητής: ένας ασθενής ταξινομητής είναι ένα μοντέλο το οποίο έχει λίγο καλύτερες επιδόσεις από το να μαντεύει τυχαία. Δεν έχει υψηλή ορθότητα και δεν είναι αρκετά δυνατό από μόνο του.
- Ισχυρός ταξινομητής: ένας ισχυρός ταξινομητής είναι ένας συνδυασμός πολλαπλών ασθενών ταξινομητών οι οποίοι λειτουργούν ταυτόχρονα ώστε να αυξηθεί η επίδοση.

Στην αρχή τα σημεία των δεδομένων (data points) έχουν την ίδια βαρύτητα. Μετά την εκπαίδευση του κάθε ασθενή ταξινομητή, η βαρύτητα των λανθασμένα ταξινομημένων δεδομένων αυξάνεται, κάνοντας τον επόμενο ταξινομητή να εστιάσει περισσότερο σε εκείνα τα σημεία.

Η διαδικασία αυτή επαναλαμβάνεται αρκετές φορές. Το τελικό μοντέλο που προκύπτει είναι ένας σταθμισμένος συνδυασμός όλων των ασθενών ταξινομητών, που έχει ως αποτέλεσμα τη δημιουργία ενός πιο ακριβή και ισχυρού μοντέλου που διορθώνει τα λάθη των επιμέρους στοιχείων του.

4.6.1 XGBoost

Ο XGBoost κατασκευάζει δέντρα απόφασης διαδοχικά, όπου κάθε νέο δέντρο προσπαθεί να διορθώσει τα λάθη που έκαναν τα προηγούμενως κατασκευασμένα δέντρα

1. Gradient Boosting Framework

- Το gradient boosting περιλαμβάνει την εκπαίδευση ενός συνόλου ασθενών μοντέλων (συνήθως δέντρα απόφασης) με διαδοχικό τρόπο. Κάθε δέντρο εκπαιδεύεται για να προβλέπει τα σφάλματα του συνδυασμένου συνόλου όλων των άλλων δέντρων.

2. Συνάρτηση στόχου

- Η συνάρτηση στόχου στο XGBoost περιλαμβάνει δύο μέρη: μια συνάρτηση απώλειας που μετρά πόσο καλά το μοντέλο ταιριάζει με τα δεδομένα εκπαίδευσης και έναν όρο κανονικοποίησης που ελέγχει την πολυπλοκότητα του μοντέλου για να αποτρέψει την υπερπροσαρμογή
- Η γενική μορφή της συνάρτησης στόχου είναι:

$$Obj = \sum_i L(y_i, \hat{y}_i) + \sum_k \Omega(f_k)$$

3. Προσθετική μάθηση

- Ο XGBoost κατασκευάζει δέντρα με προσθετικό τρόπο. Σε κάθε βήμα t , προστεται ένα νέο δέντρο f_t για να ελαχιστοποιήσει τη συνάρτηση στόχου:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$$

Εδώ το $\hat{y}_i^{(t)}$ είναι η προβλεψη για το i δειγμα στο βημα t .

4. Gradient descent

- Τα σφάλματα είναι οι αρνητικές παράγωγοι της συνάρτησης απώλειας ως προς τις προβλέψεις. Ο XGBoost χρησιμοποιεί gradient descent για να προσαρμόσει το νέο δένδρο σε αυτά τα υπολειπόμενα σφάλματα.

5. Δομή δένδρου

- Ο XGBoost χρησιμοποιεί ένα μοντέλο δένδρου απόφασης όπου κάθε εσωτερικός κόμβος αντιπροσωπεύει ένα χαρακτηριστικό και μια συνθήκη διαχωρισμού, και κάθε φύλλο αντιπροσωπεύει μια προβλεπόμενη τιμή.
- Το δένδρο κατασκευάζεται για την μεγιστοποίηση του κέρδους, το οποίο μετρά τη βελτίωση της συνάρτησης στόχου από την προσθήκη ενός διαχωρισμού

6. Κανονικοποίηση

- Ο XGBoost περιλαμβάνει κανονικοποίηση $L1$ (Lasso) και $L2$ (Ridge) για να ελέγξει την πολυπλοκότητα του μοντέλου, αποτρέποντας την υπερπροσαρμογή με την επιβολή ποινών για τα μεγάλα βάρη.
- Ο όρος κανονικοποίησης $\Omega(f_k)$ μπορεί να γραφτεί ως:

$$\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$$

Όπου T είναι ο αριθμός των φύλλων στο δένδρο, γ είναι η παραμετρος κανονικοποιησης για τον αριθμο των φυλλων και λ είναι η παραμετρος κανονικοποιησης για τα βαρη των φυλλων w_j .

Μερικά από τα πλεονεκτήματα που έχει ο XGBoost περιλαμβάνουν καλύτερες επιδόσεις και ταχύτητα εκπαίδευσης συγκριτικά με άλλους αλγορίθμους, μπορεί να διαχειριστεί ελλείπουσες τιμές και να εκτελέσει cross – validation χωρίς την ανάγκη ξεχωριστού κώδικα για αυτή τη διαδικασία. Κάποια από τα μειονεκτήματα είναι το πόσο περίπλοκες είναι οι παράμετροι του για την καλύτερη επίδοση του, ότι χρησιμοποιεί περισσότερη μνήμη για μεγάλα σύνολα δεδομένων και με τις ενσωματωμένες λειτουργίες που προσφέρει μπορεί ακόμα να προκύψουν θέματα υπερπροσαρμογής ειδικά σε μικρότερα σύνολα δεδομένων.

Όσο για τις παραμέτρους που χρησιμοποιήθηκαν ήταν οι εξής:

```
# Create the classifier
xgb = XGBClassifier(random_state=42, max_depth=4)
```

Εικόνα 18: Παράμετροι του αλγορίθμου XGBoost

Τα αποτελέσματα των μετρήσεων:

Πίνακας 5:
Αποτελέσματα μετρήσεων του αλγορίθμου XGBoost

Accuracy	Recall	F1-Score	Precision	Mean cross validation score
72%	81%	74%	68%	73%

4.6.2 RUSBoost

1. Αρχική φάση

- Αρχικοποίηση των βαρών στα δείγματα εκπαίδευσης. Αν υπάρχουν N δείγματα εκπαίδευσης, το κάθε ένα έχει αρχικό βάρος:

$$w_i = \frac{1}{N}, \text{ για όλα τα } i = 1, 2, \dots, N$$

2. Random Under-Sampling

- Εφαρμόζεται η τεχνική random under-sampling στην πλειοψηφούσα κλάση για τη δημιουργία ενός εξισορροπημένου συνόλου δεδομένων D_t .

3. Εκπαίδευση ασθενών ταξινομητών

- Γίνεται η εκπαίδευση των ασθενών ταξινομητών $h_t(x)$ στο ισοροπημένο σύνολο δεδομένων D_t .

4. Υπολογισμός σφάλματος

- Υπολογισμός του σφάλματος e_t του ασθενή ταξινομητή στο αρχικό σύνολο δεδομένων:

$$e_t = \frac{\sum_{i=1}^N w_i^{(t)} \cdot I(h_t(x_i) \neq y_i)}{\sum_{i=1}^N w_i^{(t)}},$$

όπου I η συνάρτηση δείκτη που παίρνει την τιμή 1 αν η συνθήκη είναι σωστή και 0 σε κάθε άλλη περίπτωση.

