



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σχεδιασμός και ανάπτυξη Automated Machine Learning εργαλείου για IoT εφαρμογές

ΓΕΩΡΓΑΚΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ

*Επιβλέπων:*  
ΣΩΤΗΡΙΟΣ ΝΙΚΟΛΕΤΣΕΑΣ

Πάτρα 2021



## Περίληψη

Οι διαδικασίες της μηχανικής μάθησης απαιτούν εξειδίκευση, χρόνο και πόρους για την επιτυχημένη δημιουργία αποδοτικών μοντέλων. Η αυτοματοποιημένη μηχανική μάθηση (AutoML) αποτελεί ένα καινοτόμο πεδίο της επιστήμης των υπολογιστών που αφαιρεί τον ανθρώπινο παραγόντα από τις χρονοβόρες διαδικασίες της μηχανικής μάθησης, καθιστώντας ευκολότερη την δημιουργία υπολογιστικών μοντέλων. Στην παρούσα διπλωματική εργασία σχεδιάστηκε και αναπτύχθηκε ένα εργαλείο αυτοματοποιημένης μηχανικής μάθησης για IoT εφαρμογές. Το υλοποιημένο εργαλείο αποτελεί μια εύχρονη, φιλική προς τον χρήστη εφαρμογή για υπολογιστές που αποσκοπεί στην περαιτέρω διευκόλυνση εξειδικευμένων χρηστών στην υλοποίηση διαδικασιών μηχανικής μάθησης, αλλά κυρίως στο να καταστήσει την Μηχανική Μάθηση προσιτή σε υπεξειδικευμένους ανθρώπους. Η εφαρμογή υλοποιήθηκε σε Python και βασίζεται στην Python βιβλιοθήκη auto-sklearn, μέσω της οποίας υλοποιεί την διαδικασία δημιουργίας μοντέλων μηχανικής μάθησης, που αποτελεί την βασική λειτουργία της. Δευτερεύουσες λειτουργίες της εφαρμογής αποτελούν η μετατροπή προβλημάτων χρονολογικών σειρών σε κλασσικά προβλήματα μηχανικής μάθησης, η εξαγωγή χαρακτηριστικών από σύνολα δεδομένων χρονολογικών σειρών, η εξαγωγή μοντέλων υποστηριζόμενων από μικρο-ελεγκτές, η αποθήκευση των παραγόμενων μοντέλων με σκοπό την ανάκτηση και επαναχρησιμοποίησή τους. Επιπροσθέτως, μέσω της βιβλιοθήκης PyQt, υλοποιήθηκε το γραφικό περιβάλλον χρήστη της εφαρμογής για την εύκολη πλοήγηση σε αυτήν από απλούς χρήστες. Θεωρούμε πως η βιβλιογραφική έρευνα, η επίτευξη όλων των στόχων που ορίστηκαν κατά τον θεωρητικό σχεδιασμό της εφαρμογής και το τελικό εργαλείο αποτελούν σημαντική συνεισφορά στο πεδίο της Αυτοματοποιημένης Μηχανικής Μάθησης καθιστώντας το προσιτό σε περισσότερους ανθρώπους.

**Λέξεις Κλειδιά:** AutoML, αυτοματοποιημένη μηχανική μάθηση, λογισμικό, εφαρμογή, autosklearn

## Abstract

Machine learning processes require expertise, time and resources in order to successfully produce efficient models. Automated machine learning (AutoML) is an innovative field of computer science which removes the human factor from time consuming machine learning processes making it easier to create computational models. In this diploma thesis, an automated machine learning tool for IoT applications was designed and developed. The developed tool is an easy-to-use, user-friendly computer application that aims to further facilitate experienced users in implementing machine learning processes, but mainly to make Machine Learning accessible to non-experienced people. The application was implemented in Python and it is based on the Python library "auto-sklearn", through which it implements the process of creating machine learning models, which is its main function. Secondary functions of the application are the conversion of timeseries problems into classical machine learning problems, the extraction of features from timeseries data sets, the export of models supported by microcontrollers, the storage of the generated models for retrieval and reuse. Additionally, via the PyQt library, the graphical user interface of the application was implemented for an easy navigation by unexperienced users. We consider that the research, the achievement of all the objectives which were set during the initial design of the application and the final tool are an important contribution in the field of Automated Machine Learning, making it accessible to more people.

**Key Words:** AutoML, automated machine learning, software, application, autosklearn



*Στους γονείς μου και στην αδερφή μου για την πολύτιμη στήριξή τους.*

*-Γεωργακόπουλος Γεώργιος*



# Περιεχόμενα

<b>1 Εισαγωγή</b>	<b>14</b>
1. Συνεισφορά της Διπλωματικής Εργασίας . . . . .	17
2. Διάρθρωση της Διπλωματικής Εργασίας . . . . .	18
<b>2 Θεωρητικό Υπόβαθρο</b>	<b>20</b>
1. Μηχανική Μάθηση . . . . .	20
1.1 Βασικές Αρχές Μηχανικής Μάθησης . . . . .	22
1.2 Κατηγορίες Μηχανικής Μάθησης . . . . .	23
2. Αυτοματοποιημένη Μηχανική Μάθηση . . . . .	31
2.1 Γενικά . . . . .	31
2.2 Προετοιμασία-Προεπεξεργασία Δεδομένων . . . . .	33
2.3 Επεξεργασία Χαρακτηριστικών . . . . .	33
2.4 Επιλογή μοντέλου και Βελτιστοποίηση παραμέτρων . . . . .	34
3. Χρονολογικές σειρές (Timeseries) . . . . .	36
3.1 Forecasting . . . . .	36
3.2 Εξαγωγή Χαρακτηριστικών . . . . .	36
4. Γραφικό περιβάλλον . . . . .	39
4.1 Γενικά . . . . .	39
4.2 Δομικά Στοιχεία Γραφικού Περιβάλλοντος . . . . .	39
4.3 Σκοπός Γραφικού Περιβάλλοντος . . . . .	39
5. Επίλογος . . . . .	40
<b>3 Σχεδιασμός</b>	<b>41</b>
1. Υποστηριζόμενες λειτουργίες . . . . .	41
2. Ανάλυση και Περιγραφή Αρχιτεκτονικής . . . . .	42
3. Περιγραφή Εφαρμογής . . . . .	48
3.1 “Welcome” Οθόνη . . . . .	48

3.2	“Model History” Οθόνη . . . . .	49
3.3	“Import your Data” Οθόνη . . . . .	52
3.4	“Select Your Target” Οθόνη . . . . .	54
3.5	“Roll Time Series - Extract Features Οθόνη” . . . . .	57
3.6	“Preview” Οθόνη . . . . .	60
3.7	“Parameter Tuning” Οθόνη . . . . .	61
<b>4</b>	<b>Υλοποίηση</b>	<b>65</b>
1.	Τεχνική Περιγραφή και Απαιτήσεις Συστήματος . . . . .	65
2.	Οδηγοί Εγκατάστασης . . . . .	68
2.1	Περιβάλλον Υλοποίησης . . . . .	68
2.2	Οδηγός Για Προγραμματιστές . . . . .	69
2.3	Οδηγός Για Χρήστες . . . . .	75
3.	Δομή κώδικα . . . . .	75
4.	Βιβλιοθήκες της Python . . . . .	76
4.1	csv . . . . .	76
4.2	pandas . . . . .	76
4.3	sklearn . . . . .	77
4.4	auto-sklearn . . . . .	77
4.5	tsfresh . . . . .	79
4.6	micromlgen . . . . .	80
4.7	PyQt5 . . . . .	80
4.8	pickle . . . . .	80
4.9	sqlite3 . . . . .	80
<b>5</b>	<b>Έλεγχος και Αξιολόγηση</b>	<b>82</b>
1.	Γενική μεθοδολογία ελέγχου . . . . .	82
2.	Δημιουργία Classification Μοντέλου Μηχανικής Μάθησης . . . . .	86
2.1	Σύνολο Δεδομένων Classification . . . . .	86
2.2	Περιγραφή Διαδικασίας . . . . .	87
2.3	Ανάκτηση Μοντέλου και Αξιολόγηση Αποτελέσματος . . . . .	93
3.	Δημιουργία Regression Μοντέλου Μηχανικής Μάθησης . . . . .	96
3.1	Σύνολο Δεδομένων Regression . . . . .	96
3.2	Περιγραφή Διαδικασίας . . . . .	97
3.3	Ανάκτηση Μοντέλου και Αξιολόγηση Αποτελέσματος . . . . .	102

4.	Μετατροπή Προβλήματος Time Series σε Κλασσικό Πρόβλημα Μηχανικής Μάθησης . . . . .	103
4.1	Σύνολο Δεδομένων Timeseries . . . . .	103
4.2	Περιγραφή Διαδικασίας . . . . .	104
4.3	Εξαγωγή Συνόλου Δεδομένων . . . . .	107
5.	Ανάκτηση και Αξιοποίηση Αποθηκευμένων Μοντέλων Μηχανικής Μάθησης . . . . .	109
5.1	Ανάκτηση Μοντέλου μέσω της Βάσης Δεδομένων . . . . .	110
5.2	Ανάκτηση Μοντέλου μέσω του Αποθηκευμένου Αρχείου . . . . .	112
6.	Δημιουργία Μοντέλου Micro Μηχανικής Μάθησης . . . . .	114
6.1	Σύνολο Δεδομένων Classification . . . . .	114
6.2	Δημιουργία Classification Μοντέλου για Μικροελεγκτή . . . . .	114
6.3	Εφαρμογή Classification Μοντέλου στον Μικρο-ελεγκτή UNO R3 ATmega328P . . . . .	117
7.	Μελλοντικές Επεκτάσεις . . . . .	123
8.	Συμπεράσματα . . . . .	124



# Κατάλογος Σχημάτων

2.1 Φαινόμενο Overfitting . . . . .	27
2.2 Απεικόνιση του χώρου αναζήτησης μοντέλων. Παρατηρούμε πως έχει επιλεχθεί ο αλγόριθμος KNN και στην συνέχεια πρέπει να ψυχαγωγιστούν οι υπερπαραγμετροί του. . . . .	34
2.3 Rolling . . . . .	38
3.1 Αρχιτεκτονική εφαρμογής σε υψηλό επίπεδο αφαίρεσης . . . . .	43
3.2 Αρχιτεκτονική Εφαρμογής Δεύτερου Επιπέδου Αφαίρεσης . . . . .	45
3.3 Αρχιτεκτονική Εφαρμογής Τρίτου Επιπέδου Αφαίρεσης - Χρονοσειρές	46
3.4 Αρχιτεκτονική Εφαρμογής Τρίτου Επιπέδου Αφαίρεσης - Regression/Classification . . . . .	47
3.5 Αρχιτεκτονική Εφαρμογής Τρίτου Επιπέδου Αφαίρεσης - Μικρο-ελεγκτές . . . . .	47
3.6 Welcome Οθόνη . . . . .	49
3.7 Model History Οθόνη . . . . .	50
3.8 Model History Λιστα . . . . .	51
3.9 Import your Data Οθόνη . . . . .	52
3.10 Select Target Οθόνη . . . . .	54
3.11 Model Οθόνη . . . . .	56
3.12 Select Target (timeseries) Οθόνη . . . . .	57
3.13 Rolling . . . . .	58
3.14 Feature Extraction . . . . .	59
3.15 Preview Οθόνη . . . . .	60
3.16 Regression Οθόνη . . . . .	61
3.17 Regression Οθόνη . . . . .	63
4.1 Περιεχόμενα κύριου φακέλου . . . . .	72
5.1 Εμφανίζεται μετά την επιτυχή δημιουργία μοντέλου . . . . .	83

5.2	Εμφανίζεται με την επιτχή εξαγωγή ενός MicroML μοντέλου . . . . .	84
5.3	Εμφανίζεται με την επιτυχή εισαγωγή ενός αρχείου δεδομένων στην εφαρμογή . . . . .	84
5.4	Εμφανίζεται κατά την εκκίνηση της μοντελοποίησης. Ενημερώνει τον χρήστη για τον χρόνο που θα διαρκέσει. . . . .	84
5.5	Εμφανίζεται με την εισαγωγή ενός μη έγκυρου αρχείου δεδομένων στην εφαρμογή . . . . .	84
5.6	Εμφανίζεται κατά την εκκίνηση της διαδικασίας μοντελοποίησης όταν κάποιο λάθος ανιχνευθεί . . . . .	85
5.7	Εμφανίζεται όταν δεν επιλεχθεί κάποιο μοντέλο για να φορτωθεί στην εφαρμογή . . . . .	85
5.8	Εμφανίζεται όταν το μοντέλο με το οποίο επιχειρούμε να πραγματοποιήσουμε προβλέψεις έχει εκπαιδευτεί σε διαφορετικό σύνολο δεδομένων από αυτό που έχουμε εισάγει στην εφαρμογή . . . . .	85
5.9	Εμφανίζεται στην περίπτωση που δεν έχει επιλεχθεί κανένας αλγόριθμος από την λίστα στην οδόντη παραμετροποίησης . . . . .	85
5.10	Εμφανίζεται όταν το Minimum timeshift είναι αριθμός μεγαλύτερος του Maximum timeshift . . . . .	86
5.11	Εμφανίζεται όταν η μορφή του Custom Parameters πεδίου δεν είναι τύπου Python Dicrionary . . . . .	86
5.12	Εκκίνηση . . . . .	87
5.13	Εισαγωγή αρχείου - Επιλογή τύπου προβλήματος . . . . .	88
5.14	Αναζήτηση αρχείων . . . . .	89
5.15	Επιλογή Target Μεταβλητής . . . . .	90
5.16	Επιλογή Target . . . . .	91
5.17	Ρύθμιση Παραμέτρων . . . . .	92
5.18	Αποθήκευση Μοντέλου . . . . .	93
5.19	Ανάκτηση από την Βάση Δεδομένων . . . . .	94
5.20	Ανάκτηση Iris Μοντέλου . . . . .	95
5.21	Αναλυτικές προβλέψεις του Iris Μοντέλου . . . . .	96
5.22	Ανάκτηση από την Βάση Δεδομένων . . . . .	98
5.23	Επιλογή Target Μεταβλητής . . . . .	99
5.24	Επιλογή Target Μεταβλητής . . . . .	100
5.25	Επιλογή Target Μεταβλητής . . . . .	101
5.26	Ανάκτηση Wine Quality Μοντέλου . . . . .	102
5.27	Αναλυτικές προβλέψεις του Wine Quality Μοντέλου . . . . .	103
5.28	Εισαγωγή συνόλου δεδομένων και επιλογή τύπου προβλήματος	104

5.29 Επιλογή Target Μεταβλητής και Βήματος Πρόβλεψης . . . . .	105
5.30 Rolling Διαδικασία . . . . .	106
5.31 Extract Διαδικασία . . . . .	106
5.32 Εμφάνιση predictors και target στηλών . . . . .	107
5.33 Αποθήκευση αρχείου με το σύνολο δεδομένων . . . . .	108
5.34 Αποθηκευμένο αρχείο τύπου csv . . . . .	108
5.35 Μορφή του εξαγόμενου συνόλου δεδομένων . . . . .	109
5.36 Ανάκτηση από την Βάση Δεδομένων . . . . .	110
5.37 Επιλογή μοντέλου Wine Quality από την λίστα . . . . .	111
5.38 Αναλυτικές πληροφορίες Συνδυαστικού Μοντέλου Wine Quality	112
5.39 Επιλογή MicroML . . . . .	115
5.40 Δημιουργία και Εξαγωγή μοντέλου . . . . .	116
5.41 Αποθήκευση Μοντέλου MicroML . . . . .	117
5.42 Μικροελεγκτής UNO R3 ATmega328P . . . . .	118
5.43 A/B καλώδιο . . . . .	118
5.44 Device Manager - CH340 . . . . .	119
5.45 Device Manager - CH340 . . . . .	120
5.46 Arduino IDE και Serial Monitor . . . . .	122
5.47 Serial Monitor . . . . .	122
5.48 Μικροελεγκτής εν λειτουργία . . . . .	123

# Κεφάλαιο 1

## Εισαγωγή

Βρισκόμαστε στην εποχή κατά την οποία τα Μεγάλα Δεδομένα (Big Data) έχουν κυριαρχήσει στον παγκόσμιο ιστό και κατ' επέκταση σε πολλούς τομείς της καθημερινότητας του ανθρώπου. Ιδίως τα τελευταία χρόνια, με την αύξηση της δυναμικής των υπολογιστών, την ραγδαία εξέλιξη της τεχνολογίας καιθώς και την εύκολη πρόσβαση σε αυτήν από μεγάλο ποσοστό του πληθυσμού, ο όγκος των μαζικών πληροφοριακών, διαθέσιμων δεδομένων υπερβατικούς κάθε δύο χρόνια. Πολλοί θεωρούν ότι τα Μεγάλα Δεδομένα είναι κατεξοχήν η επιστήμη του μέλλοντος.

Τα δεδομένα αυτά χαρακτηρίζονται από έξι βασικές ιδιότητες, οι οποίες περιγράφονται παρακάτω.

1. Όγκος: Πρόκειται για το μέγεθος του συνόλου των δεδομένων. Ο συνολικός όγκος των μεγάλων δεδομένων είναι τόσο μεγάλος που δεν είναι εφικτό να τα επεξεργαστούμε με παραδοσιακούς τρόπους επεξεργασίας δεδομένων.
2. Ταχύτητα: Πρόκειται για τον ρυθμό με τον οποίο τα δεδομένα παράγονται και στην συνέχεια επεξεργάζονται σε πραγματικό χρόνο.
3. Ποικιλία: Τα δεδομένα παράγονται και συναντώνται σε πολλές διαφορετικές μορφές. Μπορούμε να συναντήσουμε δεδομένα με την μορφή φωτογραφιών, βίντεο, κειμένου κ.α.
4. Μεταβλητότητα: Πρόκειται για την διαφοροποίηση των δεδομένων που συναντώνται υπό την έννοια της σημασίας τους.
5. Ακρίβεια: Στόχος της επεξεργασίας των μεγάλων δεδομένων είναι η εξαγωγή συμπερασμάτων και προβλέψεων. Η ακρίβεια των δεδομένων επηρεάζει άμεσα την ακρίβεια των τελικών συμπερασμάτων.

## 6. Αξία: Πρόκειται για την πληροφοριακή αξία των δεδομένων μετά την κατάλληλη επεξεργασία και εκμετάλλευσή τους.

Ένας παράγοντας που επηρεάζει άμεσα την παραγωγή μεγάλων δεδομένων είναι το Διαδίκτυο των Πραγμάτων, κυρίως γνωστό με την αγγλική ορολογία Internet of Things (IoT). Με απλά λόγια το IoT πρόκειται για την διασύνδεση όλων των φυσικών πραγμάτων με το διαδίκτυο και μεταξύ τους. Για την κατανόηση αυτού του εγχειρήματος είναι σημαντικό να καταγράψουμε τα πλεονεκτήματα που προκύπτουν από αυτό.

Μια συσκευή που συνδέεται στο διαδίκτυο αποκτά την ικανότητα να λαμβάνει και να στέλνει δεδομένα από και προς άλλες, επίσης διασυνδεδεμένες συσκευές. Αυτή η δυνατότητα χαρακτηρίζει αυτές τις συσκευές ως έξυπνες. Μέσω της διαδικασίας αποστολής και λήψης δεδομένων μεταξύ των πραγμάτων, δεν απαιτείται οι συσκευές να έχουν όλα τα δεδομένα αποθηκευμένα σε αυτές. Κάτι τέτοιο είναι αδύνατον δεδομένου του όγκου των διαθέσιμων δεδομένων στον κόσμο. Τα δεδομένα μπορεί να βρίσκονται αποθηκευμένα οπουδήποτε στο διαδίκτυο και το μόνο που χρειάζεται να κάνει μια συσκευή είναι να συνδεθεί και να αποκτήσει πρόσβαση σε αυτά. Οι συσκευές έχουν την δυνατότητα να συλλέγουν, να λαμβάνουν, να επεξεργάζονται και να στέλνουν δεδομένα προς άλλες συσκευές.

Οι συσκευές που συλλέγουν δεδομένα αποτελούν τους αισθητήρες. Υπάρχουν διάφορα ήδη αισθητήρων που συλλέγουν δεδομένα από το περιβάλλον τους όπως φως, θερμότητα, κίνηση κ.α. Η διασύνδεση των αισθητήρων στο IoT μας επιτρέπει την συλλογή χρήσιμων δεδομένων για την εξαγωγή πολύ χρήσιμων συμπερασμάτων.

Οι συσκευές που επεξεργάζονται τα δεδομένα και εκτελούν λειτουργίες με βάση αυτά βρίσκονται παντού γύρω μας. Οι περισσότερες πλεκτρονικές συσκευές επεξεργάζονται τα σήματα που τους δίνουμε και εκτελούν πλήθος λειτουργιών.

Ο στόχος του IoT είναι να αποτελείται από διασυνδεδεμένες συσκευές που μπορούν να συλλέγουν δεδομένα, να τα επεξεργάζονται και να τα εκμεταλλεύονται αυτόματα για να εκτελέσουν κάποια λειτουργία χωρίς την ανθρώπινη παρέμβαση. Είναι προφανές πως οι συσκευές που διασυνδέονται στο IoT παράγουν τεράστια ποσά δεδομένων καθιστώντας ακόμα μεγαλύτερη την ανάγκη για την επεξεργασία και την αξιοποίησή τους.