5. Υπολογισμός Alpha (Βάρος ταξινομητή)

- Υπολογισμός του βάρους a_t του ασθενή ταξινομητή $h_t(x)$:

$$a_t = \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right).$$

6. Ενημέρωση των βαρών

- Ενημέρωση των βαρών των δειγμάτων εκπαίδευσης:

$$w_i^{(t+1)} = w_i^{(t)} \cdot \exp(a_t \cdot I(h_t(x_i) \neq y_i))$$

- Κανονικοποίηση των βαρών ώστε να δίνουν άθροισμα 1:

$$w_i^{(t+1)} = \frac{w_i^{(t+1)}}{\sum_{j=1}^N w_j^{(t+1)}}$$

7. Τελικό μοντέλο

- Ο τελικός ισχυρός ταξινομητής $H(x)$ είναι μια σταθμισμένη πλειοψηφική ψήφος των ασθενών ταξινομητών:

$$H(x) = \text{sign} \left(\sum_{t=1}^T a_t h_t(x) \right).$$

Πλεονεκτήματα του RUSBoost είναι η διαχείριση των μη ισορροπημένων δεδομένων και η απλότητα και ευκολία κατανόησης της διαδικασίας του. Τα μειονεκτήματα του είναι απώλεια πιθανής χρήσιμης πληροφορίας λόγω της τεχνικής random under-sampling και η αστάθεια των αποτελεσμάτων.

Όσο για τις παραμέτρους που χρησιμοποιήθηκαν ήταν οι εξής:

```
# Create the classifier
rusb = RUSBoostClassifier(n_estimators=200, random_state=42, algorithm='SAMME.R')
```

Εικόνα 19: Παράμετροι του αλγορίθμου RUSBoost

Τα αποτελέσματα των μετρήσεων:

Πίνακας 6:

Αποτελέσματα μετρήσεων του αλγορίθμου RUSBoost

Accuracy	Recall	F1-Score	Precision	Mean cross validation score
66%	75%	69%	63%	67%

4.6.3 AdaBoost

1. Αρχική φάση

- Αρχικοποίηση των βαρών στα δείγματα εκπαίδευσης. Αν υπάρχουν N δείγματα εκπαίδευσης, το κάθε ένα έχει αρχικό βάρος: $w_i = \frac{1}{N}$

2. Εκπαίδευση ασθενών ταξινομητών

Για κάθε επανάληψη t (από 1 έως T):

- Γίνεται η εκπαίδευση ενός ασθενή ταξινομητή $h_t(x)$ χρησιμοποιώντας τα τρχοντα βαρη w_i

3. Υπολογισμός σφάλματος

- Υπολογισμός του ποσοστού σφάλματος e_t του ασθενή ταξινομητή $h_t(x)$:

$$e_t = \frac{\sum_{i=1}^N w_i \cdot \Pi(h_t(x_i) \neq y_i)}{\sum_{i=1}^N w_i}$$

Όπου Π είναι η συνάρτηση δείκτη που είναι 1 αν $h_t(x_i) \neq y_i$ αλλιώς είναι 0, x_i είναι η εισοδος και y_i είναι η πραγματική ετικέτα

4. Υπολογισμός Alpha (Βάρος ταξινομητή)

- Υπολογισμός του βάρους a_t του ασθενή ταξινομητή $h_t(x)$:

$$a_t = \frac{1}{2} \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right)$$

5. Ενημέρωση των βαρών

- Ενημέρωση των βαρών των δειγμάτων εκπαίδευσης:

$$w_i \leftarrow w_i \cdot \exp(a_t \cdot I(h_t(x_i) \neq y_i))$$

- Κανονικοποίηση των βαρών ώστε να δίνουν άθροισμα 1:

$$w_i \leftarrow \frac{w_i}{\sum_{j=1}^N w_j}$$

6. Τελική υπόθεση

- Ο τελικός ισχυρός ταξινομητής $H(x)$ είναι μια σταθμισμένη πλειοψηφική ψήφος των ασθενών ταξινομητών:

$$H(x) = \text{sign} \left(\sum_{t=1}^T a_t \cdot h_t(x) \right).$$

Τα πλεονεκτήματα του AdaBoost περιλαμβάνουν απλότητα και εύκολη κατανόηση στη διαδικασία που ακολουθεί, βελτιωμένη ορθότητα λόγω του συνδυασμού των ασθενών ταξινομητών σε έναν ισχυρότερον και δεν χρειάζεται χειροκίνητη επεξεργασία των παραμέτρων για να λειτουργήσει σωστά. Από την άλλη κάποια από τα μειονεκτήματα περιλαμβάνουν ευαισθησία σε θόρυβο λόγω της λανθασμένης διαδικασίας καθαρισμού δεδομένων που μπορεί να οδηγήσει σε υπερπροσαρμογή και χαμηλές επιδόσεις σε δεδομένα που δεν είναι ισορροπημένα.

Όσο για τις παραμέτρους που χρησιμοποιήθηκαν ήταν οι εξής:

```
# Create the classifier  
ada = AdaBoostClassifier(algorithm='SAMME', random_state=42)
```

Εικόνα 20: Παράμετροι του αλγορίθμου ADABOOST

- Algorithm (default='SAMME.R'): ο αλγόριθμος boosting που επιλέγεται

Τα αποτελέσματα των μετρήσεων:

Πίνακας 7:
Αποτελέσματα μετρήσεων του αλγορίθμου AdaBoost

Accuracy	Recall	F1-Score	Precision	Mean cross validation score
74%	87%	77%	69%	75%

5 Αποτελέσματα

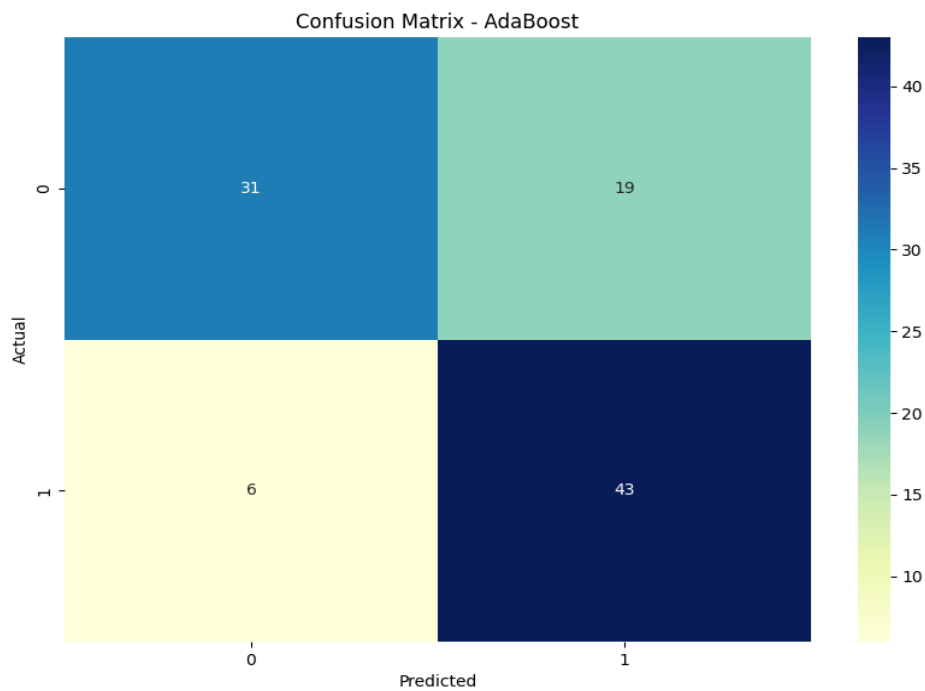
5.1 Σύγκριση Απόδοσης των Μοντέλων

Πίνακας 8:
Αποτελέσματα μετρήσεων όλων των αλγορίθμων

Αποτελέσματα				
	Accuracy	Recall	F1-Score	Precision
AdaBoost	74%	87%	77%	69%
XGBoost	72%	81%	74%	68%
Random Forest	77%	91%	80%	71%
Decision Tree	70%	75%	71%	68%
Logistic Regression	81%	85%	82%	79%
RUSBoost	66%	75%	69%	63%

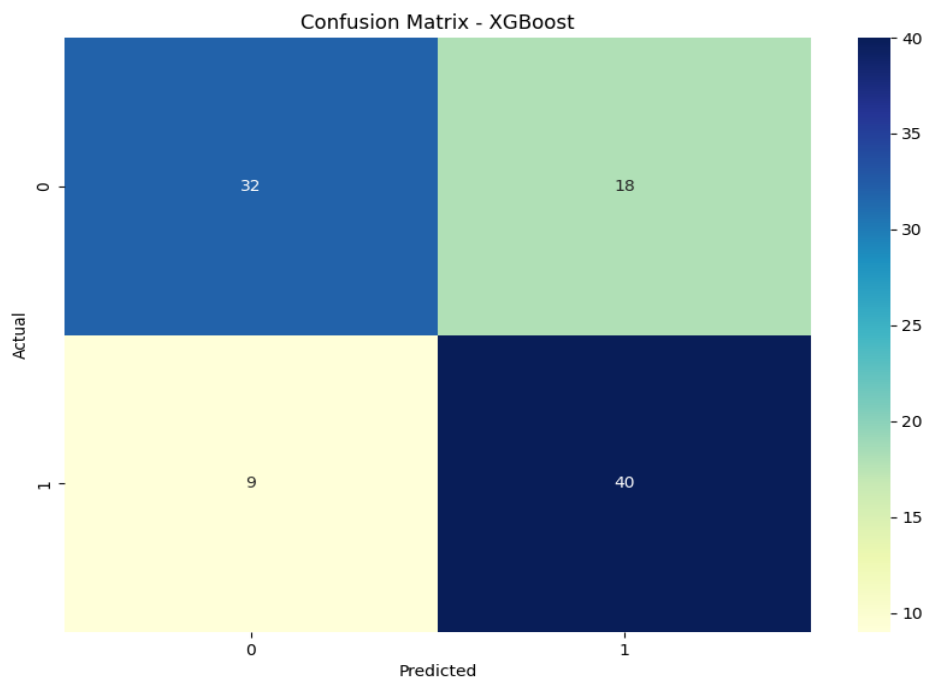
Στον Πίνακα 10 φαίνεται ότι ο Logistic Regression είχε τις καλύτερες επιδόσεις με ορθότητα 81%, ακρίβεια 79%, ευαισθησία 85% και F1-Score 82%. Με τον δεύτερο καλύτερο να είναι ο Random Forest με ορθότητα 77%, ακρίβεια 71%, ευαισθησία 91% και F1-Score 80%. Οι δυο αλγόριθμοι είναι εξίσου καλοί αλλά θεωρώ ότι ο Random Forest είναι ελάχιστα καλύτερος ως προς τη διαχείριση αυτών των περίπλοκων ιατρικών δεδομένων. Σε περίπτωση που αντιμετωπίζαμε δεδομένα τα οποία δεν είχαν τόσο περιπλοκά μοτίβα και συσχετίσεις μεταξύ τους ο Logistic Regression θα ήταν ο ιδανικός αλγόριθμος για τη δημιουργία του μοντέλου πρόβλεψης. Ακολουθούν οι πίνακες σύγκρισης για κάθε αλγόριθμο:

ADABOOST



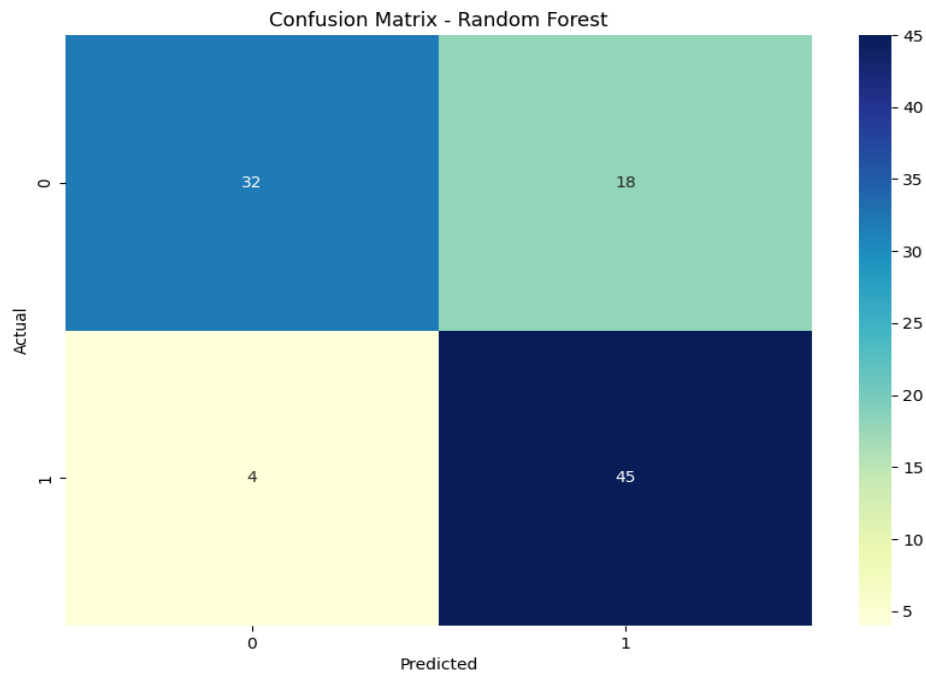
Εικόνα 21: Πίνακας σύγκρισης του αλγορίθμου ADABOOST

XGBoost



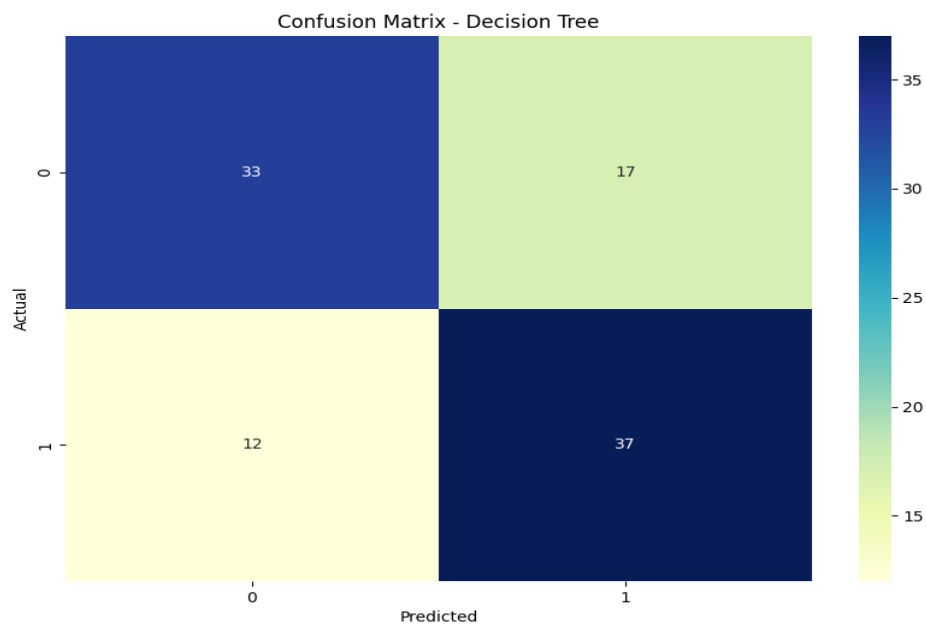
Εικόνα 22: Πίνακας σύγκρισης του αλγορίθμου XGBoost