Το IoT και τα Big Data έχουν ποικίλες χρήσιμες και πρακτικές εφαρμογές σε διάφορους τομείς της ζωής μας. Κάποιοι από αυτούς είναι:

### ► Έξυπνες πόλεις

- Αυτοματοποιημένη παρακολούθηση
- Έξυπνος φωτισμός για μείωση της κατανάλωσης ενέργειας

- Διαχείριση κίνησης μέσω έξυπνων σηματοδοτών
- Έξυπνο παρκάρισμα
- Διαχείριση ρύπων και σκουπιδιών

► Έξυπνα εργοστάσια

- Αυτοματοποίηση ροής εργασιών
- Βελτιστοποίηση συστημάτων παραγωγής
- Μείωση κόστους
- Αύξηση παραγωγικότητας και ποιότητας

► Έξυπνα σπίτια

- Απομακρυσμένος έλεγχος συσκευών
- Έξυπνες κλειδαριές

► Ψηφιακά δίδυμα

Η Μηχανική Μάθηση έχει, επίσης, πρωταγωνιστικό ρόλο στην καθημερινότητα μας, στις μέρες μας, βρίσκοντας εφαρμογή σε πολλούς τομείς. Καθίσταται αναγκαίο να μέθοδοι της μηχανική μάθησης να αυτοματοποιηθούν πλήρως και να γίνουν όσο το δυνατόν πιο προσιτές στον ευρύτερο πληθυσμό, που κατά κύριο λόγο αποτελείται από ανθρώπους οι οποίοι δεν κατέχουν το απαραίτητο γνωστικό υπόβαθρο για την κατανόηση και χρήση τους. Βασικός παράγοντας στην διαδικασία της μηχανικής μάθησης είναι ο άνθρωπος. Πολλές από τις υπό-διεργασίες της απαιτούν την παρέμβαση ενός ανθρώπου με εξειδίκευση πάνω στο αντικείμενο της ανάλυσης δεδομένων. Για παράδειγμα η επεξεργασία των δεδομένων, η ρύθμιση των μοντέλων και των παραμέτρων τους, απαιτούν χρονοβόρες δοκιμές και trial and error διαδικασίες που εκτελούνται συνήθως από κάποιον ειδικό. Η αυτοματοποίηση των διαδικασιών της μηχανικής μάθησης αποτελεί αντικείμενο διαρκούς έρευνας που εξελίσσεται με ζαγδαίους ρυθμούς καθώς η πρόοδος της τεχνολογίας την ενθαρρύνει όλο και περισσότερο. Σκοπός της είναι η απαλλαγή των ερευνητών από επαναλαμβανόμενες και τετριζμένες διεργασίες και η προώθηση της ταχείας επίλυσης νέων προβλημάτων με μικρότερες απαιτήσεις σε κόστος, χρόνο και υπολογισμούς. Η βασική θεωρία πίσω από την μηχανική μάθηση είναι η δυνατότητα των υπολογιστών να εκτελούν διεργασίες και να επιλύουν προβλήματα, μαθαίνοντας μέσω της εμπειρίας, χωρίς την βοήθεια και την παρέμβαση του ανθρώπου.

## 1. Συνεισφορά της Διπλωματικής Εργασίας

Καθώς, όπως αναφέρθηκε στις προηγούμενες ενότητες, η τεχνολογία εξελίσσεται ραγδαία, η διάδοσή της στον κόσμο και η πρόσβασή σε αυτήν από συνεχώς περισσότερους ανθρώπους είναι αναγκαία. Η μηχανική μάθηση είναι ένας τομέας που κυριαρχεί στον κόσμο των δεδομένων. Η παρούσα διπλωματική εργασία αφορά στον σχεδιασμό και την ανάπτυξη ενός εργαλείου αυτοματοποιημένης μηχανικής μάθησης για IoT εφαρμογές. Σκοπός του εργαλείου είναι να φέρει ένα βήμα πιο κοντά στις διαδικασίες μηχανικής μάθησης ανθρώπους που δεν έχουν υπόβαθρο στον τομέα της Ανάλυσης Δεδομένων καθώς και να διευκολύνει ακόμα περισσότερο όσους είναι γνώστες της επιστήμης αυτής.

Ένα από τα βασικά προβλήματα που αντιμετωπίζονται στην Μηχανική Μάθηση, ακόμα και από τους ειδικούς στον τομέα αυτόν, είναι οι χρονοβόρες και τετριμμένες διαδικασίες που απαιτούν συνεχείς δοκιμές και προσαρμογές. Για τον λόγο αυτό στην εφαρμογή που αναπτύχθηκε έγινε χρήση της Python βιβλιοθήκης, auto-sklearn, που δίνει την δυνατότητα υλοποίησης **αυτοματοποιημένης** μηχανικής μάθησης με σκοπό την εξάλειψη των χρονοβόρων και την αυτοματοποίηση των περισσότερων διαδικασιών. Το αποτέλεσμα είναι, από την μία η αύξηση της παραγωγικότητας και η καλύτερη αξιοποίηση του χρόνου των αναλυτών, και από την άλλη η ενθάρρυνση των απλών χρηστών να εξερευνήσουν και να έρθουν πιο κοντά στον κόσμο της μηχανικής μάθησης και να υλοποιήσουν τα δικά τους μοντέλα.

Ένα επιπλέον πρόβλημα που αντιμετωπίζει ένα άτομο που δεν είναι σχετικό με την επιστήμη της Πληροφορικής είναι η ελάχιστη εξουκείωσή του με τον κόσμο του προγραμματισμού. Οι περισσότερες τεχνολογίες υλοποίησης μηχανικής μάθησης απαιτούν την σύνταξη και την εκτέλεση προγραμματιστικού κώδικα σε απλές ή πιο πολύπλοκες μορφές. Για την αντιμετώπιση του προβλήματος αυτού αναπτύχθηκε ένα κατανοητό και όσο το δυνατόν πιο απλοποιημένο γραφικό περιβάλλον χρήστη με την χρήση της Python βιβλιοθήκης PyQt. Σκοπός του είναι η δυνατότητα χρήσης της εφαρμογής και δημιουργίας μοντέλων μηχανικής μάθησης δίχως να απαιτείται προγραμματιστικό υπόβαθρο από τους χρήστες. Όλες οι περίπλοκες προγραμματιστικές διεργασίες που απαιτούνται για την εκτέλεση διαφόρων λειτουργιών, πραγματοποιούνται "πίσω" από το γραφικό περιβάλλον χρήστης, παρέχοντας στον χρήστη μία απλοποιημένη και ξεκούραστη εμπειρία και πλούγηση ενθαρρύνοντας τον να πειραματιστεί.

Επιπλέον, η εφαρμογή αξιοποιεί τις πολύ χρήσιμες πληροφορίες που παρέχονται από χρονολογικές σειρές, και εξάγει ακόμα περισσότερη πληροφοριακή αξία από αυτές μέσω της Python βιβλιοθήκης tsfresh, δημιουργώντας μοντέλα μηχανικής μάθησης με μεγάλη απόδοση. Όλες οι απαραίτητες διαδικασίες για την επεξεργασία και την εξαγωγή των δεδομένων πραγματοποιούνται

αυτόματα στο πίσω μέρος της εφαρμογής παρέχοντας ευκολία στην υλοποίηση μέσω του γραφικού περιβάλλοντος χρήστη.

Για την περαιτέρω επεκτασιμότητα της εφαρμογής επιλέχθηκε να γίνει υλοποίηση micro-Μηχανικής Μάθησης. Τα μοντέλα που προκύπτουν από την διαδικασία αυτή δύναται να εφαρμοστούν πάνω σε μικρο-ελεγκτές, όπως για παράδειγμα μικρο-ελεγκτές Arduino και να πραγματοποιήσουν προβλέψεις. Μέσω της υλοποίησης αυτής το ανεπτυγμένο εργαλείο μπορεί να επεκταθεί και να αποτελέσει εργαλείο μηχανικής μάθησης για εφαρμογές στο Διαδίκτυο των Πραγμάτων.

Συμπερασματικά, η εφαρμογή που υλοποιήθηκε στην παρούσα διπλωματική εργασία αποτελεί ένα εργαλείο που απαλλάσσει τον ανθρώπινο παράγοντα από τις διαδικασίες μηχανικής μάθησης, δίνει την δυνατότητα δημιουργίας δυνατών υπολογιστικών μοντέλων, αξιοποιεί την πληροφοριακή αξία των δεδομένων χρονοσειρών, δίνει την δυνατότητα δημιουργίας μοντέλων συμβατών με μικρο-ελεγκτές και παρέχει ένα απλουστευμένο και κατανοητό περιβάλλον χρήστη μέσω του οποίου κάποιος μπορεί να υλοποιήσει όλες τις παραπάνω λειτουργίες.

## 2. Διάρθρωση της Διπλωματικής Εργασίας

Η εργασία αυτή είναι οργανωμένη σε 6 κεφάλαια. Το παρόν κεφάλαιο αποτελεί μια εισαγωγή στον κόσμο των μαζικών δεδομένων, του Internet of Things και της αυτοματοποιημένης και μη μηχανικής μάθησης.

Στο Κεφάλαιο 2 παρουσιάζεται το θεωρητικό υπόβαθρο των βασικών τεχνολογιών που σχετίζονται με τη διπλωματική αυτή. Αρχικά περιγράφονται οι βασικές αρχές και έννοιες της μηχανικής μάθησης. Στην συνέχεια περιγράφεται η αυτοματοποιημένη μηχανική μάθηση με την βασική ροή των υπό-διαδικασιών στις οποίες χωρίζεται. Ακολουθεί η περιγραφή και η χρονισμότητα της ανάλυσης και της εκμετάλλευσης των χρονοσειρών. Τέλος, παρουσιάζεται η σημασία και αναλύεται η θεωρία πίσω από το γραφικό περιβάλλον χρήστη των σύγχρονων εφαρμογών.

Στο Κεφάλαιο 3 παρουσιάζεται ο σχεδιασμός του εργαλείου που υλοποιήθηκε στην παρούσα εργασία. Συγκεκριμένα αναλύεται η αρχιτεκτονική και περιγράφεται η εφαρμογή με τις κύριες λειτουργίες της.

Στο Κεφάλαιο 4 παρουσιάζεται η διαδικασία υλοποίησης του εργαλείου. Συγκεκριμένα παρουσιάζονται όλα τα εργαλεία και οι βιβλιοθήκες που χρησιμοποιούνται και βοήθησαν στην υλοποίηση της εφαρμογής καθώς και η δομή του κώδικα.

Στο Κεφάλαιο 5 αναλύεται η μεθοδολογία ελέγχου για την ορθή λειτουργία της εφαρμογής και στην συνέχεια παρουσιάζεται η αξιολόγηση της εφαρμογής

πάνω στις λειτουργίες που υλοποιούνται μέσω αυτής.

## Κεφάλαιο 2

# Θεωρητικό Υπόβαθρο

### 1. Μηχανική Μάθηση

Η μηχανική μάθηση αποτελεί έναν τομέα της επιστήμης των υπολογιστών που πηγάζει από την έρευνα πάνω στην αναγνώριση προτύπων καθώς και την τεχνητή νοημοσύνη μέσω της υπολογιστικής μάθησης. Σκοπός της Μηχανικής Μάθησης είναι η δημιουργία αλγορίθμων, οι οποίοι μαθαίνουν χωρίς τον εξωτερικό προγραμματισμό από τον άνθρωπο και πραγματοποιούν προβλέψεις πάνω σε σύνολα δεδομένων. Συγκεκριμένα πρόκειται για την κατασκευή μοντέλων με σκοπό την πρόβλεψη πάνω σε δεδομένα, εφόσον πρώτα έχουν εκπαιδευτεί πάνω σε παραδείγματα παρόμοιων δεδομένων. [33]

**Ορισμός:** “Ενα υπολογιστικό πρόγραμμα θεωρείται πως μαθαίνει μέσω της εμπειρίας Ε σε σχέση με μια διεργασία Τ και μία μετρική απόδοσης P, εάν η απόδοσή του για το Τ, όπως μετράται από το P, βελτιώνεται μέσω της εμπειρίας Ε” -Tom Mitchell, Πανεπιστήμιο Carnegie Mellon [33]

Οι τεχνικές μηχανικής μάθησης έχουν πλέον εισχωρήσει σε πολλούς τομείς της ζωής μας και αποτελούν αναπόσπαστο κομμάτι της. Η μηχανική μάθηση έχει εφαρμογές σε:

- ▶ Έξυπνα κινητά: Η μηχανική μάθηση βρίσκεται σε μεγάλο βαθμό στα έξυπνα κινητά. Μερικά παραδείγματα είναι οι βοηθοί μέσω φωνητικών εντολών, το ξεκλείδωμα μέσω αναγνώρισης προσώπων και η αναγνώριση αντικειμένων στις κάμερες.
- ▶ Μετακινήσεις - Κυκλοφορία: Μηχανική μάθηση συναντάται στον υπολογισμό της βέλτιστης διαδρομής σε διάφορες εταιρίες παροχής υπηρεσιών μεταφοράς προσώπων. Επιπλέον συναντάται στην εκτίμηση της κατάστασης της κυκλοφορίας

- ▶ Φιλτράρισμα μηνυμάτων πλεκτρονικού ταχυδρομείου: Η μηχανική μάθηση συναντάται στην αναγνώριση και τον αποκλεισμό ανεπιθύμητων μηνυμάτων
- ▶ Μηχανές αναζήτησης
- ▶ Εξατομικευμένο marketing μέσω διαφημίσεων και προτάσεων: Κάθε χρήστης έχει ένα ψηφιακό αποτύπωμα διαμορφωμένο από την συμπεριφορά πλοήγησης του στο διαδίκτυο. Τα δεδομένων που προκύπτουν από αυτήν την συμπεριφορά χρησιμοποιούνται στην προώθηση εξατομικευμένου υλικού διαφημίσεων και προτάσεων σε κάθε χρήστη.
- ▶ Ηλεκτρονική ασφάλεια (Capchas)
- ▶ Έλεγχος απάτης σε τραπεζικές συναλλαγές
- ▶ Εξατομικευμένες τραπεζικές υπηρεσίες
- ▶ Μηχανική όραση
- ▶ Αυτό-οδηγούμενα οχήματα: Η μηχανική μάθηση συναντάται σε αυτήν την περίπτωση στην αναγνώριση αντικειμένων, υπολογισμό αποστάσεων κ.α

Μέσω των μεθόδων μηχανικής μάθησης, η καθημερινότητα του ανθρώπου διευκολύνεται και πολλοί τεχνολογικοί και μη τομείς, συνεχώς, εξελίσσονται με ραγδαίους ρυθμούς. Για την υλοποίηση ενός επιτυχημένου και ποιοτικού συστήματος μηχανικής μάθησης για την επίλυση ενός προβλήματος απαιτείται:

- ▶ Εμπειρία
- ▶ Εξειδίκευση
- ▶ Χρόνος
- ▶ Υπολογιστικοί πόροι
- ▶ Οικονομικοί πόροι

Οι βασικές έννοιες της μηχανικής μάθησης είναι οι εξής:

1. Η αναπαράσταση: ο τρόπος αναπαράστασης της γνώσης, δηλαδή οι αλγόριθμοι μηχανικής μάθησης
2. Η αξιολόγηση: Ο τρόπος αξιολόγησης των αλγορίθμων μηχανικής μάθησης, δηλαδή οι διάφορες μετρικές που χρησιμοποιούνται ανάλογα με τις ανάγκες μας και τον τύπο του προβλήματος.
3. Η βελτιστοποίηση: Ο τρόπος με τον οποίο παράγονται νέοι αλγόριθμοι και βελτιστοποιούνται κατά την διαδικασία της αναζήτησης.

## 1..1 Βασικές Αρχές Μηχανικής Μάθησης

**Συλλογή δεδομένων:** Το πρώτο βήμα της διαδικασίας μηχανικής μάθησης αποτελεί η αναζήτηση και εύρεση δεδομένων.

**Προετοιμασία και Προ-επεξεργασία δεδομένων:** Για την αποδοτική λειτουργία του τελικού μοντέλου βασική προϋπόθεση είναι η σωστή προετοιμασία και προ-επεξεργασία των δεδομένων εισόδου. Σκοπός της διεργασίας αυτής είναι η απαλοιφή του θορύβου που ενδέχεται να υπάρχει σε ακατέργαστα δεδομένα ώστε να εξαχθεί περισσότερη χρήσιμη πληροφορία από αυτά. Δεδομένα που περιέχουν θόρυβο μπορούν να οδηγήσουν σε παραπλανητικές προβλέψεις και σε χαμηλή απόδοση των αλγορίθμων μηχανικής μάθησης.

**Διαχωρισμός δεδομένων:** Εφόσον η προ-επεξεργασία έχει ολοκληρωθεί και τα δεδομένα εισόδου είναι καθαρά και σε κατάλληλη μορφή, το επόμενο βήμα είναι ο διαχωρισμός τους σε δύο διαφορετικά και ανεξάρτητα μεταξύ τους σύνολα δεδομένων. Το πρώτο υποσύνολο απαρτίζεται από τα δεδομένα που χρησιμοποιούνται για την εκπαίδευση των αλγορίθμων κατά την διαδικασία της μάθησης. [10] Το δεύτερο υποσύνολο χρησιμοποιείται για τον έλεγχο των αλγορίθμων. Σκοπός είναι να αποφευχθεί το φαινόμενο του overfitting, που θα αναλυθεί παρακάτω, και να εξασφαλιστεί η υψηλή απόδοση και γενίκευση του μοντέλου. Μετά από αυτόν τον διαχωρισμό, ένα μικρό υποσύνολο των δεδομένων εκπαίδευσης χρησιμοποιείται για την αξιολόγηση των αλγορίθμων.

**Εύρεση και εκπαίδευση αλγορίθμου:** Στην συνέχεια πρέπει να επιλεχθεί ο κατάλληλος αλγόριθμος για το τρέχον πρόβλημα που καλείται να επιλυθεί. Υπάρχουν διαφορετικοί αλγόριθμοι για διαφορετικά προβλήματα. Μόλις επιλεχθεί ο κατάλληλος αλγόριθμος πρέπει να παραμετροποιηθεί κατάλληλα ώστε να πραγματοποιεί προβλέψεις υψηλής ακρίβειας.

**Αξιολόγηση:** Ο επιλεγμένος και τελικός αλγόριθμος εφαρμόζεται πάνω στο υποσύνολο δεδομένων ελέγχου και αξιολογείται η απόδοσή του.

**Βελτιστοποίηση υπερ-παραμέτρων:** Αποτελεί μια επαναληπτική διαδικασία, όπου παραμετροποιούνται οι υπερ-παράμετροι του μοντέλου με σκοπό την βελτιστοποίηση του. Ο έλεγχος της απόδοσης γίνεται βάσει του υποσυνόλου δεδομένων ελέγχου.

## 1..2 Κατηγορίες Μηχανικής Μάθησης

### 1..2..1 Μη επιβλεπόμενη Μηχανική Μάθηση

Στην μη επιβλεπόμενη μηχανική μάθηση, οι αλγόριθμοι αναζητούν και ανακαλύπτουν συσχετίσεις και μοτίβα μεταξύ των δεδομένων εισόδου με άγνωστα δεδομένα εξόδου. Στόχος είναι η εύρεση των δομικών σχηματισμών τους. Η παρούσα διπλωματική εργασία δεν ασχολείται με την συγκεκριμένη κατηγορία μηχανικής μάθησης και την εφαρμογή της.