Random Forest



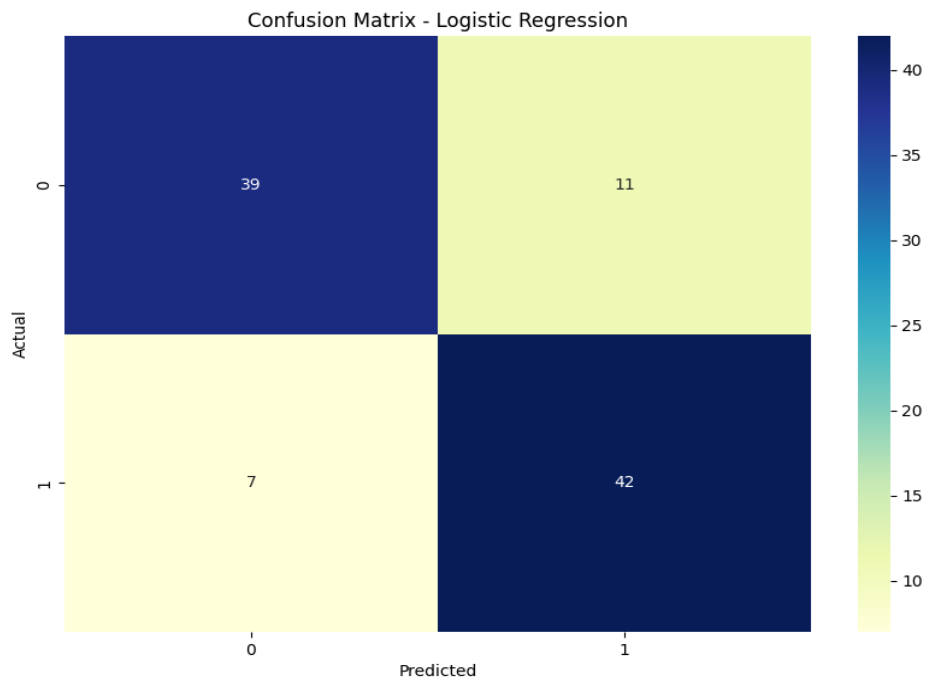
Εικόνα 23: Πίνακας σύγχυσης του αλγορίθμου Random Forest

Decision Tree



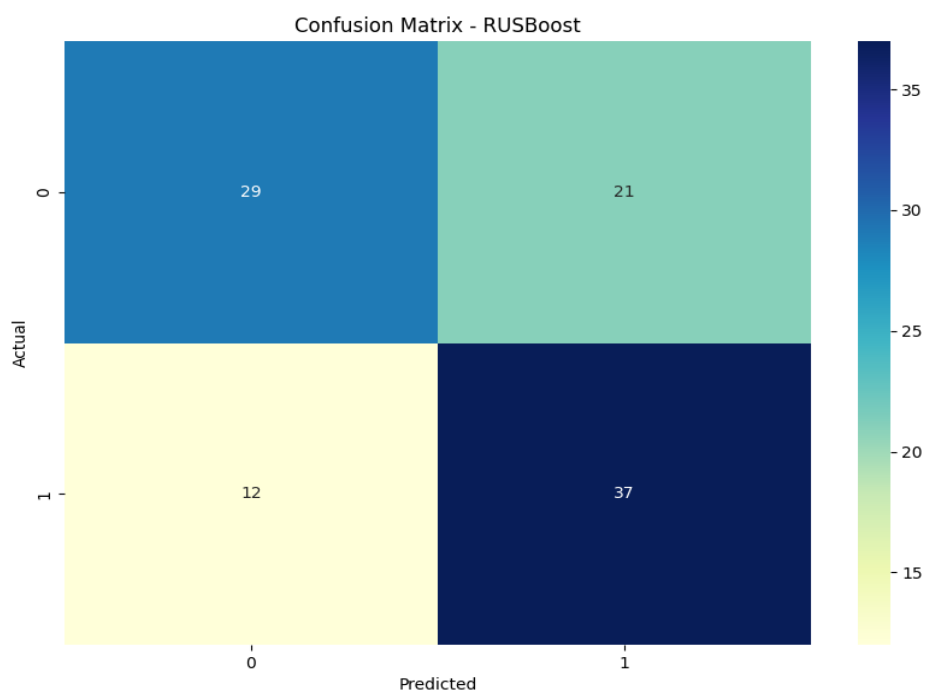
Εικόνα 24: Πίνακας σύγχυσης του αλγορίθμου Decision Tree