### 1..2..2 Ενισχυτική Μάθηση

Η Ενισχυτική Μάθηση είναι μια κατηγορία μάθησης κατά την οποία λαμβάνονται αποφάσεις με σκοπό την πραγματοποίηση των κατάλληλων ενεργειών οι οποίες θα αποφέρουν το πιο θετικό αποτέλεσμα. Το σύστημα που μαθαίνει δεν γνωρίζει κάθε φορά τι ενέργεια θα πρέπει να πραγματοποιήσει. Μόλις μία κατάσταση προκύψει, το σύστημα θα πραγματοποιήσει μία ενέργεια που θα έχει κάποιο αντίκτυπο στην κατάσταση αυτή. Το πόσο θετικό ή αρνητικό είναι το αντίκτυπο θα αποτελέσει κριτήριο για τις μελλοντικές ενέργειες του συστήματος σε μία παρόμοια κατάσταση. Βασικός στόχος είναι συνήθως η μεγιστοποίηση μιας συνάρτησης ανταμοιβής ή η ελαχιστοποίηση μιας συνάρτησης κόστους. [18] Η συγκεκριμένη κατηγορία μηχανικής μάθησης βρίσκει εφαρμογή σε διάφορους τομείς όπως:

- ▶ Η θεωρία παιγνίων
- ▶ Η θεωρία ελέγχου
- ▶ Η θεωρία πληροφορίας

Η παρούσα διπλωματική εργασία δεν θα ασχοληθεί με την συγκεκριμένη κατηγορία μηχανικής μάθησης. Σε αντίθεση με την μη επιβλεπόμενη μηχανική μάθηση, η ενισχυτική μάθηση έχει ως βασικό στόχο την χαρτογράφηση των δεδομένων εισόδου με τα δεδομένα εξόδου. Στην περίπτωση της μη επιβλεπόμενης μάθησης, στόχος είναι η εύρεση μοτίβων μεταξύ των δεδομένων. [5]

### 1..2..3 Μηχανική Μάθηση με Επίβλεψη

Η παρούσα διπλωματική εργασία θα ασχοληθεί αποκλειστικά με την συγκεκριμένη κατηγορία μηχανικής μάθησης και την εφαρμογή της. Η επιβλεπόμενη μηχανική μάθηση είναι η διαδικασία της εκμάθησης ενός αλγορίθμου να αντιστοχίζει δεδομένα εισόδου σε δεδομένα εξόδου βάσει υπαρχόντων παραδειγμάτων τέτοιων ζευγών εισόδου - εξόδου.[32]

Πιο συγκεκριμένα, το πρώτο μέρος ενός τέτοιου ζεύγους είναι το σύνολο χαρακτηριστικών και αποτελείται από ένα διάνυσμα μήκους  $n$ . Το δεύτερο μέρος του ζεύγους είναι μία πραγματική τιμή  $\hat{y}$  μια διακριτή κλάση, ανάλογα με την κατηγορία του προβλήματος μηχανικής μάθησης. Ένας αλγόριθμος μηχανικής μάθησης εκπαιδεύεται πάνω στα υπάρχοντα ζεύγη με σκοπό την εύρεση μιας συνάρτησης που θα χρησιμοποιηθεί για να προβλέπει νέα ζεύγη. Στόχος, λοιπόν, είναι η δημιουργία ενός γενικευμένου μοντέλου, χωρίς μεγάλο σφάλμα γενίκευσης, που βασίζεται πάνω στα αρχικά δεδομένα με σκοπό την εφαρμογή του για την πρόβλεψη νέων, άγνωστων δεδομένων. [14] [40]

Για την εκπαίδευση ενός μοντέλου χρησιμοποιείται ένα πεπερασμένο πλήθος δεδομένων που αποτελεί μέρος του αρχικού συνόλου δεδομένων και ονομάζεται σύνολο εκπαίδευσης. Στην επιβλεπόμενη μηχανική μάθηση, κάθε διάνυσμα χαρακτηριστικών αντιστοιχεί σε κάποια κλάση. Στην συνέχεια, ένα νέο σύνολο δεδομένων παρέχεται ως είσοδος στο εκπαιδευμένο μοντέλο, με σκοπό να γίνει έλεγχος της απόδοσης του. Το σύνολο που χρησιμοποιείται σε αυτό το σημείο της διαδικασίας αποτελείται από δεδομένα ανεξάρτητα από εκείνα που χρησιμοποιήθηκαν κατά την εκπαίδευση του μοντέλου. Το μοντέλο, με την νέα είσοδο, καλείται να προβλέψει και να κατηγοριοποιήσει τα δεδομένα σε μία από τις υπάρχουσες, διακριτές κλάσεις. Αυτή η κατηγορία μηχανικής μάθησης αποτελείται από δύο υποκατηγορίες.

**Classification** Είναι μια διαδικασία της επιβλεπόμενης μηχανικής μάθησης κατά την οποία γίνεται αναζήτηση ενός μοντέλου που θα έχει την δυνατότητα να κατηγοριοποιήσει δεδομένα σε διακριτές κλάσεις. Τα δεδομένα αντιστοιχούν σε κάποια διακριτή ετικέτα / ομάδα και κατά την διαδικασία της ταξινόμησης γίνεται η πρόβλεψη της ομάδας τους. Το παραγόμενο μοντέλο μπορεί να περιγραφεί μέσω της προγραμματιστικής λογικής if-then. Συμπερασματικά, η εφαρμογή της διαδικασίας αυτής είναι εφικτή στις περιπτώσεις όπου τα δεδομένα διαχωρίζονται υπό διαφορετικές διακριτές ετικέτες.[12]

**Regression** Είναι, επίσης, μια διαδικασία επιβλεπόμενης μηχανικής μάθησης κατά την οποία αναζητείται ένα μοντέλο, το οποίο θα μπορεί να διαχωρίζει τα δεδομένα εισόδου σε συνεχείς πραγματικές τιμές, αντίθετα με την διαδικασία Classification όπου οι τιμές εξόδου ήταν διακριτές. Ουσιαστικά, ένα μοντέλο regression κάνει προβλέψεις “ποσότητας”, επομένως η απόδοσή του μετράται με το σφάλμα των προβλέψεων σε σχέση με τις πραγματικές τιμές εξόδου.[12]

#### 1.2.4 Αλγόριθμοι Επιβλεπόμενης Μηχανικής Μάθησης

Οι αλγόριθμοι Μηχανικής Μάθησης υπό επιβλεψη απαιτούν εξωτερική παρέμβαση. Οι συγκεκριμένοι αλγόριθμοι μαθαίνουν κάποια μοτίβα από τα

σύνολα δεδομένων στα οποία εκπαιδεύονται και στην συνέχεια αξιολογούνται και χρησιμοποιούνται για νέες προβλέψεις ή ταξινομήσεις. [18]

**Naïve Bayes:** Βασίζεται στο θεώρημα του Bayes και την δεσμευμένη πιθανότητα. Κάθε Bayesian ταξινομητής, λαμβάνει ως δεδομένο ότι ένα συγκεκριμένο χαρακτηριστικό μιας κλάσης είναι στατιστικά ανεξάρτητο από κάθε άλλο χαρακτηριστικό της κλάσης αυτής. Κατασκευάζει δέντρα βάσει της πιθανότητας ενός γεγονότος να συμβεί. Το σύνολο των δέντρων αποτελεί ένα Bayesian δίκτυο. [18] Υπάρχουν τρία ήδη Bayesian ταξινομητών:

1. Ο Bernoulli Naïve Bayes ταξινομητής λαμβάνει μόνο δυαδικές τιμές και αποτελεί κατάλληλη επιλογή για μεγάλα σύνολα δεδομένων. Ένα μειονέκτημα του αλγορίθμου είναι το γεγονός ότι η συσχέτιση μεταξύ χαρακτηριστικών μειώνει την απόδοση του αλγορίθμου, καθώς απαιτεί όσο το δυνατόν ασυσχέτιστα μεταξύ τους χαρακτηριστικά. [30]
2. Ο Multinomial Naïve Bayes ταξινομητής είναι κατάλληλος για την ταξινόμηση με σύνολα δεδομένων που αποτελούνται από διακριτά μετρήσιμα χαρακτηριστικά όπως για παράδειγμα το πλήθος λέξεων ενός κειμένου. Η διαφορά του με τον Bernoulli Naïve Bayes ταξινομητή είναι το γεγονός ότι δεν περιορίζεται μόνο σε δυαδικά χαρακτηριστικά (Boolean / Binary)[30]
3. Ο Gaussian Naïve Bayes: ταξινομητής χρησιμοποιείται στην περίπτωση που τα χαρακτηριστικά είναι πραγματικές τιμές αντίθετα με τα άλλα είδη Naïve Bayes ταξινομητών. [28]

**k-Nearest Neighbors:** Ο αλγόριθμος αυτός βασίζεται στην θεωρία ότι όμοια πράγματα βρίσκονται σε μικρή απόσταση μεταξύ τους. Ο αλγόριθμος χαρακτηρίζεται από απλότητα και ευκολία στην εφαρμογή του. Επιπλέον μπορεί να εφαρμοστεί τόσο σε προβλήματα classification όσο και σε προβλήματα regression. Το βασικό μειονέκτημα του αλγορίθμου είναι ότι γίνεται αργός όσο τα χαρακτηριστικά που χρησιμοποιούνται για την πρόβλεψη αυξάνονται. [25]

**Decision Trees:** Τα δέντρα αποφάσεων ταξινομούν χαρακτηριστικά βάσει των τιμών τους. Συνήθως χρησιμοποιούνται για classification προβλήματα μηχανικής μάθησης. Κάθε δέντρο αποτελείται από κόμβους, που αναπαριστούν ομάδες δεδομένων προς ταξινόμηση, και ακμές, που αντιστοιχούν σε τιμές τις οποίες μπορούν να πάρουν τα δεδομένα στους κόμβους. [18] Τα δέντρα αποφάσεων χαρακτηρίζονται από απλότητα και ευκολία στην κατανόηση και

σύλληψη. Κάποια από τα μειονεκτήματά τους είναι η επιφρέπεια τους στο φαινόμενο του overfitting, η αστάθεια τους λόγω υψηλής διακύμανσης, το υψηλό bias σφάλμα. [24]

**Random Forest:** Αποτελείται από ένα μεγάλο πλήθος μεμονωμένων δέντρων απόφασης που δημιουργούν ένα συνδυαστικό μοντέλο. Κατά τον διαχωρισμό ενός κόμβου στην δημιουργία ενός δέντρου, ο καλύτερος διαχωρισμός επιλέγεται με κάποια τυχαία διαδικασία με σκοπό την μείωση της διακύμανσης του μοντέλου και την αποφυγή του overfitting, αυξάνοντας την απόδοση του. [30] [39]

**Extremely Randomized Trees:** Όπως και το Random Forest αποτελεί ένα συνδυαστικό μοντέλο που αποτελείται από πολλά δέντρα αποφάσεων και χρησιμοποιεί μία τυχαία διαδικασία επιλογής του κανόνα διαχωρισμού των κόμβων. Το μειονέκτημα του μοντέλου είναι ότι έχει αυξημένο bias με αντάλλαγμα το πλεονέκτημα της μειομένης διακύμανσης. [30]

**Boosting:** Πρόκειται για έναν μετα-αλγόριθμο μηχανικής μάθησης. Αποτελείται από έναν συνδυασμό ασθενών αλγορίθμων (για παράδειγμα δέντρα απόφασης), οι έξοδοι των οποίων συνεισφέρουν με ένα βάρος στο τελικό αποτέλεσμα του μοντέλου και δημιουργούν είναι ένας ισχυρό αλγόριθμο. Οι επιμέρους αλγόριθμοι προσαρμόζονται με σκοπό να εστιάζουν κάθε φορά κυρίως στις εσφαλμένες προβλέψεις προσπαθώντας να αυξήσουν την απόδοση του συνδυαστικού μοντέλου. Έχει την δυνατότητα να ελαττώσει το σφάλμα που προκύπτει από τους αδύναμους επιμέρους αλγορίθμους πραγματοποιώντας σειριακή εκπαίδευση. Η απόδοση ενός επιμέρους αλγορίθμου είναι συνάρτηση της απόδοσης του αμέσως προηγούμενου. Με αυτή τη μέθοδο, δημιουργούνται διαδοχικά μοντέλα που αποτελούν ένα συνδυαστικό, συνολικό και τελικό μοντέλο. Η αξιολόγηση του τελικού μοντέλου γίνεται με voting ή με τον weighted arithmetic mean, ανάλογα με τον τύπο του προβλήματος μηχανικής μάθησης που καλούμαστε να αντιμετωπίσουμε. [7]

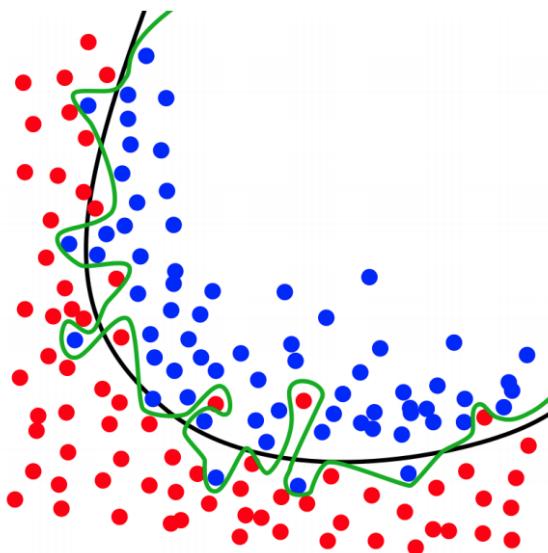
**Discriminant Analysis:** Πρόκειται για αλγορίθμους που αναζητούν έναν γραμμικό συνδυασμό χαρακτηριστικών που περιγράφει ή διαχωρίζει τα δεδομένα εισόδου στις διακριτές κλάσεις που υπάρχουν στο πρόβλημα. [8]

**SVM:** Ο συγκεκριμένος αλγόριθμος χρησιμοποιείται κυρίως για classification προβλήματα και βασίζεται στον υπολογισμό διαστημάτων. Ο αλγόριθμος σχεδιάζει διαστήματα ανάμεσα στις κλάσεις έτσι ώστε να έχουν την μέγιστη απόσταση μεταξύ τους και να ελαχιστοποιήσει το σφάλμα ταξινόμησης. [18]

**Linear Classifier (SGD):** Ο αλγόριθμος αυτός αφορά σε προβλήματα ταξινόμησης. Ο αλγόριθμος αποφασίζει την κλάση των δεδομένων εισόδου μέσω του γραμμικού συνδυασμού των χαρακτηριστικών τους. Έχει πλεονέκτημα όταν το σύνολο δεδομένων εισόδου αποτελείται από πολλά χαρακτηριστικά, καθώς είναι αποδοτικός όσο μη-γραμμικοί ταξινομητές ένω ταυτόχρονα υλοποιείται και εκπαιδεύεται σε λιγότερο χρόνο. [6]

### 1.2.5 Overfitting

Βασική προϋπόθεση για την δημιουργία ενός επιτυχημένου μοντέλου μηχανικής μάθησης είναι η αποφυγή του φαινόμενου του overfitting. Το φαινόμενο αυτό παρατηρείται όταν το μοντέλο μηχανικής μάθησης που παράγεται είναι παραμετροποιημένο ώστε να παρουσιάζει υψηλή διακύμανση(variance). Ως αποτέλεσμα, το μοντέλο δεν έχει την δυνατότητα να γενικεύσει το μοτίβο του συνόλου δεδομένων εισόδου και εμφανίζει χαμηλή απόδοση όταν δέχεται ως είσοδο νέα δεδομένα. Συνήθως το μοντέλο έχει δημιουργηθεί και παραμετροποιηθεί θεωρώντας μέρος του θορύβου και της ποικιλομορφίας των δεδομένων ως το υποκείμενο μοτίβο που κυριαρχεί σε αυτά. Ένα επιτυχημένο μοντέλο πρέπει να μπορεί να εξάγει το υποβόσκον μοτίβο της εισόδου, γενικεύοντας την ώστε να μπορεί να πραγματοποιεί αξιόπιστες προβλέψεις πάνω σε νέα, άγνωστα δεδομένα.[9]



Σχήμα 2.1: Φαινόμενο Overfitting

Παρατηρούμε πως η μαύρη γραμμή αποτελεί μία καλύτερη γενίκευση του μοτίβου των δεδομένων μας παρόλο που δεν τα ταξινομεί όλα με απόλυτη

ακρίβεια. Συνεπώς θα ανταποκριθεί πιο αποδοτικά σε νέα άγνωστα δεδομένα. Η πράσινη γραμμή ταξινομεί με απόλυτη ακρίβεια τα δεδομένα όμως θα έχει πολύ χαμηλότερη απόδοση για νέα άγνωστα δεδομένα εφόσον δεν έχει γενικεύσει το μοτίβο αποδοτικά.

#### 1.2.6 Μέθοδοι Επικύρωσης Σφάλματος

**Holdout:** Με αυτήν την τεχνική, το σύνολο των δεδομένων μας υποδιαιρείται σε δύο διαφορετικά και ανεξάρτητα υποσύνολα. Συγκεκριμένα το αρχικό σύνολο θα χωρίστει σε ένα υποσύνολο που θα χρησιμοποιηθεί για την εκπαίδευση του μοντέλου και σε ένα υποσύνολο που θα χρησιμοποιηθεί για τον έλεγχο της απόδοσης του. Συνήθως το 80% των δεδομένων του αρχικού συνόλου χρησιμοποιούνται ως το υποσύνολο εκμάθησης και το υπόλοιπο 20% ως το υποσύνολο για τον έλεγχο.<sup>[11]</sup> Η συγκεκριμένη τεχνική εμφανίζει κάποια μειονεκτήματα. Στην περίπτωση που το αρχικό σύνολο δεδομένων μας είναι αραιό, η παράληψη ενός υποσυνόλου από την διαδικασία εκπαίδευσης μπορεί να επηρεάσει αρνητικά την απόδοση του τελικού μοντέλου. Επιπλέον δεν υπάρχει εγγύηση ότι ο διαχωρισμός του αρχικού συνόλου γίνεται ομοιόμορφα και με τρόπο τέτοιον ώστε να υπάρχει δίκαιη και αντιπροσωπευτική αναπαράσταση των κατηγοριών των δεδομένων. Για την αποφυγή των παραπάνω προβλημάτων που παρουσιάζει η τεχνική holdout χρησιμοποιούνται τεχνικές αναδειγματοληψίας όπως αυτή που ονομάζεται Cross Validation.

**Cross Validation:** Μέσω της τεχνικής Cross Validation το σύνολο των δεδομένων υποδιαιρείται σε k διαφορετικά, ανεξάρτητα και προσεγγιστικά ίσα υποσύνολα (folds). Επαναληπτικά, ένα από τα k υποσύνολα θα χρησιμοποιείται ως σύνολο δεδομένων έλεγχου, για τον έλεγχο του μοντέλου, ενώ τα υπόλοιπα k-1 για την εκπαίδευσή του. Αυτή η διαδικασία, λοιπόν, θα επαναληφθεί για κάθε τέτοιο υποσύνολο πετυχαίνοντας με αυτόν τον τρόπο μεγαλύτερη ποικιλία στα δεδομένα εισόδου. Απλουστευμένα, σε κάθε επανάληψη 1 υποσύνολο χρησιμοποιείται για έλεγχο και k-1 υποσύνολα για εκπαίδευση. Το μοντέλο εκπαιδεύεται και ελέγχεται k φορές, όσα δηλαδή και τα υποσύνολα. Το σφάλμα υπολογίζεται ως ο μέσος όρος των εκτιμήσεων που πραγματοποιήθηκαν για κάθε μία από τις k επαναλήψεις. Η συγκεκριμένη τεχνική είναι πολύ χρήσιμη ειδικά σε περιπτώσεις που τα δεδομένα δεν είναι αρκετά και χρειαζόμαστε περισσότερους συνδυασμούς από σύνολα εκπαίδευσης και έλεγχου για να πετύχουμε καλύτερη απόδοση στο μοντέλο και να αποφύγουμε το φαινόμενο του overfitting.<sup>[11]</sup> Το πλεονέκτημα της μεθόδου αυτής έναντι της τεχνικής holdout, είναι ότι γίνεται χρήση και εκμετάλλευση των δεδομένων ολόκληρου του αρχικού συνόλου. Με αυτόν τον τρόπο γίνεται ομοιόμορφος διαχωρισμός των δεδομένων, ώστε να υπάρχει αντιπροσωπευτική και δίκαιη διαίρεση των

κατηγοριών των δεδομένων. Επιπλέον όλες οι υποδιαιρέσεις χρησιμοποιούνται στην διαδικασία εκπαίδευσης του μοντέλου με αποτέλεσμα να υπάρχουν αρκετά δεδομένα ακόμα και σε προβλήματα με αραιά σύνολα δεδομένων.

### 1.2.7 Μετρικές

Η αξιολόγηση της απόδοσης των μοντέλων γίνεται μέσω κάποιων μετρικών. Μια μετρική, λοιπόν μπορεί να περιγραφεί ως εργαλείο αξιολόγησης της απόδοσης ενός μοντέλου. Υπάρχουν πολλές διαφορετικές μετρικές οι οποίες εστιάζουν σε διαφορετικές προσεγγίσεις της μέτρησης της απόδοσης λαμβάνοντας υπόψη διαφορετικά χαρακτηριστικά των μοντέλων. Τα μοντέλα classification μετρούνται με διαφορετικές μετρικές από τα regression μοντέλα. Το γεγονός αυτό είναι λογικό καθώς στην πρώτη περίπτωση το μοντέλο ταξινομεί τιμές σε συγκεκριμένες διακριτές κλάσεις ενώ στην δεύτερη περίπτωση το μοντέλο προσπαθεί να προσεγγίσει όσο πιο ακριβώς γίνεται μία πραγματική τιμή.