Logistic Regression



Εικόνα 25: Πίνακας σύγκρισης του αλγορίθμου Logistic Regression

RUSBoost

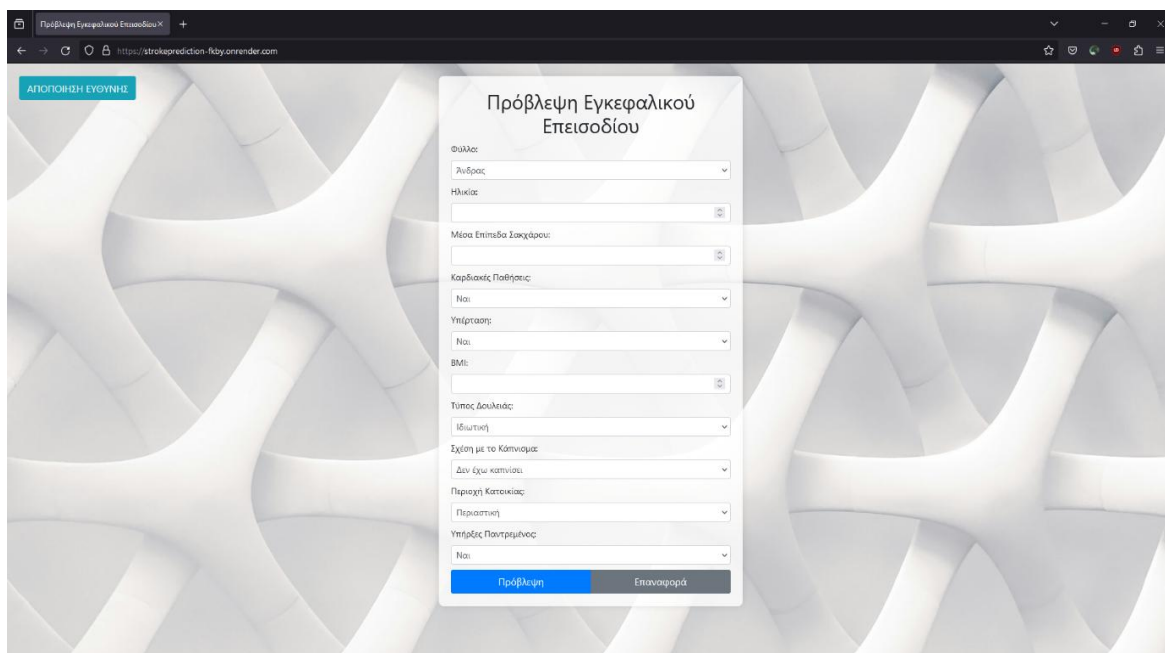


Εικόνα 26: Πίνακας σύγκρισης του αλγορίθμου RUSBoost

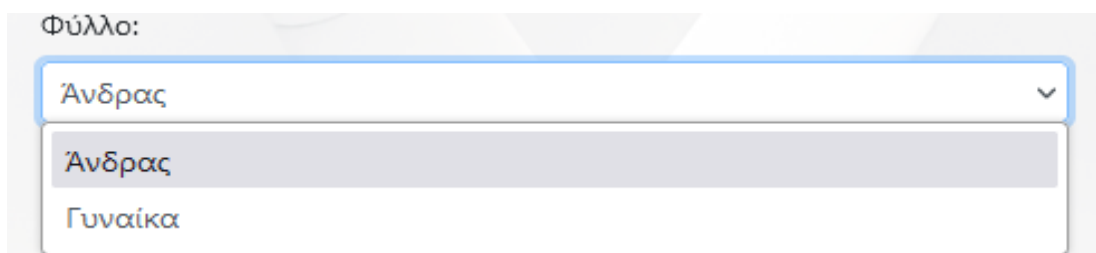
6 Ανάπτυξη Εφαρμογής

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε html, CSS, jQuery και bootstrap. Για την δημιουργία της ιστοσελίδας που είναι προσβάσιμη από όλους χρησιμοποιήθηκε flask και για το hosting της ιστοσελίδας χρησιμοποιήθηκε το render[19].

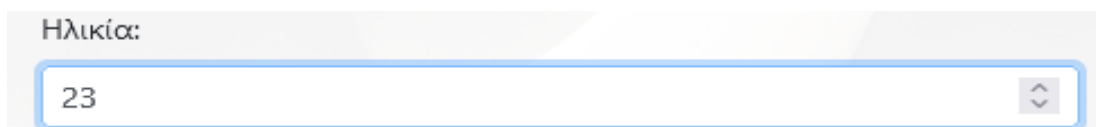
Ακολουθούν κάποιες εικόνες από την εφαρμογή:



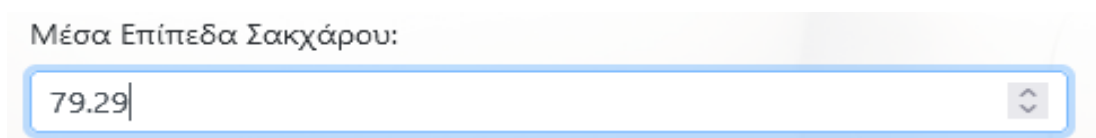
Εικόνα 27: Η εφαρμογή πρόβλεψης εγκεφαλικού επεισοδίου



Εικόνα 28: Επιλογή φύλλου



Εικόνα 29: Εισαγωγή ηλικίας



Εικόνα 30: Εισαγωγή μέσου επιπέδου σακχάρου

Καρδιακές Παθήσεις:

Ναι

Ναι

Όχι

Εικόνα 31: Επιλογή καρδιακών παθήσεων

Υπέρταση:

Ναι

Ναι

Όχι

Εικόνα 32: Επιλογή κατάστασης της πίεσης

BMI:

23.86

Εικόνα 33: Εισαγωγή Δείκτη Μάζας Σώματος

Τύπος Δουλειάς:

Ιδιωτική

Ιδιωτική

Αυτοαπασχολούμενος

Παιδί

Δημόσιος Υπάλληλος

Δεν δουλεύω

Εικόνα 34: Επιλογή τύπου δουλειάς

Σχέση με το Κάπνισμα:

Δεν έχω καπνίσει

Δεν έχω καπνίσει

Άγνωστο

Υπήρξα καπνιστής

Είμαι καπνιστής

Εικόνα 35: Επιλογή κατάστασης καπνίσματος

Περιοχή Κατοικίας:

Περιαστική

Περιαστική

Αστική

Εικόνα 36: Επιλογή περιοχής κατοικίας

Υπήρξε Παντρεμένος:

Ναι

Ναι

Όχι

Εικόνα 37: Επιλογή αν ο χρήστης έχει παντρευτεί ποτέ

ΑΠΟΠΟΙΗΣΗ ΕΥΘΥΝΗΣ

ΑΠΟΠΟΙΗΣΗ ΕΥΘΥΝΗΣ
ΑΥΤΗ Η ΙΣΤΟΣΕΛΙΔΑ ΔΕΝ ΠΑΡΕΧΕΙ
ΙΑΤΡΙΚΗ ΣΥΜΒΟΥΛΗ. Οι πληροφορίες
που περιέχονται σε αυτόν τον ιστότοπο
προορίζονται μόνο για ενημερωτικούς
σκοπούς. Καμία ένδειξη σε αυτόν τον
ιστότοπο δεν προορίζεται να
υποκαταστήσει επαγγελματικές ιατρικές
συμβουλές, διάγνωση ή θεραπεία.
Πάντα να ζητάτε τη συμβουλή του
γιατρού σας ή άλλου ειδικευμένου
παρόχου υγειονομικής περίθαλψης για
οποιαδήποτε ερώτηση μπορεί να έχετε
σχετικά με μια ιατρική κατάσταση ή
θεραπεία και μην αγνοείτε ποτέ την
επαγγελματική ιατρική συμβουλή ή
καθυστερείτε να την αναζητήσετε για
κάτι που έχετε διαβάσει σε αυτόν τον
ιστότοπο.

Εικόνα 38: Κείμενο αποποίησης ευθύνης

Αποτέλεσμα Πρόβλεψης



Δεν υπάρχει κίνδυνος εγκεφαλικού

Κλείσιμο

Εικόνα 39: Εμφάνιση αποτελέσματος πρόβλεψης με τη μορφή "pop-up" (χωρίς κίνδυνο εγκεφαλικού)

Αποτέλεσμα Πρόβλεψης



Υπάρχει κίνδυνος εγκεφαλικού

Κλείσιμο

Εικόνα 40: Εμφάνιση αποτελέσματος πρόβλεψης με τη μορφή "pop-up" (με κίνδυνο εγκεφαλικού)

7 Συμπεράσματα

Στη παρούσα έρευνα δοκιμάστηκαν διάφοροι αλγόριθμοι μηχανικής μάθησης για τη δημιουργία ενός μοντέλου με σκοπό την πρόβλεψη ενός εγκεφαλικού επεισοδίου. Επειδή μέχρι και σήμερα τα εγκεφαλικά επεισόδια αποτελούν μια από τις συχνότερες αιτίες θανάτου παγκοσμίως είναι πιο αναγκαία από ποτέ αυτού του είδους τα διαγνωστικά και προβλεπτικά εργαλεία. Ωστόσο, οι παραδοσιακοί τρόποι, αν και πολύ σημαντικοί στη σωστή και άμεση διάγνωση, αρκετές φορές δεν λαμβάνουν υπόψη τους όλους τους παράγοντες που μπορούν να προκαλέσουν ένα εγκεφαλικό επεισόδιο δημιουργώντας έτσι την ανάγκη μιας πιο στοχευμένης προσέγγισης.

Λαμβάνοντας υπόψη τα δεδομένα των ασθενών, όπως την ηλικία, τον Δείκτη Μάζας Σώματος (BMI) και την υπέρταση, αυτή η έρευνα κατάφερε να αναπτύξει διάφορα προβλεπτικά μοντέλα με τη χρήση εναλλακτικών αλγορίθμων. Έγινε διεξοδική ανάλυση των δεδομένων και των χαρακτηριστικών των ασθενών με σκοπό την καλύτερη κατανόηση και ερμηνεία τους. Επίσης, δοκιμάστηκαν διάφοροι αλγόριθμοι όπως ο Logistic Regression, Decision Tree, ADA-Boost και ο Random Forest. Κάθε αλγόριθμος αναλύθηκε διεξοδικά και αξιολογήθηκε ως προς την ορθότητα (accuracy), την ακρίβεια (precision), την ευαισθησία (recall) και το F1-Score του, προκειμένου να εντοπιστεί ο καταλληλότερος. Οι αλγόριθμοι Random Forest και Logistic Regression έδειξαν αρκετά καλά και σταθερά αποτελέσματα στις μετρικές επίδοσης τους, το οποίο είναι σημαντικό διότι μας δείχνει ότι η μηχανική μάθηση μπορεί να βελτιώσει σημαντικά την ακρίβεια και την αξιοπιστία των προβλέψεων. Αυτού του είδους η βελτίωση στην προγνωστική ακρίβεια προσφέρει πολύτιμη υποστήριξη για την έγκαιρη παρέμβαση, επιτρέποντας στους ειδικούς να εντοπίζουν και να εφαρμόζουν πιο έγκαιρα προληπτικά μέτρα.

Μέσω της έρευνας διαπιστώνεται η εξαιρετική σημασία που έχουν οι τεχνικές προεπεξεργασίας των δεδομένων όπως είναι η εξισορρόπηση των δεδομένων και η κωδικοποίηση των χαρακτηριστικών, οι οποίες είναι κρίσιμες στη βελτίωση της απόδοσης των μοντέλων. Η χρήση τεχνικών όπως η Random Undersampling και η κωδικοποίηση ετικετών διασφάλισε ότι τα μοντέλα εκπαιδεύτηκαν σε ισορροπημένα σύνολα δεδομένων, αυξάνοντας με αυτόν τον τρόπο περαιτέρω την ακρίβεια τους.

Η μελλοντική έρευνα θα πρέπει να επικεντρωθεί στη βελτίωση αυτών των μοντέλων με την ενσωμάτωση μεγαλύτερης ποικιλίας συνόλων δεδομένων και την εξερεύνηση της ενσωμάτωσης επιπρόσθετων παραγόντων κινδύνου, όπως πληροφορίες για το γενετικό υλικό και τον τρόπο ζωής κάθε ασθενή, για περαιτέρω βελτίωση της προγνωστικής ακρίβειας. Επιπλέον, η ανάπτυξη εργαλείων που επιτρέπουν στους κλινικούς ιατρούς να ερμηνεύουν και να χρησιμοποιούν εύκολα τις προβλέψεις των μοντέλων στην καθημερινή τους πρακτική είναι κρίσιμη για την πρακτική εφαρμογή αυτών των τεχνολογιών.

Preprocessing

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from imblearn.under_sampling import RandomUnderSampler
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from xgboost import XGBClassifier
from imblearn.ensemble import RUSBoostClassifier
import matplotlib.pyplot as plt
import seaborn as sns

# Load the data
data = pd.read_csv('/Users/gpbiz/Desktop/Project/healthcare-dataset-stroke-data.csv')

# Count the occurrences of each class in the 'stroke' column
stroke_counts_a = data['stroke'].value_counts()

# Plot the distribution of the 'stroke' column before preprocessing
plt.figure(figsize=(8, 6))
stroke_counts_a.plot(kind='bar', color=['blue', 'orange'])
plt.title('Distribution of stroke cases before preprocessing')
plt.xlabel('Stroke')
plt.ylabel('Count')
plt.xticks(ticks=[0, 1], labels=['No Stroke', 'Stroke'], rotation=0)
plt.grid(axis='y')

# Annotate the bars with the counts
for i in range(len(stroke_counts_a)):
    plt.text(i, stroke_counts_a[i] + 50, str(stroke_counts_a[i]), ha='center', va='bottom')

# Remove grid lines
plt.grid(False)
plt.show()

# Drop the 'id' column
data = data.drop('id', axis=1)

# Fill the missing values in the 'bmi' column with the mean value
mean_bmi = data['bmi'].mean()
data['bmi'] = data['bmi'].fillna(mean_bmi)

# Display rows where BMI is greater than 60
bmi_above_60 = data[data['bmi'] > 60]
print(f"\nRows where BMI is greater than 60: {bmi_above_60}")
```

```

# Identifying and handling outliers in the BMI column
Q1 = data['bmi'].quantile(0.25)
Q3 = data['bmi'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Filtering out the outliers
data = data[(data['bmi'] >= lower_bound) & (data['bmi'] <= upper_bound)]

print(data.head())

# One-hot encode the categorical columns
# categorical_columns = ['gender', 'ever_married', 'work_type', 'Residence_type', 'smoking_status']
# data = pd.get_dummies(data, columns=categorical_columns)

# Label encode the categorical columns
categorical_columns = ['gender', 'ever_married', 'work_type', 'Residence_type', 'smoking_status']
label_encoders = {}
for column in categorical_columns:
    le = LabelEncoder()
    data[column] = le.fit_transform(data[column])
    label_encoders[column] = le

# Split the data into features and target variable
X = data.drop('stroke', axis=1)
y = data['stroke']

# Print class distribution before undersampling
print("Class distribution before undersampling:")
print(y.value_counts())

# Undersampling
rus = RandomUnderSampler(random_state=1)
X_res, y_res = rus.fit_resample(X, y)

# Print class distribution after undersampling
print("\nClass distribution after undersampling:")
print(y_res.value_counts())

# Train Models on Selected Features
# Split the data into training and validation sets
X_train, X_val, y_train, y_val = train_test_split(X_res, y_res, test_size=0.2, random_state=42)

```

```

# Define models
models = [
    RandomForestClassifier(random_state=42, n_estimators=120, criterion='entropy', max_depth=6),
    DecisionTreeClassifier(random_state=42, max_depth=7),
    AdaBoostClassifier(algorithm='SAMME', random_state=42),
    RUSBoostClassifier(n_estimators=200, random_state=42, algorithm='SAMME.R'),
    XGBClassifier(random_state=42, max_depth=4),
    LogisticRegression(random_state=42, C=1, solver='liblinear')
]

# Initialize array to store aggregated feature importances
feature_importances = np.zeros(len(X.columns))

# Train models on the selected features and collect feature importances
for model in models:
    model.fit(X_train, y_train)
    if hasattr(model, 'feature_importances_'):
        importances = model.feature_importances_
    elif hasattr(model, 'coef_'):
        importances = np.abs(model.coef_[0])
    feature_importances += importances

# Average the feature importances
feature_importances /= len(models)

# Normalize the averaged importances from 1 to 10
min_importance = feature_importances.min()
max_importance = feature_importances.max()
normalized_importances = 1 + 9 * (feature_importances - min_importance) / (max_importance - min_importance)