Οι μετρικές που χρησιμοποιούνται στην παρούσα διπλωματική εργασία για την αξιολόγηση των παραγόμενων μοντέλων παρατίθενται και περιγράφονται παρακάτω. [1] [2] Για την αξιολόγηση μοντέλων μηχανικής μάθησης κατηγορίας Classification έγινε χρήση των μετρικών:

- ▶ **Accuracy:** Με απλά λόγια η συγκεκριμένη μετρική αφορά το πλήθος των ορθών προβλέψεων του μοντέλου μας σε σχέση με το συνολικό πλήθος των προβλέψεων που πραγματοποίησε.
- ▶ **Precision:** Η συγκεκριμένη μετρική αποτελεί το πιλίκο των ορθά θετικών προβλέψεων με το σύνολο των θετικών προβλέψεων. Με απλά λόγια αφορά το ποσοστό των θετικών προβλέψεων που ήταν στην πραγματικότητα ορθές. Η χειρότερη απόδοση αντιστοιχεί στο 0 και η καλύτερη στο 1.
- ▶ **Average Precision:** Πρόκειται για τον σταθμισμένο μέσο όρο των Precisions που επιτυγχάνονται σε κάθε κατώφλι. Το βάρος που χρησιμοποιείται είναι η αύξηση στην Recall μετρική σε σχέση με τον υπολογισμό της στο προηγούμενο κατώφλι. Στον παρακάτω τύπο το  $P_n$  και  $R_n$  είναι αντίστοιχα το Precision και το Recall στο  $n$ -στο κατώφλι.

$$AP = \sum_n (R_n - R_{n-1})P_n$$

- ▶ **Logistic Loss**

- ▶ **ROC AUC:** Πρόκειται για το εμβαδόν κάτω από την καμπύλη ROC (Area Under Curve). Εάν το εμβαδόν είναι 1.0 τότε ο ταξινομητής δεν πραγματοποιεί καμία λάθος πρόβλεψη. Η καμπύλη ROC είναι το διάγραμμα του Ρυθμού Ορθά Θετικών προβλέψεων(TPR - True Positive Rate) σε σχέση με τον Ρυθμό Εσφαλμένα Θετικών προβλέψεων(FPR - False Positive Rate). Ο TPR αποτελεί το ποσοστό των πραγματηρίσεων που ταξινομήθηκαν ορθά ως θετικές σε σχέση με όλες τις θετικές πραγματηρίσεις. Ο FPR αποτελεί το ποσοστό των πραγματηρίσεων που ταξινομήθηκαν εσφαλμένα ως θετικές σε σχέση με όλες τις αρνητικές πραγματηρίσεις. Οι ταξινομητές των οποίων το ROC διάγραμμα είναι πιο κοντά στο αριστερά πάνω σημείο, είναι πιο αποδοτικοί.
- ▶ **Recall:** Η μετρική Recall είναι το πηλίκο των ορθά θετικών προβλέψεων με το σύνολο των ορθά θετικών και των εσφαλμένα αρνητικών προβλέψεων. Με απλά λόγια είναι το ποσοστό των ορθά θετικών προβλέψεων που ταυτοποιήθηκαν επιτυχώς.
- ▶ **F1 Score:** Σκοπός της μετρικής αυτής είναι η αναζήτηση της ισορροπίας μεταξύ Precision και Recall. Πρόκειται για τον αριθμονικό μέσο μεταξύ των 2 αυτών μετρικών.

$$F1 = 2 * \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}}$$

Για την αξιολόγηση μοντέλων μηχανικής μάθησης κατηγορίας Regression έγινε χρήση των μετρικών:

- ▶ **Mean / Median Absolute Error (MAE):** Πρόκειται για τον μέσο όρο της απόλυτης τιμής της διαφοράς ανάμεσα στην πραγματική τιμή και την τιμή που προέκυψε από την πρόβλεψη του μοντέλου.

$$MSE = \frac{1}{n} \sum (y - \hat{y})$$

- ▶ **Mean Squared Error (MSE):** Πρόκειται για τον μέσο όρο της τετραγωνικής διαφοράς ανάμεσα στην πραγματική τιμή και την τιμή που προέκυψε από την πρόβλεψη του μοντέλου.

$$MSE = \frac{1}{n} \sum (y - \hat{y})^2$$

- ▶ **Mean Squared Logarithmic Error** Η συγκεκριμένη μετρική αφορά αποκλειστικά τις ποσοστιαίες διαφορές μεταξύ των πραγματικών και των τιμών που προέβλεψε το μοντέλο, συνεπώς να συμπεριφέρεται ισοδύναμα στις διαφορές ανάμεσα σε μικρές και ανάμεσα σε μεγάλες τιμές.

$$MSLE = \frac{1}{N} \sum_{i=0}^n (\log(y_i + 1) - \log(\hat{y}_i + 1))^2$$

- ▶ **R2 Score** Πρόκειται για μια μετρική που μας δείχνει εάν το μοντέλο μας είναι πιο αποδοτικό σε σύγκριση με την γραμμή που απεικονίζει τον μέσο όρο των παρατηρήσεων. Η γραμμή αυτή αποτελεί σημείο αναφοράς(baseline).

$$R^2 = 1 - \frac{MSE(model)}{MSE(baseline)}$$

## 2. Αυτοματοποιημένη Μηχανική Μάθηση

### 2..1 Γενικά

Ο συνδυασμός των απαιτήσεων για την δημιουργία ενός συστήματος μηχανικής μάθησης έχει ως αποτέλεσμα την εμφάνιση εμπόδιων και δυσκολιών, που αποθαρρύνουν την ανάπτυξη του. Μερικά από αυτά τα εμπόδια είναι:

- ▶ **Χρόνος:** Ο χρόνος είναι πολύτιμος και περιορισμένος, ιδίως στην εποχή μας, όπου όλα εξελίσσονται και “τρέχουν” με ιλιγγιώδεις ρυθμούς. Η έλλειψη χρόνου είναι ένα πολύ συχνό φαινόμενο που αποτελεί βασικό εμπόδιο για την ανάπτυξη ενός συστήματος μηχανικής μάθησης.
- ▶ **Γνωστικό Υπόβαθρο:** Όπως έχει προαναφερθεί, η εμπειρία και η εξειδίκευση είναι απαραίτητα στοιχεία για την υλοποίηση καθώς και την σωστή χοίστη ενός συστήματος μηχανικής μάθησης. Το γεγονός αυτό καθιστά ένα τέτοιο σύστημα προσιτό μόνο σε άτομα με το απαραίτητο γνωστικό υπόβαθρο στην επιστήμη των υπολογιστών και ειδικότερα στην επιστήμη των δεδομένων.
- ▶ **Πόροι:** Πολλές φορές οι οικονομικοί καθώς και οι υπολογιστικοί πόροι είναι περιορισμένοι, δημιουργώντας εμπόδια στην υλοποίηση και εφαρμογή τέτοιων συστημάτων.
- ▶ **Συνδυασμός των παραπάνω:** Πολλές φορές δεν πληρούνται περισσότερες από μία απαιτήσεις.

Η αυτοματοποιημένη μηχανική μάθηση αποτελεί ένα νέο, καινοτόμο πεδίο της μηχανικής μάθησης που αναπτύσσεται με ταχείς ρυθμούς. Πρόκειται για την διαδικασία αυτοματοποίησης των επαναληπτικών και χρονοβόρων υποδιαδικασιών της δημιουργίας μοντέλων μηχανικής μάθησης. [15]

Ένα σημαντικό πλεονέκτημα της εφαρμογής του αυτοματισμού στην μηχανική μάθηση είναι η απομάκρυνση του ανθρώπινου παράγοντα από την διαδικασία και τις λειτουργίες της. Το θετικό αποτέλεσμα αυτής της αποκοπής, είναι η αύξηση της παραγωγικότητας και της αποτελεσματικότητας ενός χρήστη ή αναλυτή, καθώς του δίνεται η δυνατότητα να αφιερώσει τον χρόνο

του στην επίλυση του πραγματικού προβλήματος και στις διαδικασίες που έχουν αξία, αντί για την ολοκλήρωση τετριψένων εργασιών.

Η αυτοματοποιημένη μηχανική μάθηση:

- ▶ Απαλλάσσει τους εξειδικευμένους αναλυτές δεδομένων από τις χρονοβόρες διαδικασίες της μηχανικής μάθησης, με αποτέλεσμα να εστιάζουν σε πιο ουσιαστικές εργασίες.
- ▶ Κάνει την μηχανική μάθηση πιο προσιτή, έτσι ώστε περισσότεροι χρήστες να μπορούν να ευνοηθούν από όσα έχει να προσφέρει, ανεξάρτητα από το επαγγελματικό, εκπαιδευτικό ή γνωστικό υπόβαθρο τους.
- ▶ Μειώνει τις απαιτήσεις σε υπολογιστικούς και οικονομικούς πόρους.

Σκοπός, λοιπόν, της αυτοματοποιημένης μηχανικής μάθησης είναι η ελαχιστοποίηση των πόρων, του χρόνου και του υπολογιστικού κόστους που απαιτείται για την ανάπτυξη εφαρμογών μηχανικής μάθησης, με σκοπό την επίλυση προβλημάτων. Ένας επιπλέον στόχος της αυτοματοποιημένης μηχανικής μάθησης είναι, ταυτόχρονα, η επίτευξη της όσον το δυνατόν υψηλότερης απόδοσης του συστήματος.

Συμπερασματικά, μέσω της καινοτομίας αυτής, η ανάπτυξη εφαρμογών μηχανικής μάθησης ενθαρρύνεται και επιταχύνεται, επιτυγχάνοντας υψηλές επιδόσεις, χαμηλό υπολογιστικό και οικονομικό κόστος, χωρίς απαραίτητη προϋπόθεση την ύπαρξη εξειδικευμένου προσωπικού. [38] Εκτός από την εμφάνιση και ανάπτυξη καινοτόμων τεχνολογιών, είναι επίσης απαραίτητη στις μέρες μας η διάδοση τους στον ευρύτερο πληθυσμό. Η ευκολία στην χρήση τους και η προσβασιμότητα τους, επιτρέπει σε ανθρώπους που δεν διαθέτουν το ανάλογο τεχνολογικό και γνωστικό υπόβαθρο να επωφεληθούν από τα άλματα της τεχνολογίας εξελίσσοντας συνολικά την κοινωνία.

### 2..1.1 Υπο-διαδικασίες

Η αυτοματοποιημένη μηχανική μάθηση μπορεί να αναλυθεί σε τρεις βασικές υποδιεργασίες. Η ανάλυση αυτής ενδέχεται να συναντάται διαφοροποιημένη, όμως ο κορυφώντας είναι κοινός σε όλες τις διαφοροποιήσεις. Η ροή διαδικασιών της αυτοματοποιημένης μηχανικής μάθησης είναι ο συνδυασμός αλγορίθμων με σκοπό την αντιστοίχιση ενός διανύσματος χαρακτηριστικών σε μία τιμή. Στην παρούσα διπλωματική εργασία, η ροή διαδικασιών της αυτοματοποιημένης μηχανικής μάθησης αποτελείται από αλγορίθμους για:

1. Την προετοιμασία των δεδομένων εισόδου - Data Preparation
2. Την επεξεργασία των χαρακτηριστικών πρόβλεψης - Feature Engineering

### 3. Την επιλογή μοντέλου - Modeling

Για κάθε αλγόριθμο των παραπάνω κατηγοριών πρέπει να γίνει η ρύθμιση και βελτιστοποίηση των υπερπαραμέτρων του.

#### 2..2 Προετοιμασία-Προεπεξεργασία Δεδομένων

Η προετοιμασία και προ-επεξεργασία των δεδομένων εισόδου είναι η πρώτη και πολύ σημαντική διεργασία που πρέπει να πραγματοποιηθεί στην συνολική φάση εργασιών της δημιουργίας ενός μοντέλου μηχανικής μάθησης. Είναι γνωστό πως όσο πιο καθαρή και χρήσιμη είναι η πληροφορία των δεδομένων εισόδου, τόσο πιο ακριβείς και χρήσιμες θα είναι οι προβλέψεις του τελικού μοντέλου. Κάποιες από τις τεχνικές προ-επεξεργασίας δεδομένων είναι οι εξής:

1. Rescalling
2. Imputation
3. One Hot Encoding
4. Balancing[\[26\]](#)

#### 2..3 Επεξεργασία Χαρακτηριστικών

Σε αυτήν την κατηγορία υπάρχουν οι εξής υποκατηγορίες:

- ▶ Επιλογή Χαρακτηριστικών
- ▶ Εξαγωγή Χαρακτηριστικών
- ▶ Δημιουργία χαρακτηριστικών

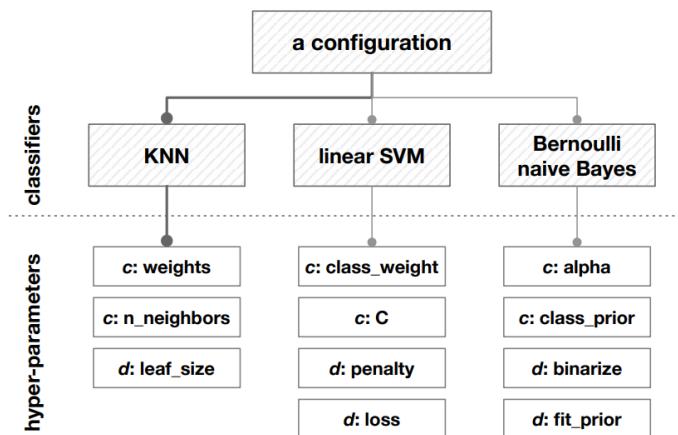
Ο σκοπός της διαδικασίας επιλογής χαρακτηριστικών είναι η απλοποίηση του χώρου μέσω της επιλογής των πιο χρήσιμων χαρακτηριστικών που έχουν σημαντικό ρόλο στην απόδοση του τελικού μοντέλου. Η απλοποίηση του μοντέλου μπορεί να οδηγήσει στην αποφυγή του overfitting και στην αύξηση της ακρίβειας. Μέσω της εξαγωγής και της δημιουργίας χαρακτηριστικών χρησιμοποιώντας τα ήδη υπάρχοντα, συνήθως αυξάνουμε τις διαστάσεις του χώρου χαρακτηριστικών με σκοπό την δυνατότητα ευκολότερης γενίκευσης των δεδομένων, για την δημιουργία ενός αποδοτικού μοντέλου. [\[26\]](#)

## 2..4 Επιλογή μοντέλου και Βελτιστοποίηση παραμέτρων

Η επιλογή και δημιουργία ενός μοντέλου αποτελείται από δύο κύρια στοιχεία:

1. Χώρος Αναζήτησης
2. Μέθοδοι Βελτιστοποίησης[26] [38]

Θα πρέπει να βρεθεί στον χώρο αναζήτησης μοντέλων, ο κατάλληλος αλγόριθμος για το πρόβλημα μας και στην συνέχεια να ρυθμιστούν οι υπερπαραμέτροι του αφενός αυτόματα και αφετέρου με τον βέλτιστο τρόπο ώστε να έχουμε ένα αποδοτικό αποτέλεσμα. Ο χώρος αναζήτησης αποτελείται από τους αλγορίθμους μηχανικής μάθησης μαζί με τις υπερ-παραμέτρους τους. Η δομή του χώρου είναι ιεραρχική με την έννοια ότι οι υπερ-παραμέτροι κάθε αλγορίθμου ρυθμίζονται μόνο στην περίπτωση που ο αντίστοιχος αλγόριθμος έχει επιλεχθεί για χρήση.



Σχήμα 2.2: Απεικόνιση του χώρου αναζήτησης μοντέλων. Παρατηρούμε πως έχει επιλεχθεί ο αλγόριθμος KNN και στην συνέχεια πρέπει να ρυθμιστούν οι υπερπαραμέτροι του.

Σκοπός της διαδικασίας είναι η ελαχιστοποίηση μιας συνάρτησης κόστους, μέσω της κατάλληλης ρύθμισης των υπερ-παραμέτρων του μοντέλου.[38]

**Βελτιστοποίηση Υπερ-παραμέτρων:** Μέσω της αυτοματοποίησης της συγκεκριμένης διαδικασίας, δύναται να επιτευχθεί μεγαλύτερη απόδοση ενώ παραχθάνει ο άνθρωπος απαλλάσσεται από αυτήν. Έχει πλέον αποδειχθεί πως η διαδικασία της ρύθμισης και βελτιστοποίησης των υπερ-παραμέτρων του μοντέλου που επιλέγεται, οδηγεί σε πιο ακριβή αποτελέσματα σε σύγκριση με την προκαθορισμένη επιλογή των τιμών των υπερ-παραμέτρων. [34] [35]

#### 2..4.1 Ensembles- Συνδυαστικά μοντέλα

Για την επίτευξη μεγαλύτερης ακρίβειας στις προβλέψεις ενός μοντέλου χρησιμοποιούνται διάφορες τεχνικές συνδυασμού μοντέλων, (ensembling). Αυτές οι τεχνικές συνδυάζουν πολλαπλά απλά μοντέλα, κάθε ένα από τα οποία συνεισφέρει με κάποιο βάρος στις τελικές προβλέψεις, με σκοπό την δημιουργία ενός συνολικού βέλτιστου μοντέλου. Τα συνδυαστικά μοντέλα, ensembles, είναι αποδοτικότερα και ακριβέστερα από τα μεμονωμένα μοντέλα από τα οποία αποτελούνται με την προϋπόθεση ότι τα επιμέρους μοντέλα είναι

1. Αποδοτικά, το κάθε ένα ξεχωριστά
2. Τα σφάλματα τους είναι ασυσχέτιστα[22]

#### 2..4.2 Μετα-μάθηση (Meta-Learning)

Όταν ο άνθρωπος ξεκινά την διαδικασία ανάπτυξης κάποιας καινούριας δεξιότητας, δεν ξεκινάει από το μηδέν, αλλά χρησιμοποιεί την εμπειρία που κατέχει από παρελθοντικές παρόμοιες δεξιότητες, με σκοπό να αποκτήσει ένα προβάδισμα στην διαδικασία της μάθησης. Αυτή η τεχνική θα μπορούσε περιγραφεί ως η εκμάθηση της διαδικασίας της μάθησης ή αλλιώς η συνειδητοποίηση του τρόπου με τον οποίο κάποιος μαθαίνει. Η μέθοδος της μετα-μάθησης μπορεί να χρησιμοποιηθεί για την διευκόλυνση της δημιουργίας υπολογιστικών μοντέλων και την επιτάχυνση της διαδικασίας της μάθησης, όπως ακριβώς και στον άνθρωπο. [23] Πρακτική εφαρμογή της μετα-μάθησης μπορεί να γίνει σε όλα τα στάδια της διαδικασίας δημιουργίας μοντέλων μηχανικής μάθησης, όπως για παράδειγμα στην προετοιμασία και προ-επεξεργασία των συνόλων δεδομένων ή στην βελτιστοποίηση των παραμέτρων των αλγορίθμων.

Πρόκειται για την διαδικασία κατά την οποία γίνονται παρατηρήσεις της συμπεριφοράς διαφορετικών προσεγγίσεων μηχανικής μάθησης σε διαφορετικά προβλήματα, με σκοπό την εκμάθηση μέσω της εμπειρίας και των μεταχαρακτηριστικών που προκύπτουν από αυτές. Τέτοια χαρακτηριστικά μπορούν να κατηγοριοποιηθούν σε:

- ▶ **Γενικά χαρακτηριστικά:** Προκύπτουν άμεσα από το αρχικό σύνολο δεδομένων εισόδου και είναι τα απλούστερα μετα-χαρακτηριστικά που μπορούν να εξαχθούν. Για παράδειγμα, μπορεί να είναι το πλήθος των δεδομένων, η φύση των δεδομένων, το πλήθος των χαρακτηριστικών προβλεψης.
- ▶ **Στατιστικά χαρακτηριστικά:** Προκύπτουν από την στατιστική ανάλυση των δεδομένων εισόδου και την εφαρμογή απλών συναρτήσεων στατιστικής πάνω σε αυτά. Για παράδειγμα μπορεί να είναι μέσος όρος, συσχέτιση, διακύμανση κ.α.

- ▶ **Χαρακτηριστικά Θεωρίας Πληροφορίας:** Αφορούν την ποσότητα της χρήσιμης πληροφορίας που μπορεί να εξαχθεί από το σύνολο δεδομένων.
- ▶ **Χαρακτηριστικά βασισμένα στο μοντέλο(Model-based):** Αφορούν χαρακτηριστικά των μοντέλων που έχουν ήδη εξεταστεί. Για παράδειγμα, οι υπερ-παράμετροι των μοντέλων που εκπαιδεύονται για την επίλυση κάποιου προβλήματος.
- ▶ **Ορόσημα (Landmarks):** Αφορούν την απόδοση πολύ απλών αλγορίθμων που επιλέγονται για την επίλυση του προβλήματος δεδομένου του αρχικού συνόλου δεδομένων. Για παράδειγμα οι αποδόσεις αλγορίθμων όπως ο 1-Nearest Neighbor.

Μέσω της εμπειρίας αυτής, η εκμάθηση πάνω σε νέα δεδομένα γίνεται πολύ γρηγορότερα, επιταχύνοντας την διαδικασία δημιουργίας μοντέλων μηχανικής μάθησης, καθώς απαιτούνται λιγότεροι δοκιμής και σφάλματος. [37]

### 3. Χρονολογικές σειρές (Timeseries)

Μια χρονοσειρά αποτελείται από μία ακολουθία σημείων στο πεδίο του χρόνου. Συνήθως τα σημεία βρίσκονται σε διαδοχικές στιγμές που απέχουν το ίδιο μεταξύ τους αποτελώντας μια διακριτού χρόνου ακολουθία δεδομένων. Η μελέτη και ανάλυση των χρονοσειρών οδηγεί στην εξαγωγή πολύ χρήσιμων συμπερασμάτων και στατιστικών στοιχείων. Στην καθημερινότητα μας, η μελέτη και αξιοποίηση των χρονοσειρών παρατηρείται στην πρόγνωση του καιρού, στην αναγνώριση προτύπων, στην οικονομία και την στατιστική κ.α.

#### 3..1 Forecasting

Βασικός στόχος της ανάλυσης χρονοσειρών είναι η πραγματοποίηση προβλέψεων για μελλοντικές τιμές που μας ενδιαφέρουν. Η διαδικασία της δημιουργίας μοντέλων, εκπαιδευμένων σε ιστορικά δεδομένα, με απώτερο σκοπό την χρήση τους για την πρόβλεψη μελλοντικών παρατηρήσεων ονομάζεται **Forecasting**.

#### 3..2 Εξαγωγή Χαρακτηριστικών

Μέσω της διαδικασίας του Rolling, δίνεται η δυνατότητα μετατροπής μιας μοναδικής χρονοσειράς σε ένα σύνολο από πολλαπλές χρονοσειρές. Κάθε μία

από τις νέες χρονοσειρές που δημιουργούνται από αυτήν την διαδικασία τελειώνουν ένα (ή n ) βήμα αργότερα από την προηγούμενη της. Η διαδικασία αυτή δημιουργεί πολλαπλές χρονοσειρές στα δεδομένα εισόδου με σκοπό να εφαρμοστεί σε αυτές η διαδικασία της εξαγωγής νέων χαρακτηριστικών που θα χρησιμοποιηθούν για την δημιουργία ενός μοντέλου. Για την καλύτερη κατανόηση της διαδικασίας ακολουθεί ένα παράδειγμα.

Έστω ότι έχουμε το παρακάτω σύνολο δεδομένων:

ID	Time	Feature	Target
1	1	6	1
2	2	7	2
3	3	545	0

Για κάθε εγγραφή του πίνακα των χαρακτηριστικών θα δημιουργηθεί μία νέα χρονοσειρά που θα αποτελείται από όλες τις προηγούμενες εγγραφές έως αυτήν. Σύμφωνα με το παράδειγμα μας ο πίνακας μετά την διαδικασία του Rolling θα έχει την εξής μορφή:

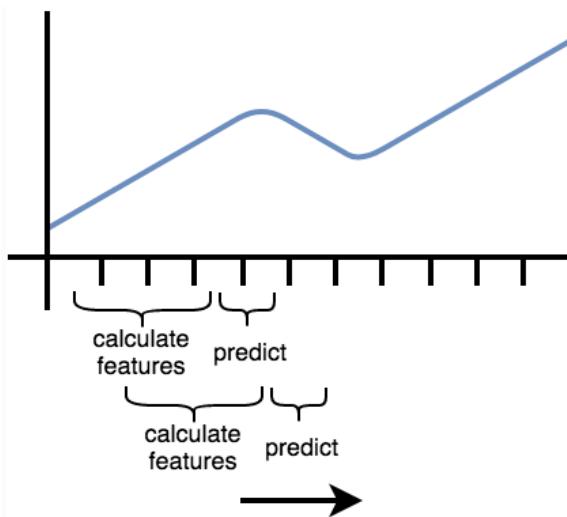
ID	Time	Feature	Target
1	1	6	1
2	1	6	
2	2	7	2
3	1	6	
3	2	7	
3	3	545	0

Παρατηρούμε πως έχουν δημιουργηθεί 3 χρονοσειρές ξεκινώντας αρχικά από μία. Αυτό μας επιτρέπει να εξάγουμε στην συνέχεια χρήσιμα χαρακτηριστικά από τις νέες χρονοσειρές. Παρατηρούμε, επιπλέον, ότι κάθε χρονοσειρά έχει ένα ID και σύμφωνα με αυτό αντιστοιχεί σε μία τιμή του target χαρακτηριστικού . Το ID αναφέρεται στην αρχική εγγραφή από την οποία προέκυψε κάθε χρονοσειρά όπως φαίνεται και στους πίνακες.

Για την εξαγωγή των χαρακτηριστικών θα εκμεταλλευτούμε τις χρονοσειρές ομαδοποιώντας τις βάσει του ID τους. Για παράδειγμα, για κάθε χρονοσειρά θα εξάγουμε τον μέσο όρο των τιμών της και θα δημιουργήσουμε ένα νέο σύνολο δεδομένων που θα μας δώσει επιπλέον χρήσιμη πληροφορία. Το τελικό σύνολο δεδομένων θα έχει την μορφή:

ID	Avg Feature	Feature	Target
1	6	6	1
2	6.5	7	2
3	186	545	0

Παρατηρούμε ότι πλέον έχουμε αφαιρέσει τον χρόνο από το σύνολο δεδομένων μας και έχουμε καταλήξει σε μία μορφή που υποδεικνύει ένα κλασσικό πρόβλημα μηχανικής μάθησης. Χρησιμοποιώντας, λοιπόν, τα εξαγόμενα χαρακτηριστικά δημιουργούμε ένα νέο σύνολο δεδομένων πάνω στο οποίο μπορούμε να εκπαιδεύσουμε μοντέλα μηχανικής μάθησης και στην συνέχεια να πραγματοποιήσουμε προβλέψεις για την Target μεταβλητή.



Σχήμα 2.3: Rolling

## 4. Γραφικό περιβάλλον

### 4..1 Γενικά

Το γραφικό περιβάλλον χρίστη, ευρέως γνωστό και με τα αρχικά GUI (Graphical User Interface), αποτελεί μια μορφή διεπαφής μεταξύ ανθρώπου και μηχανής. Πρόκειται για μια εικονική αναπαράσταση των λειτουργιών του υπολογιστή που προβάλλεται στον χρήστη.[\[27\]](#) Μέσω της διεπαφής αυτής η αλληλεπίδραση και η επικοινωνία του ανθρώπου με τον υπολογιστή διευκολύνεται. Πριν την εμφάνιση και την εφαρμογή του GUI, οι χρήστες καλούνταν να αλληλεπιδράσουν με τον υπολογιστή τους μέσω της γραμμής εντολών. Αντιθέτως, σήμερα, οι περισσότεροι χρήστες αλληλεπιδρούν με τη βοήθεια του ποντικιού και του πληκτρολογίου τους, κάνοντας κλικ πάνω σε γραφικά στοιχεία. Η καινοτομία αυτή εμφανίστηκε στις αρχές της δεκαετίας του 1970 και έφερε τον υπολογιστή πιο κοντά στον μέσο χρήστη.

### 4..2 Δομικά Στοιχεία Γραφικού Περιβάλλοντος

Τα GUIs αποτελούνται κατά βάση από 3 στοιχεία: ένα windowing σύστημα, ένα imaging μοντέλο και ένα API(application program interface - Διασύνδεση προγραμματισμού εφαρμογών). Το windowing σύστημα είναι υπεύθυνο για τα μενού και τα παράθυρα που προβάλλονται στο χρήστη. Το imaging μοντέλο ορίζει τα γραφικά και τις γραμματοσειρές. Το API, ορίζει τον τρόπο με τον οποίο τα γραφικά προβάλλονται στην οθόνη. Τα διαφορετικά GUIs μοιράζονται κοινά χαρακτηριστικά τα οποία μπορούν να περιγραφούν με τα αρχικά WIMP, τα οποία αποτελούν στοιχεία του windowing συστήματος και του imaging μοντέλου(Windows, Icons, Menus, Push-buttons - Παράθυρα, εικονίδια, μενού, κουμπιά). Μέσω αυτών, ο χρήστης αναθέτει εντολές προς εκτέλεση στον υπολογιστή, χωρίς να απαιτείται η γνώση κάποιας γλώσσας προγραμματισμού ή μηχανής.

### 4..3 Σκοπός Γραφικού Περιβάλλοντος

Ένα καλά σχεδιασμένο γραφικό περιβάλλον είναι κατανοητό, απλό στην χρήση και απαλλάσει τον χρήστη από τις δυσκολίες επικοινωνίας του με τον υπολογιστή. Ο στόχος του είναι να ενισχύει την εμπειρία χρήσης μιας εφαρμογής και να ενθαρρύνει όποιον την χρησιμοποιεί να αφοσιώνεται ολοκληρωτικά στην ανάλυση και την λύση του τρέχοντος προβλήματος. Επιπλέον, οδηγεί σε μειωμένο χρόνο εκμάθησης και αύξηση της παραγωγικότητας του χρήστη και της αποτελεσματικότητας της εφαρμογής. Αντίθετα, ένα μη σωστά δομημένο και σχεδιασμένο GUI μπορεί να οδηγήσει σε σύγχυση, μειωμένη παραγωγικότητα και αποσυγκέντρωση από το βασικό στόχο του χρήστη.[\[27\]](#)

Το γραφικό περιβάλλον αποτελεί βασικό στοιχείο της υλοποίησης ενός εργαλείου καθώς ενισχύει την εμπειρία χρήσης του, βελτιώνει την εμφάνιση του και αυξάνει την προσβασιμότητα του, εφόσον είναι προσιτό από ένα ευρύτερο κοινό χρηστών.

## 5. Επίλογος

Το παρόν κεφάλαιο αποτελεί το θεωρητικό υπόβαθρο της διπλωματικής. Στο επόμενο κεφάλαιο περιγράφεται ο σχεδιασμός και η αρχιτεκτονική του εργαλείου που υλοποιήθηκε στο πλαίσιο της παρούσας εργασίας.

# Κεφάλαιο 3

## Σχεδιασμός

Στο παρόν κεφάλαιο περιγράφεται ο σχεδιασμός και η αρχιτεκτονική του εργαλείου. Το επόμενο κεφάλαιο αφορά στην υλοποίηση του, εμπεριέχοντας οδηγούς εγκατάστασης, περιγραφές των βασικών τεχνολογιών που χρησιμοποιήθηκαν και δομικών στοιχείων του κώδικα.

### 1. Υποστηριζόμενες λειτουργίες

Σκοπός της παρούσας διπλωματικής εργασίας ήταν η υλοποίηση ενός εργαλείου αυτοματοποιημένης μηχανικής μάθησης για IoT εφαρμογές που παρέχει τις παρακάτω δυνατότητες στους χρήστες:

- ▶ Εισαγωγή αρχείου τύπου csv(comma separated values), επιλέγοντάς το από το σύστημα αρχείων του υπολογιστή ή παρέχοντας το path στο οποίο βρίσκεται.
- ▶ Επιλογή τύπου προβλήματος μηχανικής μάθησης ανάμεσα σε Classification, Regression ή Timeseries.
- ▶ Επιλογή target και predictor χαρακτηριστικών με σκοπό την δημιουργία υπολογιστικού μοντέλου μηχανικής μάθησης.
- ▶ Δυνατότητα δημιουργίας πολλαπλών χρονοσειρών από δεδομένα μιας χρονοσειράς, με μήκος παραθύρου που ορίζεται από τον χρήστη.
- ▶ Εξαγωγή νέων χρήσιμων χαρακτηριστικών από πολλαπλές χρονοσειρές και δυνατότητα αποθήκευσης του νέου συνόλου δεδομένων σε αρχείο τύπου csv.
- ▶ Μετατροπή προβλημάτων Timeseries σε Regression ή Classification προβλήματα.

- ▶ Υποστήριξη λειτουργίας forecasting για την πρόβλεψη μελλοντικών τιμών σε χρονικό διάστημα που ορίζεται από τον χρήστη.
- ▶ Ρύθμιση παραμέτρων και προτιμήσεων και δημιουργία μοντέλων μηχανικής μάθησης υπό επίβλεψη.
- ▶ Αποθήκευση των παραγόμενων μοντέλων σε βάση δεδομένων τύπου SQLite.
- ▶ Δυνατότητα προβολής των αποθηκευμένων μοντέλων και εμφάνιση σχετικών λεπτομερειών.
- ▶ Δυνατότητα χρήσης των αποθηκευμένων μοντέλων για την πραγματοποίηση προβλέψεων πάνω σε νέα σύνολα δεδομένων
- ▶ Εξαγωγή και αποθήκευση των παραγόμενων μοντέλων μηχανικής μάθησης σε κωδικοποιημένα αρχεία μορφής pickle.
- ▶ Εξαγωγή μοντέλων σε συμβατή μορφή για την υποστήριξη από μικροελεγκτές.

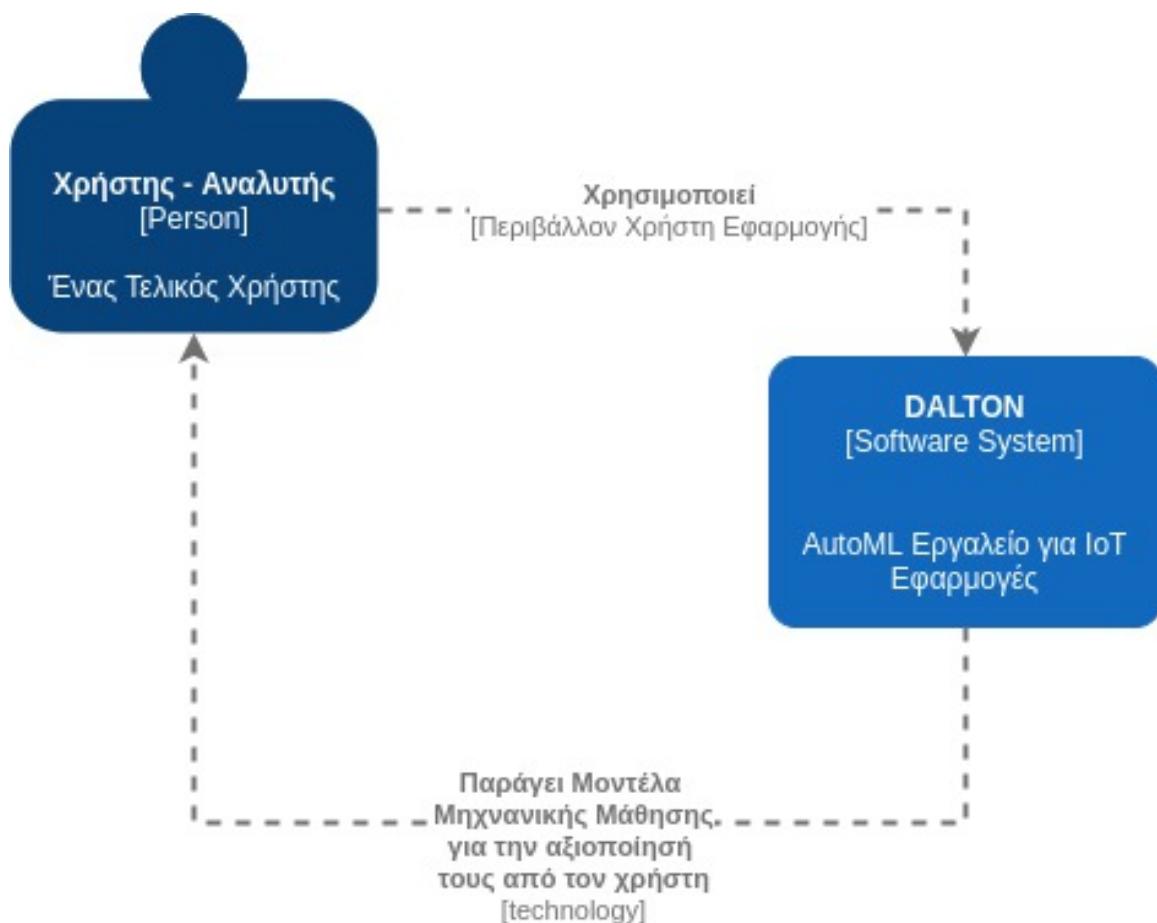
## 2. Ανάλυση και Περιγραφή Αρχιτεκτονικής

Η αρχιτεκτονική λογισμικού του συστήματος μπορεί να χαρακτηριστεί ως event-driven και παρατηρείται πολύ συχνά σε εφαρμογές με γραφικό περιβάλλον και απαιτούν αλληλεπίδραση με τον χρήστη. Ο event-driven προγραμματισμός χαρακτηρίζεται από το γεγονός ότι η ροή του προγράμματος εξαρτάται από συγκεκριμένα συμβάντα όπως για παράδειγμα, το κλικ του ποντικιού ή το πάτημα ενός κουμπιού από τον χρήστη. Η εφαρμογή ανιχνεύει αυτά τα γεγονότα, και μέσω μιας διαδικασίας διαχείρισης γεγονότων, πυροδοτεί συγκεκριμένες ενέργειες που είναι συσχετισμένες με αυτά. Η κεντρική διαδικασία διαχείρισης συμβάντων είναι συνήθως ένας βρόγχος ο οποίος εκτελείται διαρκώς στο παρασκήνιο ανιχνεύοντας καινούρια συμβάντα. Με την ανίχνευση ενός νέου συμβάντος, πρέπει να αποφασιστεί και να κληθεί η συσχετισμένη με το συμβάν, διαδικασία διαχείρισης του, που με τη σειρά της θα πυροδοτήσει την κατάλληλη ενέργεια (π.χ. ένα μπλοκ κώδικα που υλοποιεί μία λειτουργία). Μέσω της Python βιβλιοθήκης PyQt παρέχεται ένας μηχανισμός παραγωγής και διαχείρισης συμβάντων που ονομάζεται Signals and Slots.

- ▶ **Signal:** Ένα σήμα που παράγεται όταν ένα συγκεκριμένο συμβάν παρατηρείται. Συνδέεται με ένα Slot.
- ▶ **Slot:** Συνδέεται με ένα συγκεκριμένο Signal. Πρόκειται για οποιοδήποτε στοιχείο Python, που μπορεί να κληθεί και να εκτελεστεί (για παράδειγμα μία συνάρτηση) κατά την παρατήρηση του αντίστοιχου συμβάντος.

Στην εφαρμογή που αναπτύχθηκε, τα events παράγονται από τους χρήστες που αλληλεπιδρούν με αυτήν μέσω των στοιχείων του γραφικού περιβάλλοντος (κομπιά, περιοχές εισαγωγής κειμένου/αρχείων, widgets). Μόλις παρατηρηθεί μία τέτοια αλληλεπίδραση, ο αντίστοιχος κώδικας Python εκτελείται στο παρασκήνιο πυροδοτώντας συγκεκριμένες λειτουργίες της εφαρμογής. [31] [13] [17]

Στο σχήμα 3.1 αποτυπώνεται το πρώτο επίπεδο αρχιτεκτονικής της εφαρμογής όπου αποτελεί την πιο αφαιρετική αναπαράσταση της. Παρατηρούμε ότι ο τελικός χρήστης αλληλεπιδρά με το γραφικό περιβάλλον της εφαρμογής. Η εφαρμογή έχει ως βασική λειτουργία την δημιουργία μοντέλων μηχανικής μάθησης, ανάμεσα σε άλλες δευτερεύουσες, αλλά χρήσιμες λειτουργίες, με σκοπό την αξιοποίησή τους από τον τελικό χρήστη.



Σχήμα 3.1: Αρχιτεκτονική εφαρμογής σε υψηλό επίπεδο αφαίρεσης

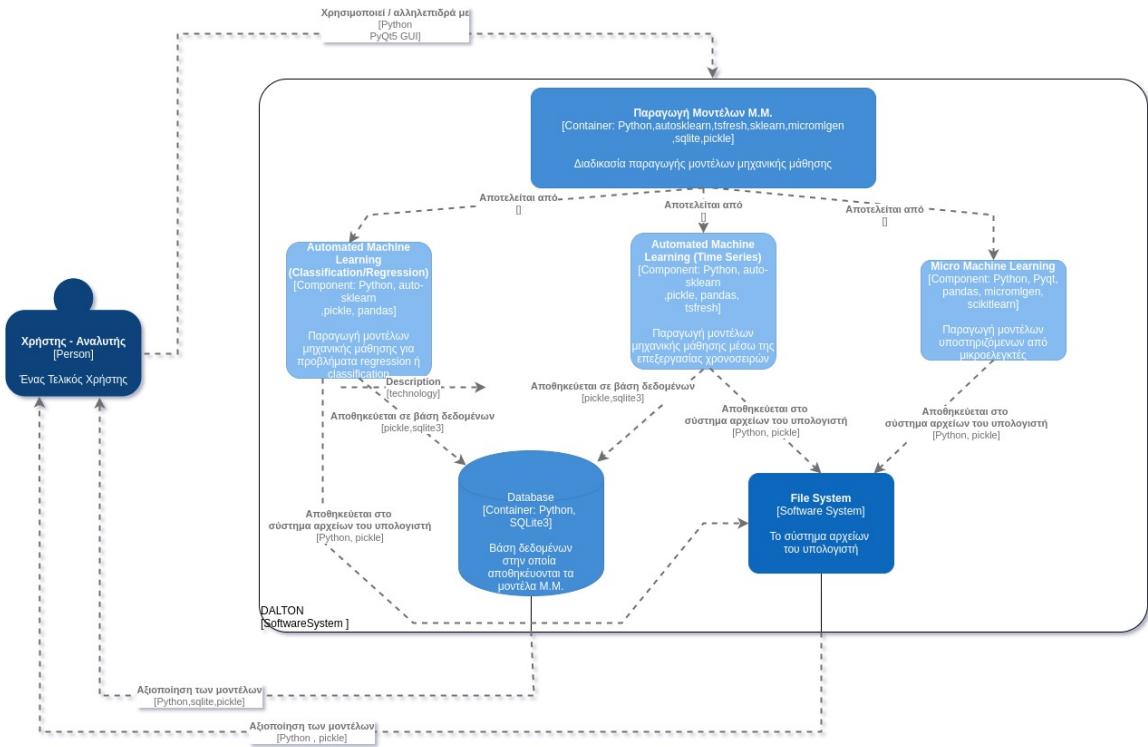
Το σχήμα 3.2 αποτελεί ένα δεύτερο επίπεδο αφαίρεσης της αρχιτεκτονικής

της εφαρμογής. Περισσότερα στοιχεία της εφαρμογής είναι εμφανή στο σχήμα και διακρίνονται κάποιες υπο-διαδικασίες που εκτελούνται. Ο χρήστης στην συγκεκριμένη περίπτωση αλληλεπιδρά όπως και πριν με την εφαρμογή μέσω του γραφικού περιβάλλοντος χρήστη. Παρατηρούμε πως η κύρια λειτουργία της παραγωγής μοντέλων μηχανικής μάθησης διακρίνεται πλέον σε 3 υποδιαδικασίες που αποτελούν τις βασικές λειτουργίες του εργαλείου. Οι υποδιαδικασίες είναι οι παρακάτω:

1. Παραγωγή μοντέλων αυτοματοποιημένης μηχανικής μάθησης για κατηγορίες προβλημάτων Classification ή Regression(Auto-ML)
2. Παραγωγή μοντέλων αυτοματοποιημένης μηχανικής μάθησης για κατηγορίες προβλημάτων Time Series(Auto-ML)
3. Παραγωγή μοντέλων μηχανικής μάθησης με σκοπό την εφαρμογή τους σε μικρο-ελεγκτές(Micro-ML)

Οι δύο πρώτες διαδικασίες που αναφέρονται στην λίστα διαφέρουν μόνο στον τρόπο διαχείρισης των δεδομένων εισόδου, καθώς όταν πρόκειται για πρόβλημα χρονοσειρών, απαιτούνται πρώτα κάποιες ενέργειες επεξεργασίας και παραμετροποίησης με σκοπό το πρόβλημα να μετατραπεί σε κοινό πρόβλημα ταξινόμησης. Για παράδειγμα, η διαδικασία του Rolling και της Εξαγωγής νέων Χαρακτηριστικών από τα δεδομένα εισόδου υλοποιείται μόνο στην περίπτωση των χρονοσειρών πριν την τελική δημιουργία του μοντέλου μηχανικής μάθησης.