# Plot normalized feature importances for all features
plt.figure(figsize=(10, 6))
sns.barplot(x=normalized_importances, y=X.columns, hue=X.columns, dodge=False, palette='viridis', legend=False)
plt.title('Feature Importance')
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.xlim(1, 10)
plt.show()

# Save the preprocessed data with selected features
new_data = pd.DataFrame(X_res, columns=X.columns)
new_data['stroke'] = y_res.values
new_data.to_csv('/Users/gpbiz/Desktop/Project/PreprocessedDatasetUnder_SelectedFeatures.csv', index=False)

```

```

# Plot the distribution of the 'stroke' column after preprocessing
stroke_counts_b = y_res.value_counts()
plt.figure(figsize=(8, 6))
stroke_counts_b.plot(kind='bar', color=['blue', 'orange'])
plt.title("Distribution of stroke cases after preprocessing")
plt.xlabel('Stroke')
plt.ylabel('Count')
plt.xticks(ticks=[0, 1], labels=['No Stroke', 'Stroke'], rotation=0)
plt.grid(axis='y')

# Annotate the bars with the counts
for i in range(len(stroke_counts_b)):
    plt.text(i, stroke_counts_b[i], str(stroke_counts_b[i]), ha='center', va='bottom')

# Remove grid lines
plt.grid(False)
plt.show()

# Plot histograms and box plots for other features
def plot_features_distribution(df, features):
    num_features = len(features)
    fig, axes = plt.subplots(2, num_features, figsize=(5 * num_features, 10))

    for i, feature in enumerate(features):
        # Histogram
        sns.histplot(df[feature], kde=True, ax=axes[0, i])
        axes[0, i].set_title(f'Histogram of {feature}')

        # Box plot
        sns.boxplot(x=df[feature], ax=axes[1, i])
        axes[1, i].set_title(f'Boxplot of {feature}')

    plt.tight_layout()
    plt.show()

# List of features to plot
features = new_data.columns

# Group features in sets of three
for i in range(0, len(features), 3):
    group = features[i:i+3]
    plot_features_distribution(new_data, group)

pd.set_option('display.max_columns', None)

# Verification of one-hot encoding
#print(new_data.head())

```

Random Forest

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn import tree
import graphviz
import numpy as np
import pickle

# Load the preprocessed data
data = pd.read_csv('/Users/gpbiz/Desktop/Project/PreprocessedDatasetUnder.csv')

# Split the data into features and target
X = data.drop('stroke', axis=1)
y = data['stroke']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=14)

# Create a decision tree classifier
rf = RandomForestClassifier(random_state=42, n_estimators=120, criterion='entropy', max_depth=6)

# Train the classifier on the training data
rf.fit(X_train, y_train)

# Perform a 10-fold cross-validation on the training data
scores = cross_val_score(rf, X_train, y_train, cv=10)

# Make predictions on the test data
y_pred = rf.predict(X_test)

# Calculate metrics
f1 = f1_score(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
```

```

# Print the metrics
print(f"F1 Score: {f1}")
print(f"Accuracy: {accuracy}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"Confusion Matrix:\n {conf_matrix}")
print(f"Mean cross-validation score: {scores.mean()}")

# Get the first tree from the forest
estimator = rf.estimators_[0]

# Define feature_names and class_names
feature_names = X.columns
class_names = [str(x) for x in y.unique()]

dot_data = tree.export_graphviz(estimator, out_file=None,
                                feature_names=feature_names,
                                class_names=class_names,
                                filled=True)

# Graph
graph = graphviz.Source(dot_data, format="png")
graph.render("random_forest_graphviz")

# Confusion matrix heatmap
plt.figure(figsize=(10, 7))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='YlGnBu')
plt.title("Confusion Matrix - Random Forest")
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

# Save the model
with open('stroke_model.pkl', 'wb') as f:
    pickle.dump(rf, f)

```

Decision Tree

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn import tree
import graphviz

# Load the preprocessed data
data = pd.read_csv('/Users/gpbiz/Desktop/Project/PreprocessedDatasetUnder.csv')

# Split the data into features and target
X = data.drop('stroke', axis=1)
y = data['stroke']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=14)

# Create a decision tree classifier
dt = DecisionTreeClassifier(random_state=42, max_depth=7)

# Train the classifier on the training data
dt.fit(X_train, y_train)

# Perform a 10-fold cross-validation on the training data
scores = cross_val_score(dt, X_train, y_train, cv=10)

# Make predictions on the test data
y_pred = dt.predict(X_test)

# Calculate metrics
f1 = f1_score(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
```

```

# Print the metrics
print(f"F1 Score: {f1}")
print(f"Accuracy: {accuracy}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"Confusion Matrix:\n {conf_matrix}")
print(f"Mean cross-validation score: {scores.mean()}")

# Define feature_names and class_names
feature_names = X.columns
class_names = [str(x) for x in y.unique()]

dot_data = tree.export_graphviz(dt, out_file=None,
                                feature_names=feature_names,
                                class_names=class_names,
                                filled=True)