Στις δύο πρώτες διαδικασίες το παραγόμενο μοντέλο αποθηκεύεται στο σύστημα αρχείων του υπολογιστή το τελικού χρήστη καθώς και στην βάση δεδομένων της εφαρμογής μας. Στην τρίτη διαδικασία που αφορά στο Micro-ML, το παραγόμενο μοντέλο αποθηκεύεται με την μορφή ενός .h, C αρχείου μόνο στο σύστημα αρχείων του υπολογιστή. Ο χρήστης μπορεί να ανακτήσει και να αξιοποιήσει τα μοντέλα που βρίσκονται στο σύστημα αρχείων του υπολογιστή ή στην τοπική βάση δεδομένων της εφαρμογής.

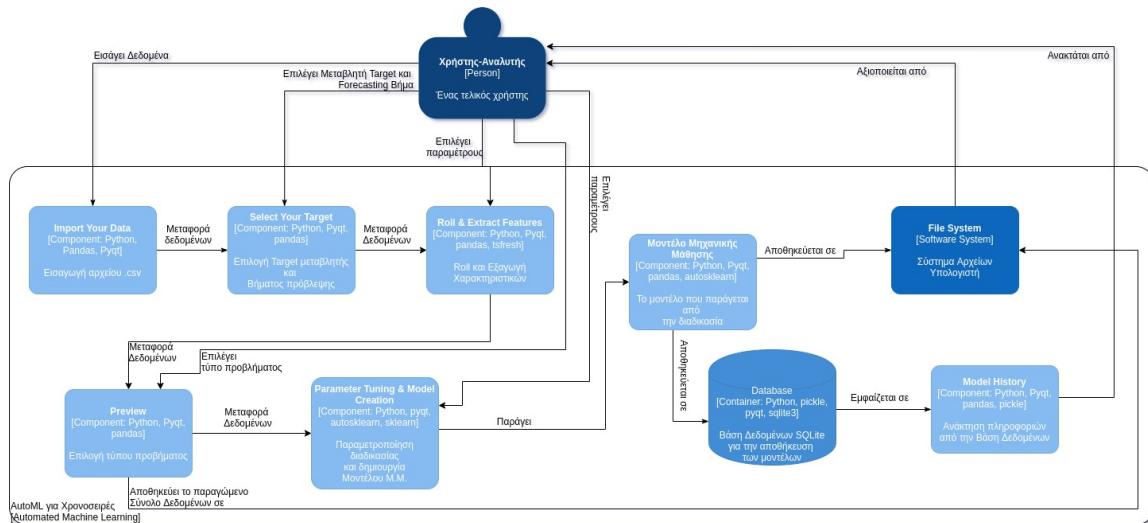


Σχήμα 3.2: Αρχιτεκτονική Εφαρμογής Δεύτερου Επιπέδου Αφαίρεσης

Τα σχήματα 3.3, 3.4 και 3.5 αποτελούν το τρίτο επίπεδο αφαίρεσης της αρχιτεκτονικής της εφαρμογής. Τα σχήματα αυτά αναπαριστούν πιο λεπτομερώς την αρχιτεκτονική των τριών βασικών υπο-διαδικασιών που είδαμε στο δεύτερο επίπεδο αφαίρεσης, που αναπαριστάται στο σχήμα 3.2, και περιλαμβάνουν τα βασικά στοιχεία τους. Ο χρήστης αλληλεπιδρά με την εφαρμογή μέσω του γραφικού περιβάλλοντος και των έχει ωριστών οδονών που την αποτελούν. Τα βασικά στοιχεία των διαδικασιών σχεδόν ταυτίζονται με τις διακριτές οδόνες του συστήματος που υλοποιούν βασικές ενέργειες. Στην συνέχεια του κεφαλαίου περιγράφεται κάθε οδόντη έχει ωριστά με τις διαθέσιμες επιλογές που περιλαμβάνει.

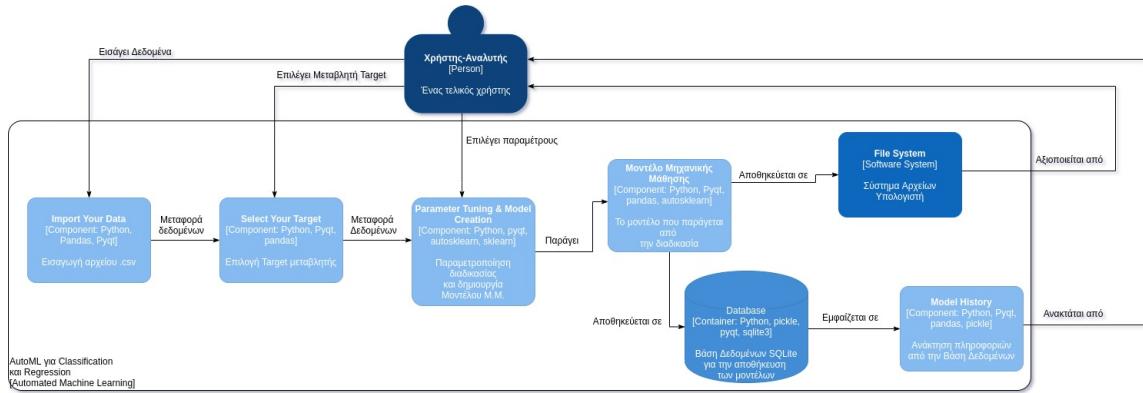
Το σχήμα 3.3 αφορά την διαδικασία δημιουργίας μοντέλων μηχανικής μάθησης όταν το σύνολο δεδομένων είναι μια χρονοσειρά. Ο χρήστης αρχικά πρέπει να εισάγει το σύνολο δεδομένων στο σύστημα. Στην συνέχεια πρέπει να επιλέξει την target μεταβλητή από το σύνολο των δεδομένων ώστε να διαχωρίστούν τα χαρακτηριστικά. Επόμενη κομβική λειτουργία είναι η διαδικασία του Rolling και της Έξαγωγής Χαρακτηριστικών. Αυτή η διαδικασία δημιουργεί ένα νέο σύνολο δεδομένων, που αποτελείται από τα εξαγόμενα χαρακτηριστικά και το επιλεγμένο target χαρακτηριστικό, και μπορεί να αποθηκευτεί στο σύστημα αρχείων του υπολογιστή του χρήστη για μετέπειτα χρήση και αξιοποίηση. Ο

χρήστης έχει την δυνατότητα να δει το νέο σύνολο δεδομένων και να επιλέξει τον νέο τύπο προβλήματος ανάμεσα σε Regression και Classification. Στην συνέχεια θα κληθεί να παραμετροποιήσει την διαδικασία δημιουργίας του μοντέλου μηχανικής μάθησης το οποίο θα αποθηκευτεί προαιρετικά στην τοπική βάση δεδομένων και στο σύστημα αρχείων του υπολογιστή από όπου μπορεί να το ανακτήσει και να το αξιοποιήσει. Στην περίπτωση που θέλει να δει πληροφορίες για το μοντέλο, αυτό γίνεται μέσω της οθόνης Model History που ανακτά το μοντέλο από την βάση δεδομένων.



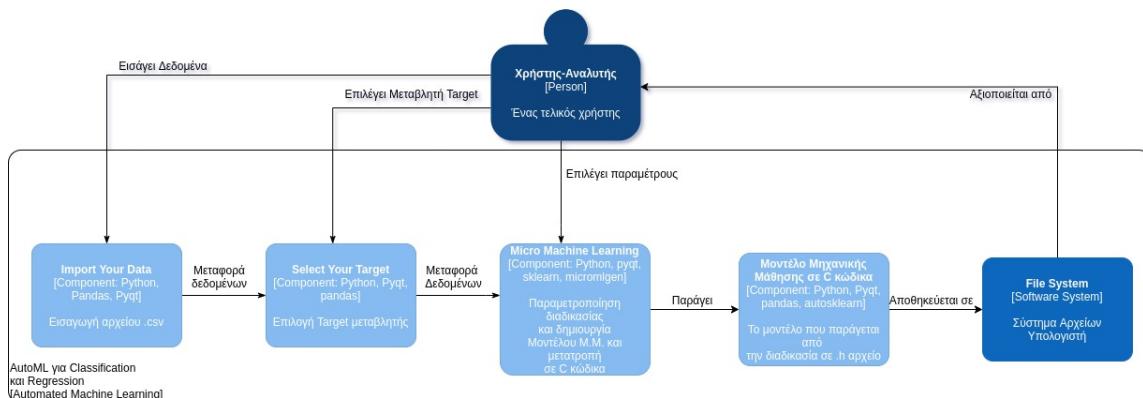
Σχήμα 3.3: Αρχιτεκτονική Εφαρμογής Τρίτου Επιπέδου Αφαίρεσης - Χρονοσειρές

Το σχήμα 3.4 αφορά την διαδικασία δημιουργίας μοντέλων μηχανικής μάθησης όταν η κατηγορία προβλήματος είναι Regression ή Classification. Η διαδικασία αυτή είναι πιο απλή καθώς το σύνολο δεδομένων εισόδου δεν χρειάζεται ιδιαίτερη μεταχείριση πριν την τελική διαδικασία παραγωγής του μοντέλου μηχανικής μάθησης, όπως συμβαίνει στην περίπτωση των χρονοσειρών. Αρχικά ο χρήστης πρέπει να εισάγει το σύνολο δεδομένων στο σύστημα. Στην συνέχεια πρέπει να οριστεί το target χαρακτηριστικό. Σε επόμενο βήμα βασική είναι η παραμετροποίηση της διαδικασίας δημιουργίας του μοντέλου και τέλος η παραγωγή, η αποθήκευσή και η ανάκτησή του όπως και στην περίπτωση των χρονοσειρών.



Σχήμα 3.4: Αρχιτεκτονική Εφαρμογής Τρίτου Επιπέδου Αφαίρεσης - Regression/Classification

Το σχήμα 3.5 αφορά την διαδικασία δημιουργίας μοντέλων μηχανικής μάθησης όταν η κατηγορία προβλήματος Classification και σκοπός είναι η εξαγωγή του μοντέλου σε μορφή υποστηριζόμενη από μικρο-ελεγκτές. Στην περίπτωση αυτή ο χρήστης αλληλεπιδρώντας με το γραφικό περιβάλλον της εφαρμογής, καλείται να εισάγει το σύνολο δεδομένων στην εφαρμογή και στην συνέχεια να επιλέξει το target χαρακτηριστικό όπως και στις παραπάνω δύο περιπτώσεις. Στην συνέχεια ο χρήστης παραμετροποιεί την διαδικασία μοντελοποίησης και μέσω της εφαρμογής εξάγει σε C κώδικα το μοντέλο μηχανικής μάθησης. Το παραγόμενο μοντέλο αποθηκεύεται σε αρχείο τύπου .h στο σύστημα αρχείων του υπολογιστή για την αξιοποίησή του από τον χρήστη.



Σχήμα 3.5: Αρχιτεκτονική Εφαρμογής Τρίτου Επιπέδου Αφαίρεσης - Μικρο-ελεγκτές

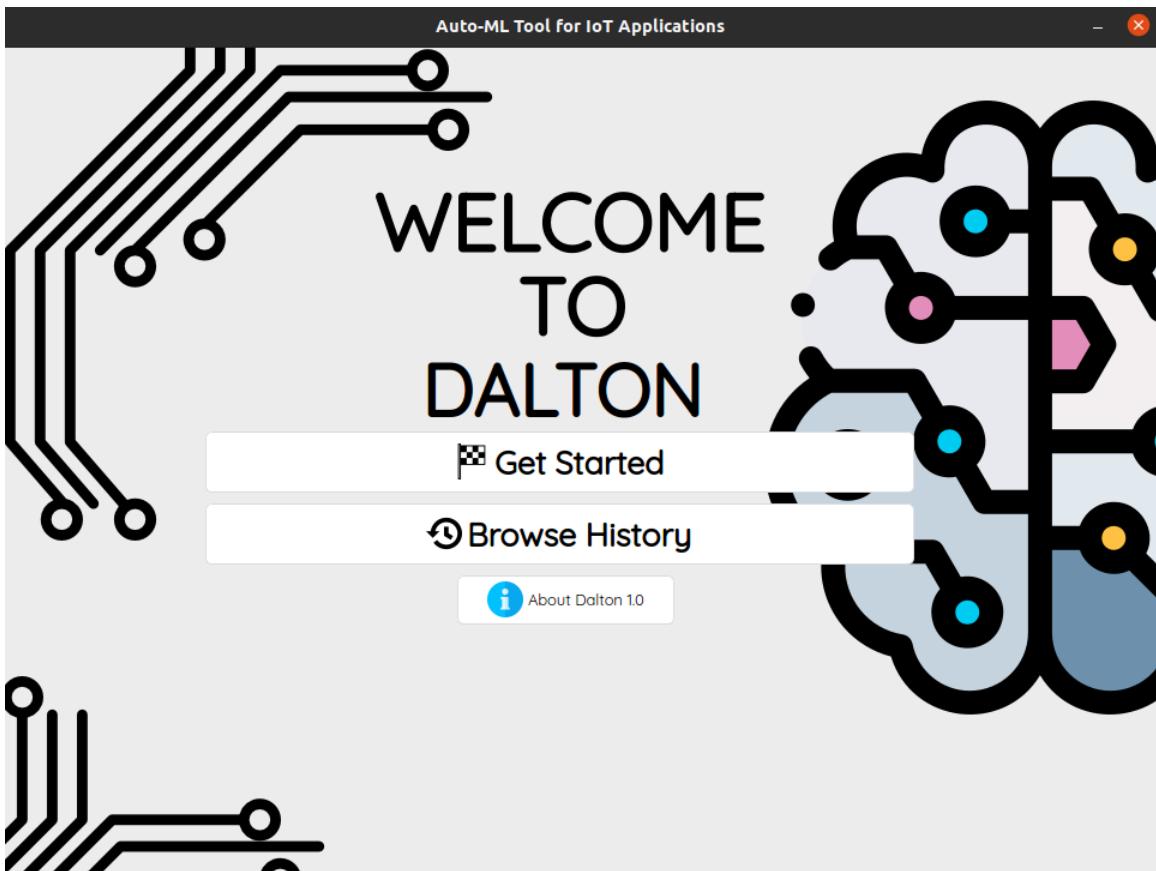
### 3. Περιγραφή Εφαρμογής

Η μορφή και ο τρόπος ροής του εργαλείου που αναπτύχθηκε, έχει αρκετά κοινά χαρακτηριστικά με μία εφαρμογή τύπου wizard. Μια Wizard εφαρμογή είναι μία διαδικασία που διαχωρίζεται, παρατίθεται σε βήματα και δίνει την δυνατότητα στους χρήστες να παρέχουν σε αυτήν δεδομένα και πληροφορίες μέσω επιλογών στην οθόνη. Βασικό χαρακτηριστικό αυτής της μορφής σχεδιασμού είναι ότι αποτελείται από πολλαπλές οθόνες που αντιστοιχούν στα επιμέρους βήματα της διαδικασίας. Επιπλέον, η σειρά με την οποία εμφανίζονται οι οθόνες (βήματα) στον χρήστη είναι συγκεκριμένη και προκαθορισμένη. Μετά την εισαγωγή της πληροφορίας, το σύστημα υπολογίζει και οδηγεί τον χρήστη σε ένα συγκεκριμένο μονοπάτι βημάτων ανάλογα με τις επιλογές που κάνει σε κάθε βήμα. Με αυτόν τον τρόπο, υπάρχει μια λογική διακλαδώσεων στο σύστημα, όμως η βασική ροή είναι κατά κύριο λόγο γραμμική, εφόσον η μία οθόνη (υπο-διαδικασία) διαδέχεται την επόμενη με τον χρήστη να προχωράει μπροστά σε κάθε βήμα. Συνήθως, τα επιμέρους βήματα είναι αλληλένδετα μεταξύ τους, υπό την έννοια ότι η πληροφορία που δίνεται στην εφαρμογή σε κάποιο βήμα, είναι απαραίτητη για το επόμενο και κατ' επέκταση για την σωστή ροή και ολοκλήρωση της συνολικής διεργασίας. [21]

Το εργαλείο που αναπτύχθηκε κατά την εκπόνηση της εργασίας αποτελείται από ένα σύνολο διαδοχικών οθονών με τις οποίες ο χρήστης αλληλεπιδρά. Κάθε οθόνη αποτελεί μία υπό-διαδικασία μιας βασικής διαδικασίας που υλοποιείται στο σύστημα. Η σειρά εμφάνισης των οθονών έχει μελετηθεί και σχεδιαστεί με τέτοιο τρόπο, ώστε να διασφαλίζεται η εύκολη και σωστή πλοήγηση του χρήστη στην εφαρμογή, καθώς και η ορθή λειτουργία των διαδικασιών που υποστηρίζονται από αυτήν. Η πλοήγηση στην εφαρμογή και η εναλλαγή οθονών πραγματοποιείται με κάποιο κουμπί όπως για παράδειγμα τα Next και Back κουμπιά. Στη συνέχεια παρατίθενται οι διαφορετικές οθόνες της εφαρμογής που αντιπροσωπεύουν διαφορετικές υποδιαδικασίες της.

#### 3..1 “Welcome” Οθόνη

Η οθόνη αυτή είναι η αρχική οθόνη του συστήματος. Σκοπός της είναι το καλωσόρισμα του χρήστη στην εφαρμογή και η προβολή των επιλογών που μπορεί να πραγματοποιήσει.



Σχήμα 3.6: Welcome Οθόνη

Συγκεκριμένα οι δυνατότητες που έχει ο χρήστης είναι οι εξής:

- ▶ Να ξεκινήσει την διαδικασία δημιουργίας ενός μοντέλου, μέσω του **Get Started** κουμπιού.
- ▶ Να μεταφερθεί στην οθόνη Model History, μέσω του **Model History** κουμπιού.
- ▶ Να μεταφερθεί στην οθόνη About, στην οποία υπάρχουν πληροφορίες για την εφαρμογή, μέσω του **About** κουμπιού.

### 3..2 “Model History” Οθόνη

Με την επιλογή **Browse Model History**, ο χρήστης μεταφέρεται στην οθόνη Model History. Ο χρήστης έχει την δυνατότητα να επιλέξει κάποιο από τα αποθηκευμένα μοντέλα της λίστας, ανάλογα με την κατηγορία μηχανικής

μάθησης στην οποία ανήκει και να προβάλει πληροφορίες σχετικές με αυτό μέσω της επιλογής Show Model Summary. Επιπλέον δύναται να δει αναλυτικά τα επιμέρους μοντέλα που απαρτίζουν το σύνθετο μοντέλο που έχει επιλέξει μέσω της επιλογής Show more... .

The screenshot shows the 'Auto-ML Tool for IoT Applications' interface with a title bar 'Auto-ML Tool for IoT Applications'. Below it is a section titled 'MODEL HISTORY' with a house icon. There are two tabs: 'Classification Models' (selected) and 'Regression Models'. A 'Show Model Summary' button is present. On the left, there is a table with three rows:

Model Name	TimeStamp
1 unnamed_model	2021-05-19 18:40:13.739250
2 iris_model	2021-05-20 13:00:28.497045

Row 3 ('heart\_disease') is highlighted. To the right of the table is a large text area containing 'auto-sklearn results:' followed by several metrics:

```
auto-sklearn results:  
Dataset name: heart  
Metric: accuracy  
Best validation score: 0.875000  
Number of target algorithm runs: 9  
Number of successful target algorithm runs: 9  
Number of crashed target algorithm runs: 0  
Number of target algorithms that exceeded the time limit: 0  
Number of target algorithms that exceeded the memory limit: 0
```

A 'Show more...' button is located at the bottom of this text area.

Σχήμα 3.7: Model History Οθόνη

**MODEL HISTORY**

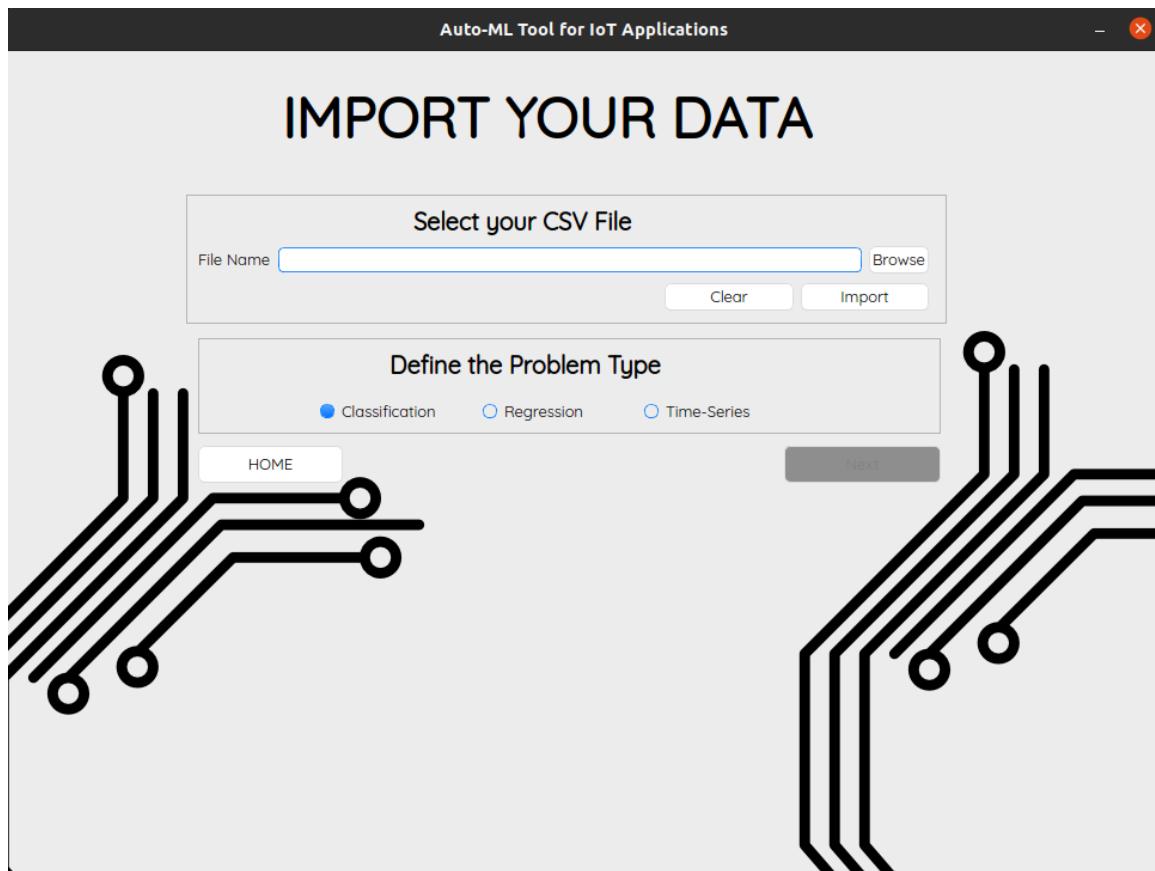
**MODEL LIST**

Weight	Information
1   0.240000000...	<pre>SimpleClassificationPipeline({'balancing:strategy': 'weighting', 'classifier:_choice_': 'random_forest', 'data_preprocessing:categorical_transformer:categorical_encoding:_choice_': 'one_hot_encoding', 'data_preprocessing:categorical_transformer:category_coalescence:_choice_': 'minority_coalescer', 'data_preprocessing:numerical_transformer:imputation:strategy': 'most_frequent', 'data_preprocessing:numerical_transformer:rescaling:_choice_': 'standardize', 'feature_preprocessor:_choice_': 'random_trees_embedding', 'classifier:random_forest:bootstrap': 'False', 'classifier:random_forest:criterion': 'entropy', 'classifier:random_forest:max_depth': 'None', 'classifier:random_forest:max_features': 0.5718514283457562, 'classifier:random_forest:max_leaf_nodes': 'None', 'classifier:random_forest:min_impurity_decrease': 0.0, 'classifier:random_forest:min_samples_leaf': 7, 'classifier:random_forest:min_samples_split': 6, 'classifier:random_forest:min_weight_fraction_leaf': 0.0, 'data_preprocessing:categorical_transformer:category_coalescence:minority_coalescer:minimum_fraction': 0.0018103963790451537, 'feature_preprocessor:random_trees_embedding:bootstrap': 'False', 'feature_preprocessor:random_trees_embedding:max_depth': 10, 'feature_preprocessor:random_trees_embedding:max_leaf_nodes': 'None', 'feature_preprocessor:random_trees_embedding:min_samples_leaf': 9, 'feature_preprocessor:random_trees_embedding:min_samples_split': 15, 'feature_preprocessor:random_trees_embedding:min_weight_fraction_leaf': 1.0, 'feature_preprocessor:random_trees_embedding:n_estimators': 96}, dataset_properties={ 'task': 1, 'sparse': False, 'multilabel': False, 'multiclass': False, 'target_type': 'classification'}</pre>

Back

Σχήμα 3.8: Model History Λιστά

### 3..3 “Import your Data” Οθόνη



Σχήμα 3.9: Import your Data Οθόνη

Με την επιλογή **Get Started**, ο χρήστης μεταφέρεται στην οθόνη Import Your Data. Στην κορυφή της οθόνης, η εφαρμογή δίνει την δυνατότητα στον χρήστη να επιλέξει ένα αρχείο από τα αρχεία του υπολογιστή του, το οποίο θα χρησιμοποιηθεί ως το dataset που θα αποτελέσει την είσοδο στο σύστημα. Αυτό πραγματοποιείται μέσω του κουμπιού Browse, με το πάτημα του οποίου εμφανίζεται ένα παράθυρο στο οποίο ο χρήστης μπορεί να περιγραφεί στα αρχεία του και να επιλέξει αυτό που επιθυμεί. Το αρχείο θα πρέπει να είναι τύπου .csv. Ένας άλλος τρόπος επιλογής αρχείου είναι μέσω της δίλωσης του PATH του αρχείου βάσει συστήματος διαχείρησης αρχείων. Αυτή η δίλωση μπορεί να γίνει στο πεδίο κειμένου δίπλα στο File Name. Μετά την επιλογή του αρχείου, για να φορτωθεί στην εφαρμογή ο χρήστης θα πρέπει να πατήσει το κουμπί Import. Τότε ένα μήνυμα θα προβληθεί που θα υποδηλώνει πως το αρχείο είναι έγκυρο και οτι η εισαγωγή του έγινε επιτυχώς. Σε κάθε άλλη

περίπτωση, θα εμφανιστεί μίνυμα σφάλματος και ο χρήστης θα πρέπει να προσπαθήσει ξανά.

Στη συνέχεια θα πρέπει να δηλωθεί ο τύπος του προβλήματος μπχανικής μάθησης. Αυτό πραγματοποιείται στο πλαίσιο Define the Problem Type, μέσω τριών radio κουμπιών. Οι διαθέσιμες επιλογές είναι οι εξής:

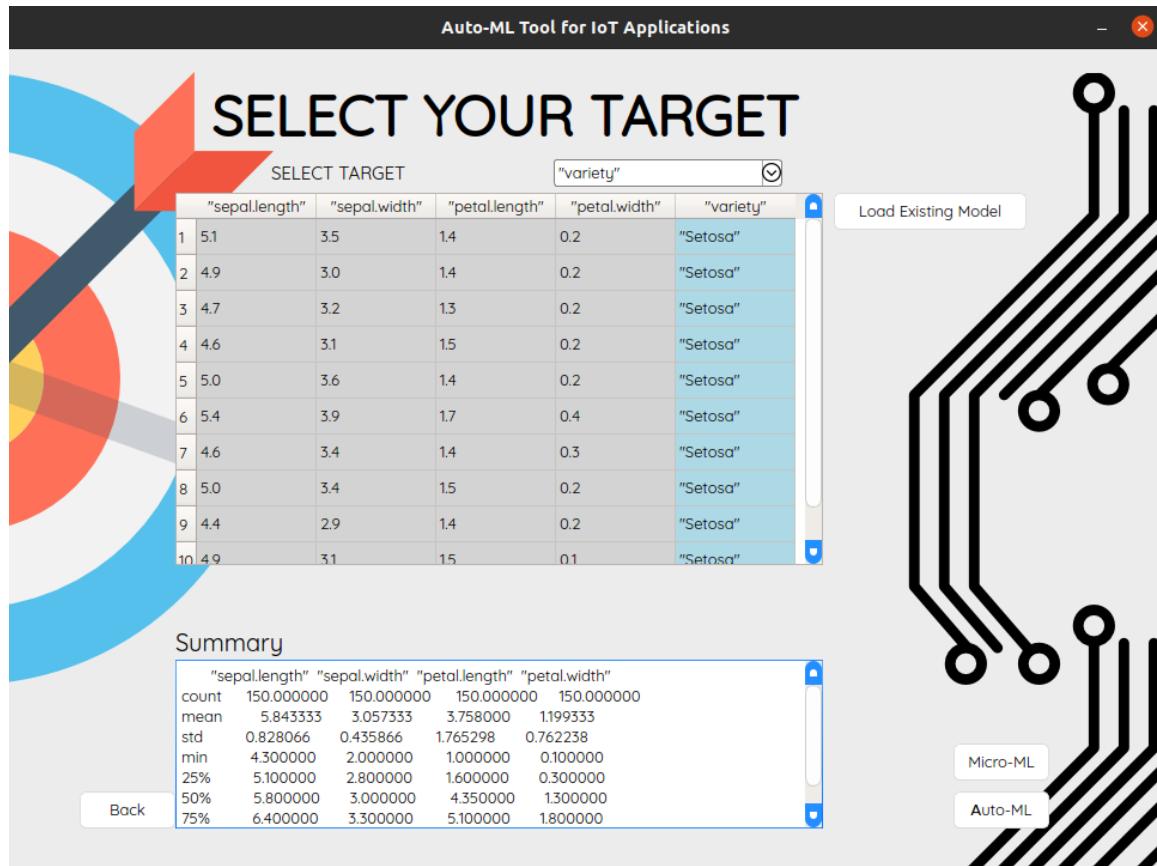
1. Classification
2. Regression
3. Time Series

Μόνο μία επιλογή μπορεί να γίνει σε αυτό το σημείο.

Όταν το αρχείο και ο τύπος του προβλήματος έχουν οριστεί, ο χρήστης μπορεί να προχωρήσει την διαδικασία μέσω του Next κουμπιού. Ανά πάσα στιγμή έχει την δυνατότητα να επιστρέψει στην αρχική οθόνη της εφαρμογής μέσω του HOME κουμπιού.

## 3..4 “Select Your Target” Οθόνη

### 3..4.1 Τύπος προβλήματος: Classification/Regression



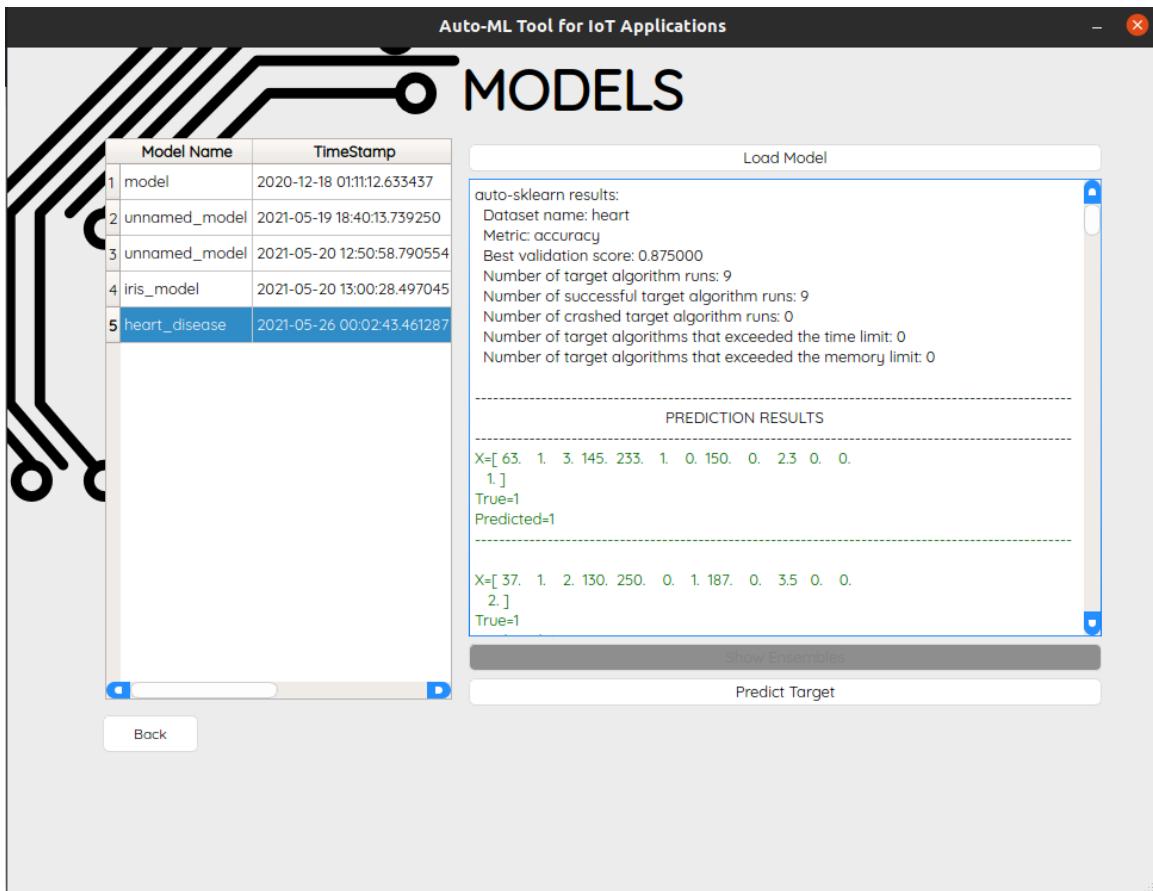
Σχήμα 3.10: Select Target Οθόνη

Στην παραγράφο αυτή περιγράφεται η λειτουργία της οθόνης στην περίπτωση που ο τύπος του προβλήματος έχει οριστεί ως (Classification / Regression). Στην συγκεκριμένη οθόνη, εμφανίζεται, ενδεικτικά, σε έναν πίνακα, η μορφή και κάποιο μέρος των περιεχομένων του αρχείου που επιλέχθηκε και εισήχθη στην εφαρμογή. Ο χρήστης καλείται να επιλέξει, μέσω μιας λίστας, ποια θα είναι η στήλη που θα αποτελεί το Target χαρακτηριστικό. Αυτή η λίστα εμφανίζεται μετά την επιλογή του κουμπιού που βρίσκεται αριστερά της ετικέτας SELECT TARGET, και περιέχει όλες τις στήλες του αρχείου με το όνομα τους. Μόλις ο χρήστης διαλέξει μία από τις επιλογές, τότε η αντίστοιχη στήλη θα αλλάξει χρώμα και θα έχει επιλεχθεί ως το Target χαρακτηριστικό. Επιπλέον, στο κάτω μέρος της οθόνης εμφανίζεται ένα σύνολο πληροφοριών

σχετικών με το επιλεγμένο dataset. Για κάθε στήλη του dataset εμφανίζονται τα εξής στατιστικά δεδομένα:

- ▶ **count** : Το πλήθος των τιμών της στήλης.
- ▶ **mean**: Η μέση τιμή της στήλης.
- ▶ **std**: Η τυπική απόκλιση της στήλης.
- ▶ **min**:Η ελάχιστη τιμή της στήλης.
- ▶ **25%**: Το 25% των τιμών της στήλης.
- ▶ **50%**: Ταυτίζεται με τον μέσο όρο.
- ▶ **75%**: Το 75% των τιμών της στήλης.
- ▶ **max**:Η μέγιστη τιμή των τιμών.

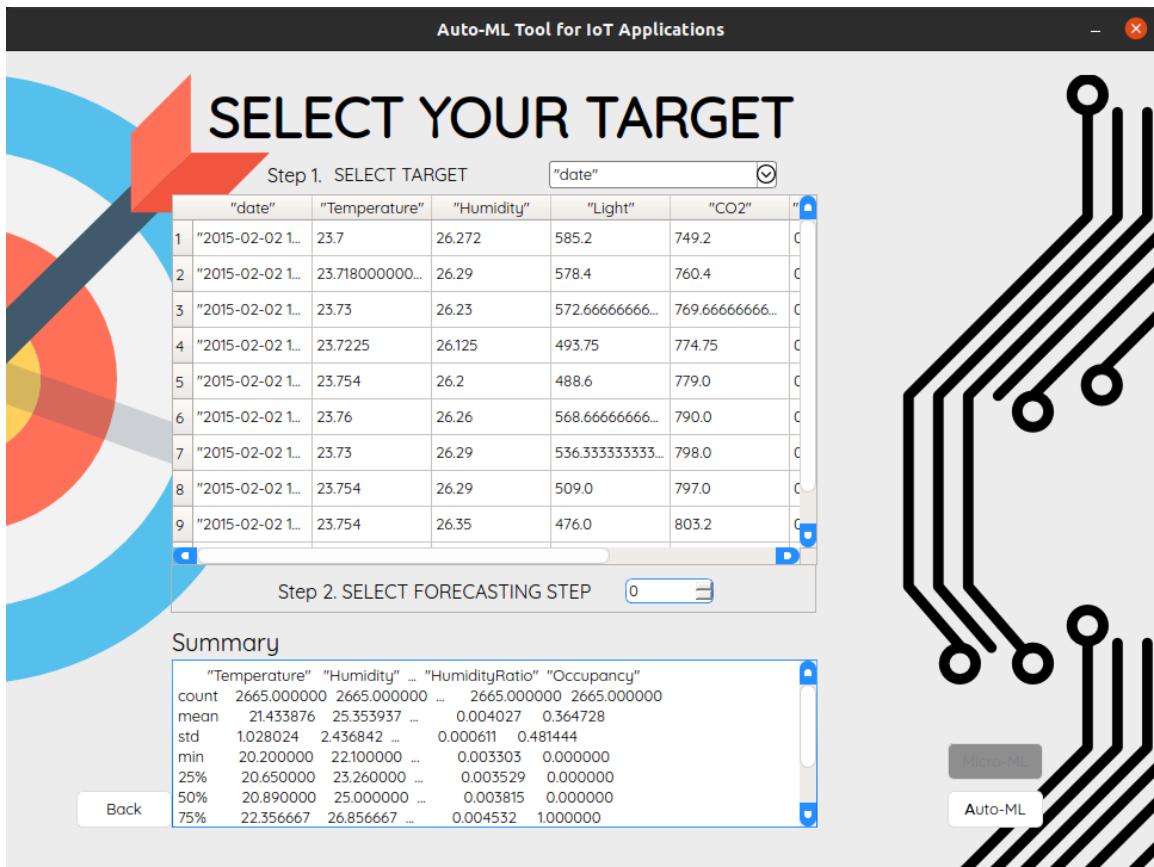
Ο χρήστης μπορεί να συνεχίσει την διαδικασία, εφόσον έχει επιλέξει την Target στήλη, μέσω του κομπιού Next, που θα τον οδηγήσει στην οθόνη “Parameter Tuning” Διαφορετικά έχει την δυνατότητα να επιλέξει το κομπί Load Existing Model και να μεταβέι στην οθόνη **Model**.



Σχήμα 3.11: Model Οθόνη

Εδώ μπορεί να επιλέξει ανάμεσα σε διάφορα αποθηκευμένα μοντέλα, να δει πληροφορίες για αυτά καθώς και να πραγματοποιήσει προβλέψεις, πάνω στο σύνολο δεδομένων που έχει εισάγει στην εφαρμογή, μέσω του κουμπιού Predict Target.