# Convert the dot file to a graph
graph = graphviz.Source(dot_data, format="png")

# Save the graph to a file
graph.render("decision_tree_graphviz")

# Plot the confusion matrix as a heatmap
plt.figure(figsize=(10, 7))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='YlGnBu')
plt.title('Confusion Matrix - Decision Tree')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

```

ADABOOST

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.ensemble import AdaBoostClassifier

# Load the preprocessed data
data = pd.read_csv('/Users/gpbiz/Desktop/Project/PreprocessedDatasetUnder.csv')

# Split the data into features and target
X = data.drop('stroke', axis=1)
y = data['stroke']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=14)

# Create a decision tree classifier
ada = AdaBoostClassifier(algorithm='SAMME', random_state=42)

# Train the classifier on the training data
ada.fit(X_train, y_train)

# Perform a 10-fold cross-validation on the training data
scores = cross_val_score(ada, X_train, y_train, cv=10)

# Make predictions on the test data
y_pred = ada.predict(X_test)

# Calculate metrics
f1 = f1_score(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
```

```
# Print the metrics
print(f"F1 Score: {f1}")
print(f"Accuracy: {accuracy}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"Confusion Matrix:\n {conf_matrix}")
print(f"Mean cross-validation score: {scores.mean()}")

# Plot the confusion matrix as a heatmap
plt.figure(figsize=(10, 7))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='YlGnBu')
plt.title('Confusion Matrix - AdaBoost')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

RUSBoost

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from imblearn.ensemble import RUSBoostClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score

# Load the preprocessed data
data = pd.read_csv('/Users/gpbiz/Desktop/Project/PreprocessedDatasetUnder.csv')

# Split the data into features and target
X = data.drop('stroke', axis=1)
y = data['stroke']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=14)

# Create a decision tree classifier
rusb = RUSBoostClassifier(n_estimators=200, random_state=42, algorithm='SAMME.R')

# Train the classifier on the training data
rusb.fit(X_train, y_train)

# Perform a 10-fold cross-validation on the training data
scores = cross_val_score(rusb, X_train, y_train, cv=10)

# Make predictions on the test data
y_pred = rusb.predict(X_test)

# Calculate metrics
f1 = f1_score(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
```

```
# Print the metrics
print(f"F1 Score: {f1}")
print(f"Accuracy: {accuracy}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"Confusion Matrix:\n {conf_matrix}")
print(f"Mean cross-validation score: {scores.mean()}")

# Plot the confusion matrix as a heatmap
plt.figure(figsize=(10, 7))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='YlGnBu')
plt.title('Confusion Matrix - RUSBoost')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

XGBoost

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from xgboost import XGBClassifier

# Load the preprocessed data
data = pd.read_csv('/Users/gpbiz/Desktop/Project/PreprocessedDatasetUnder.csv')

# Split the data into features and target
X = data.drop('stroke', axis=1)
y = data['stroke']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=14)

# Create a decision tree classifier
xgb = XGBClassifier(random_state=42, max_depth=4)

# Train the classifier on the training data
xgb.fit(X_train, y_train)

# Perform a 10-fold cross-validation on the training data
scores = cross_val_score(xgb, X_train, y_train, cv=10)

# Make predictions on the test data
y_pred = xgb.predict(X_test)

# Calculate metrics
f1 = f1_score(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
```

```
# Print the metrics
print(f"F1 Score: {f1}")
print(f"Accuracy: {accuracy}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"Confusion Matrix:\n {conf_matrix}")
print(f"Mean cross-validation score: {scores.mean()}")

# Plot the confusion matrix as a heatmap
plt.figure(figsize=(10, 7))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='YlGnBu')
plt.title('Confusion Matrix - XGBoost')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

Logistic Regression

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler, MinMaxScaler
import numpy as np
from sklearn.metrics import roc_curve, roc_auc_score

# Load the preprocessed data
data = pd.read_csv('/Users/gpbiz/Desktop/Project/PreprocessedDatasetUnder.csv')

# Split the data into features and target
X = data.drop('stroke', axis=1)
y = data['stroke']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=14)

# Scale the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Create a decision tree classifier
lr = LogisticRegression(random_state=42, C=1, solver='liblinear')

# Train the classifier on the training data
lr.fit(X_train_scaled, y_train)

# Perform a 10-fold cross-validation on the training data
scores = cross_val_score(lr, X_train_scaled, y_train, cv=10)

# Make predictions on the test data
y_pred = lr.predict(X_test_scaled)

# Calculate metrics
f1 = f1_score(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
```

```
# Print the metrics
print(f"F1 Score: {f1}")
print(f"Accuracy: {accuracy}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"Confusion Matrix:\n {conf_matrix}")
print(f"Mean cross-validation score: {scores.mean()}")

# Plot the confusion matrix as a heatmap
plt.figure(figsize=(10, 7))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='YlGnBu')
plt.title('Confusion Matrix - Logistic Regression')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

8 Βιβλιογραφία

- [1] World Health Organization (WHO). Stroke, Cerebrovascular accident. (<https://www.emro.who.int/health-topics/stroke-cerebrovascular-accident/index.html>)
- [2] Amalie Holmgaard Mersh, the lack of care for stroke survivors comes at a great cost. Euractiv, 2023. (<https://www.euractiv.com/section/health-consumers/news/the-lack-of-care-for-stroke-survivors-comes-at-a-great-cost/>)
- [3] Stevens E., Dr Emmett E., Dr Wang Y., Prof McKevitt C. & Prof Wolfe C. The Burden of Stroke in Europe. (https://www.stroke.org.uk/sites/default/files/the_burden_of_stroke_in_europe_-_challenges_for_policy_makers.pdf)
- [4] Department of Public Health (<https://www.mass.gov/info-details/stroke-signs-and-symptoms-act-fast>)
- [5] Όμιλος Ιατρικού Αθηνών, Αγγειακό Εγκεφαλικό Επεισόδιο. (<https://www.iatriko.gr/el/disease/aggeiako-egkefaliko-epeisodio?cl=609>)
- [6] Kononenko I., Machine learning for medical diagnosis: history, state of the art and perspective (<https://www.sciencedirect.com/science/article/abs/pii/S093336570100077X>)
- [7] Khosla, A., Cao, Y., Lin, C. C., Chiu, H. K., Hu, J., & Lee, H. (2010). An Integrated machine learning approach to stroke prediction. Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 183-192. Association for computing Machinery. Ανακτήθηκε από: <https://dl.acm.org/doi/abs/10.1145/1835804.1835830>.
- [8] Al-Zubaidi, H., Dweik, M., & Al-Mousa, A. (2022). Stroke Prediction Using Machine Learning Classification Methods. *Conference Paper*, November 2022, 1-8. DOI: 10.1109/ACIT57182.2022.10022050. Ανακτήθηκε από: <https://www.researchgate.net/publication/367403115>.
- [9] Wu, Y., & Fang, Y. (2020). Stroke Prediction with Machine Learning Methods among Older Chinese. *International Journal of Environmental Research and Public Health*, 17(6), 1828, 1-11. Ανακτήθηκε από: <https://www.mdpi.com/1660-4601/17/6/1828>.
- [10] Mridha, K., Ghimire, S., Shin, J., Aran, A., Uddin, M., & Mridha, M. F. (2023). Automated Stroke Prediction Using Machine Learning: An Explainable and Exploratory Study With a Web Application for Early Intervention. *IEEE Access*, 11, 52288-52307. DOI: 10.1109/ACCESS.2023.3278273.

- [11] Dev, S., Wang, H., Nwosu, C. S., Jain, N., Veeravalli, B., & John, D. (2022). A predictive analytics approach for stroke prediction using machine learning and neural networks. *Healthcare Analytics*, 2, 100032, (σ. 1-9). <https://doi.org/10.1016/j.health.2022.100032>.
- [12] Kokkotis C, Giarmatzis G, Giannakou E, Moustakidis S, Tsatalas T, Tsiptsios D, Vadikolias K, Aggelousis N. An Explainable Machine Learning Pipeline for Stroke Prediction on Imbalanced Data. *Diagnostics*. 2022; 12(10):2392. <https://doi.org/10.3390/diagnostics12102392>.
- [13] Gangavarapu Sailasya and Gorli L Aruna Kumari, “Analyzing the Performance of Stroke Prediction using ML Classification Algorithms” *International Journal of Advanced Computer Science and Applications(IJACSA)*, 12(6), 2021. <http://doi.org/10.14569/IJACSA.2021.0120662>.
- [14] Bandi, V., Bhattacharyya, D., & Midhunchakkravarthy, D. (2020). Prediction of Brain Stroke Severity Using Machine Learning. *Revue d'Intelligence Artificielle*, 34(6), 753-761. DOI: 10.18280/ria.340609. Ανακτήθηκε από <https://doi.org/10.18280/ria.340609>.
- [15] Islam, M. M., Akter, S., Rokunojjaman, M., Rony, J. H., Al Amin, A., & Kar, S. (2021). Stroke Prediction Analysis using Machine Learning Classifiers and Feature Technique. *International Journal of Electronics and Communications Systems*, 1(2), 17-22. DOI: 10.24042/ijecs.v1i2.10393. Ανακτήθηκε από: <https://www.researchgate.net/publication/357303113>.
- [16] Zhu, E., Chen, Z., Ai, P., Wang, J., Zhu, M., Xu, Z., Liu, J., & Ai, Z. (2023). Analyzing and predicting the risk of death in stroke patients using machine learning. *Frontiers in Neurology*, 14, 1096153. DOI: 10.3389/fneur.2023.1096153. Ανακτήθηκε από <https://www.frontiersin.org/articles/10.3389/fneur.2023.1096153/full>.
- [17] <https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset/data>
- [18] <https://scikit-learn.org/stable/>
- [19] <https://render.com>