### 3..4.2 Τύπος Προβλήματος: Χρονοσειρά

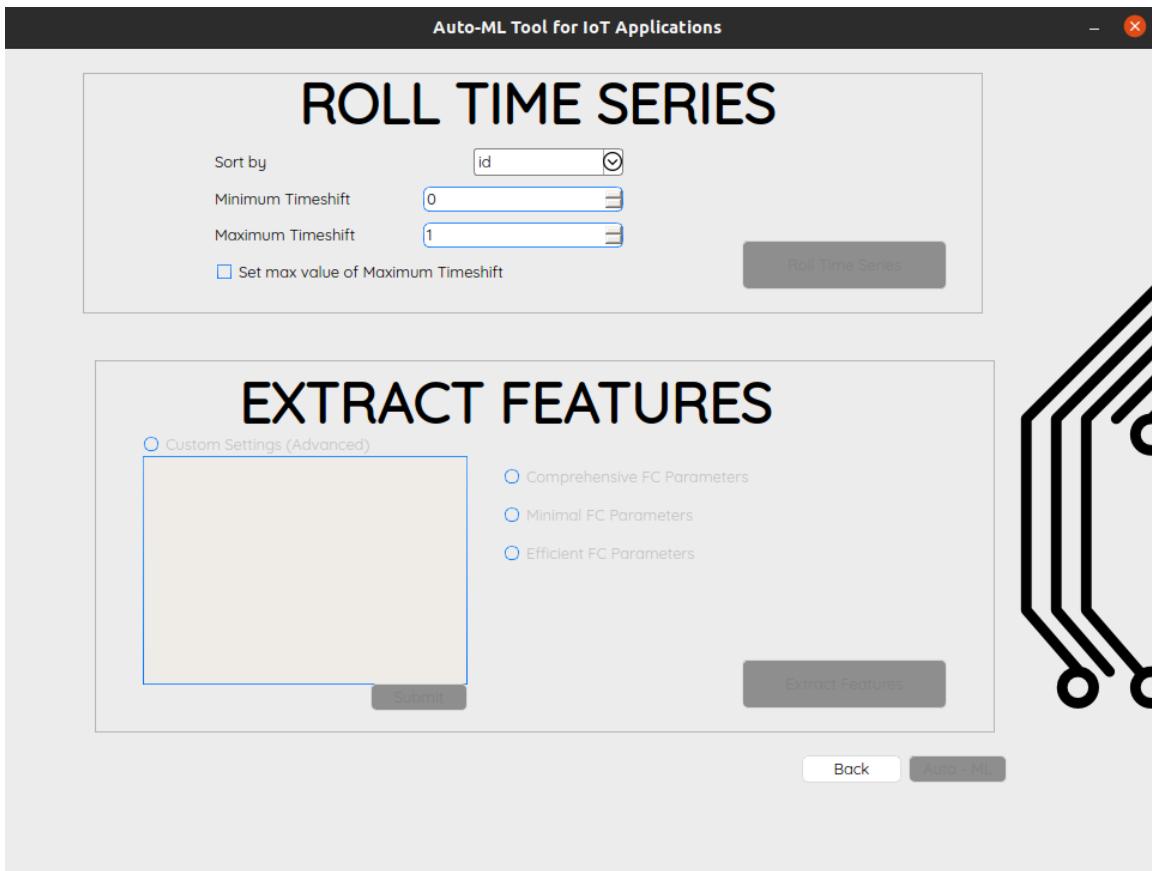


Σχήμα 3.12: Select Target (timeseries) Οθόνη

Στην παραγραφο αυτή περιγράφεται η λειτουργία στην περίπτωση που ο τύπος του προβλήματος έχει οριστεί ως Χρονοσειρά (Timeseries). Η οθόνη έχει την ίδια μορφή και λειτουργία και σε αυτήν την περίπτωση, με την διαφορά ότι ο χρήστης μπορεί να ορίσει, επιπλέον, ένα βήμα πρόβλεψης( Forecasting Step) για την χρονοσειρά, μέσω της επιλογής Select Forecasting Step. Μόλις ο χρήστης ορίσει αυτήν την παράμετρο, θα μεταφερθεί στην οθόνη “Roll Time Series - Extract Features”.

### 3..5 “Roll Time Series - Extract Features Οθόνη”

Αυτή η οθόνη εμφανίζεται στον χρήστη μόνο στην περίπτωση που ο τύπος του προβλήματος είναι “Χρονοσειρά”. Η οθόνη αποτελείται από δύο μέρη. Το πρώτο μέρος είναι το “Roll Time Series”.



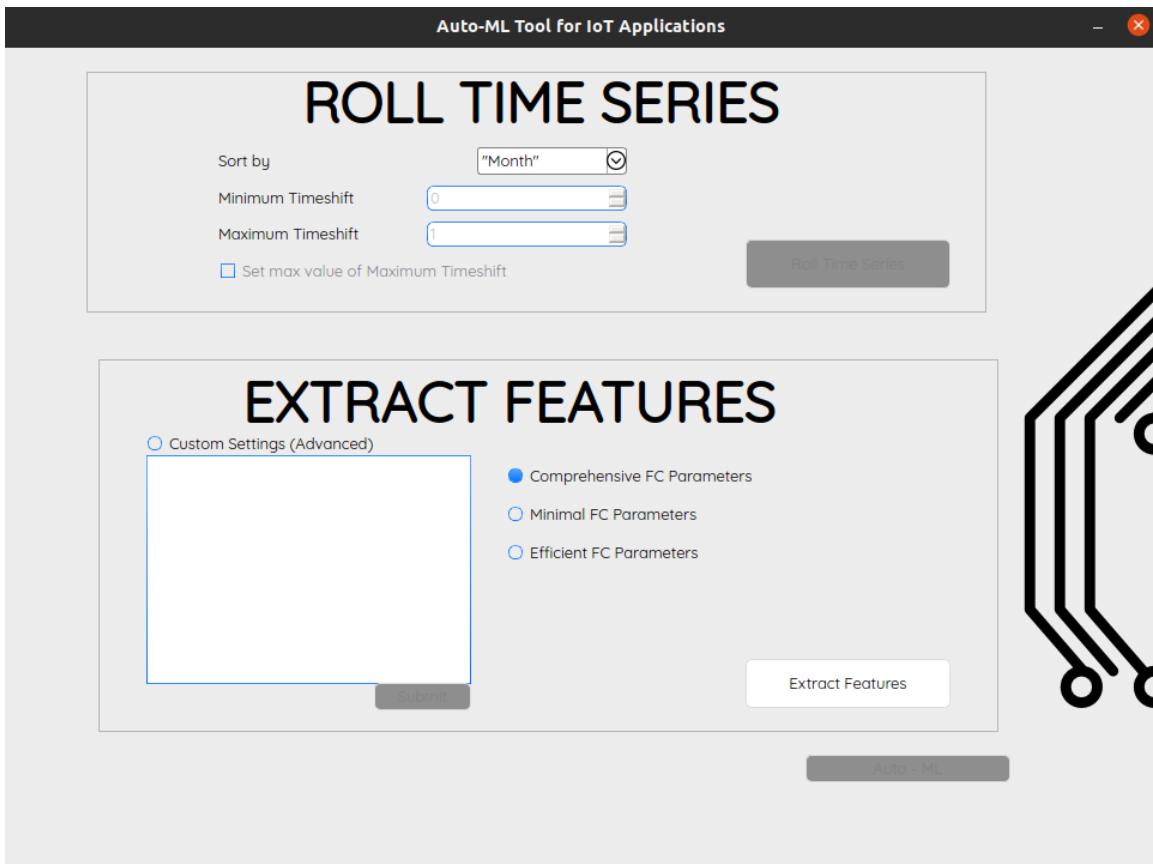
Σχήμα 3.13: Rolling

Στο σημείο αυτό ο χρήστης έχει την δυνατότητα ορισμού κάποιων παραμέτρων. Συγκεκριμένα:

- ▶ **Sort by**
- ▶ **Minimum timeshift**
- ▶ **Maximum timeshift.**
- ▶ Στην περίπτωση που θέλει να επιλέξει την μέγιστη τιμή της παραμέτρου **Maximum timeshift** μπορεί να επιλέξει το κουτί Set Maximum value of Maximum timeshift

Όταν όλες οι παράμετροι έχουν οριστεί, ο χρήστης μπορεί να πατήσει το κουμπί “Roll Time Series”, ώστε να ξεκινήσει την διαδικασία του rolling.

Το δεύτερο μέρος είναι το “Extract Features” και είναι διαθέσιμο αμέσως μόλις ολοκληρωθεί η προηγούμενη διαδικασία του rolling.



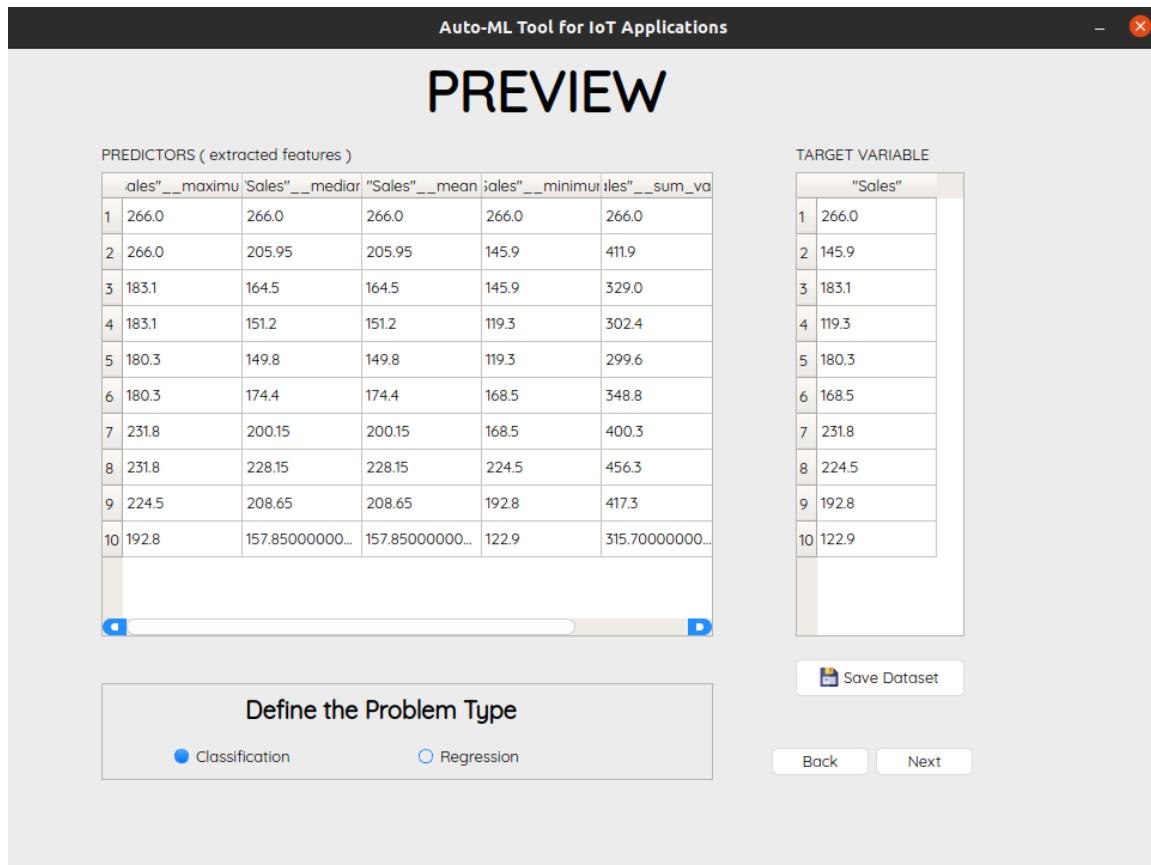
Σχήμα 3.14: Feature Extraction

Στο σημείο αυτό ο χρήστης μπορεί να παραμετροποιήσει την διαδικασία παραγωγής και εξαγωγής νέων χαρακτηριστικών. Τα σύνολα των παραμέτρων που μπορεί να δώσει στο σύστημα είναι ένα από τα παρακάτω:

- ▶ **Comprehensive FC Parameters**
- ▶ **Minimal FC Parameters**
- ▶ **Efficient FC Parameters**
- ▶ **Custom Settings:** Δυνατότητα ορισμού από τον χρήστη βάσει της προτίμησης του. Μετά τον ορισμό ο χρήστης θα πρέπει να πατήσει το submit κουμπί για να γίνουν οι αλλαγές. Στην περίπτωση που η είσοδος που έρισε δεν ακολουθεί το σωστό συντακτικό, θα εμφανιστεί το αντίστοιχο μήνυμα λάθους.

Όταν οριστούν και αυτές οι παράμετροι, το Auto - ML κουμπί γίνεται διαθέσιμο και με το πάτημα του, ο χρήστης μεταβαίνει στην οθόνη “Preview”.

### 3..6 “Preview” Οθόνη

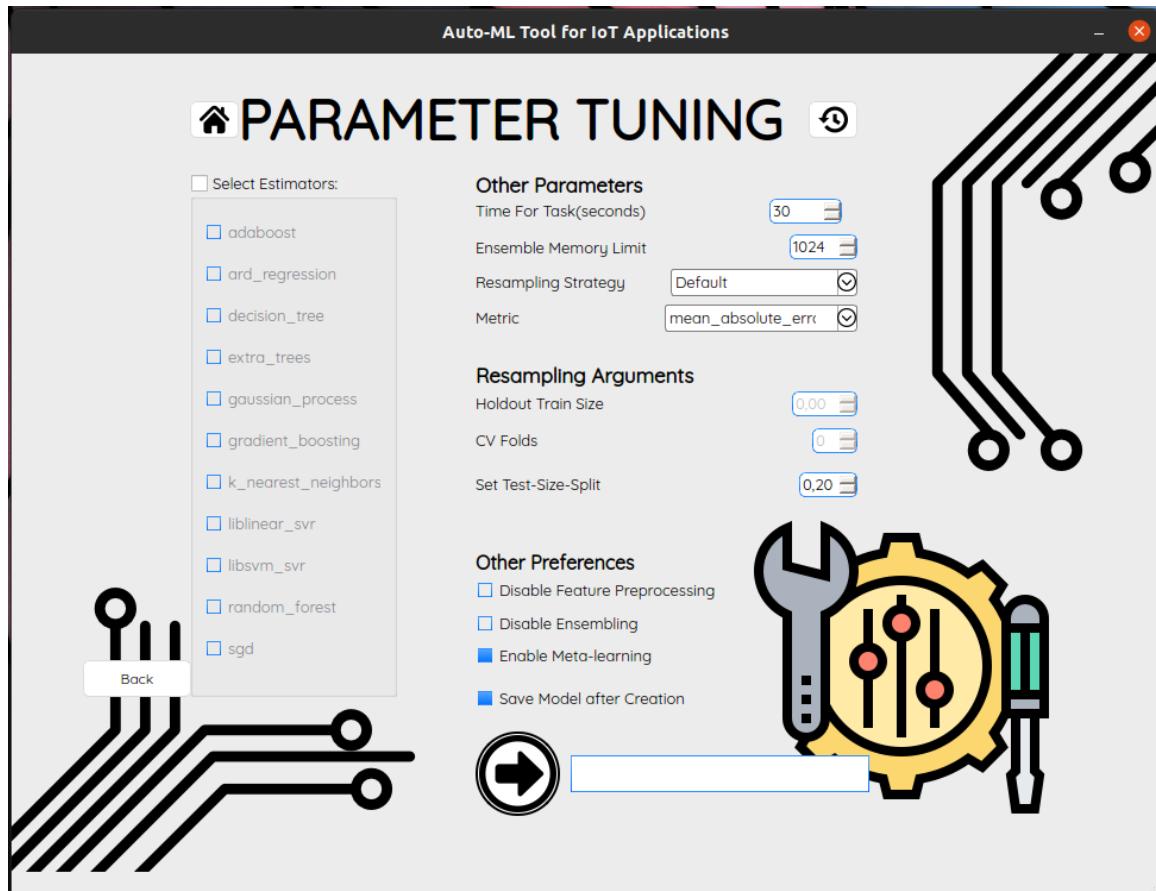


Σχήμα 3.15: Preview Οθόνη

Η συγκεκριμένη οθόνη εμφανίζεται στον χρήστη μόνο στην περίπτωση που ο τύπος προβλήματος είναι Time-Series. Η δομή της είναι παρόμοια με αυτήν της οθόνης 3.4. Αποτελείται από μία προεπισκόπηση σε πίνακα του dataset που έχει παραχθεί από την διαδικασία εξαγωγής δεδομένων και μία προεπισκόπηση της στήλης που αποτελεί το target χαρακτηριστικό. Οι στήλες του πίνακα αποτελούν τα χαρακτηριστικά του dataset, και οι γραμμές του αποτελούν τις εγγραφές. Ο χρήστης σε αυτό το σημείο δεν καλείται να επιλέξει target χαρακτηριστικό, καθώς αυτό έχει ήδη επιλεγεί στην αντίστοιχη οθόνη. Καλείται, όμως, να ορίσει τον τύπο του προβλήματος ξανά, καθώς το πρόβλημα χρονοσειρών έχει πλέον μετατραπεί σε πρόβλημα Classification ή Regression. Μετά τον ορισμό του προβλήματος ο χρήστης θα μεταβεί μέσω του Next κουμπιού στην οθόνη “Parameter Tuning” για τον ορισμό των παραμέτρων της διαδικασίας δημιουργίας μοντέλου.

### 3..7 “Parameter Tuning” Οθόνη

Οι οδόνες στις αποίες ορίζονται οι παράμετροι του συστήματος, που θα χρησιμοποιηθούν στην διαδικασία δημιουργίας ενός νέου μοντέλου, ακολουθούν παρόμοια λογική και έχουν την ίδια δομή και οργάνωση, ανεξάρτητα από τον τύπο του προβλήματος.



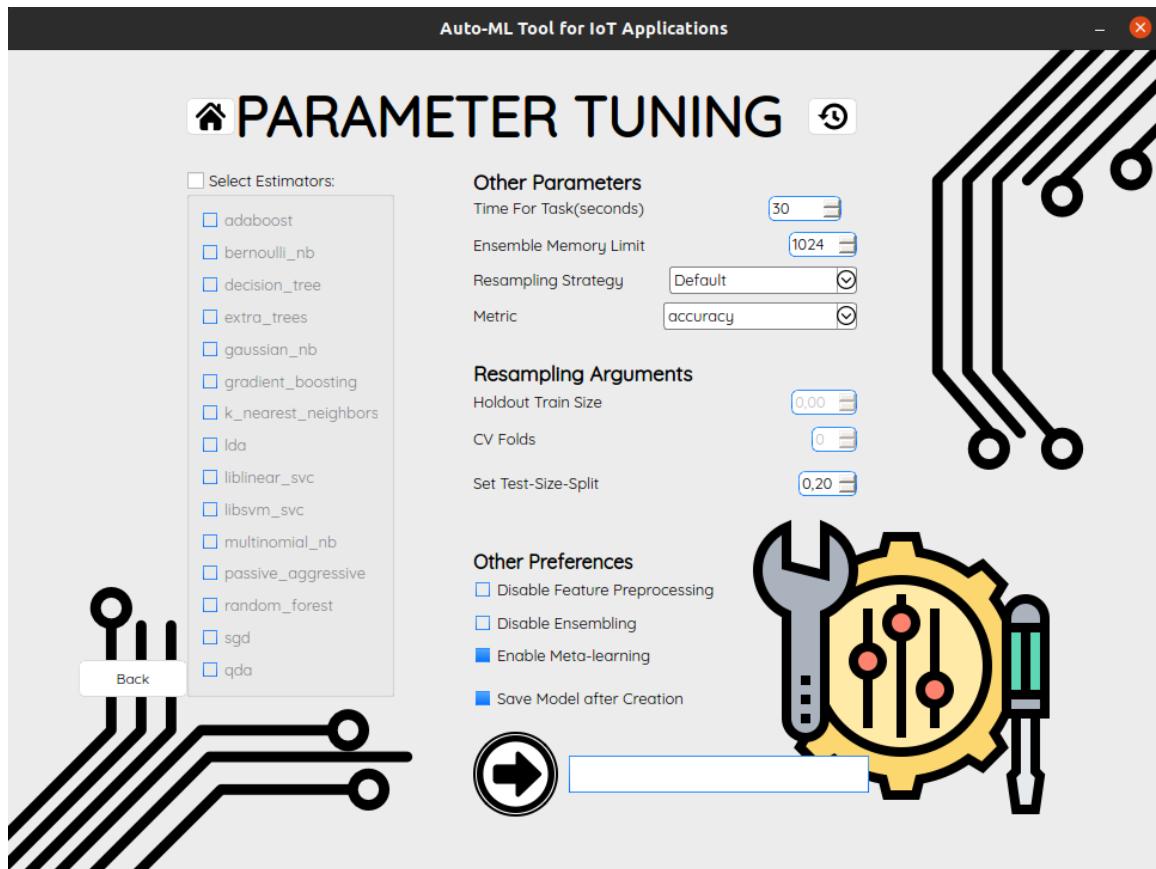
Σχήμα 3.16: Regression Οθόνη

Το σχήμα 3.16 απεικονίζει την οθόνη, στην οποία ο χρήστης έχει την δυνατότητα να παραμετροποιήσει την διαδικασία δημιουργίας μοντέλου, στην περίπτωση που το πρόβλημα μηχανικής μάθησης είναι ένα πρόβλημα **regression**. Στα αριστερά της οθόνης βρίσκονται οι διαθέσιμοι στατιστικοί αλγόριθμοι. Στην περίπτωση που η επιλογή “Select Estimators” δεν είναι κλικαρισμένη, όλοι οι αλγόριθμοι θα ελεγχθούν και θα χρησιμοποιηθούν για την διαδικασία δημιουργίας του τελικού μοντέλου ή του συνόλου μοντέλων (ensemble). Στην περίπτωση που ο χρήστης κλικάρει την επιλογή “Select Estimators”, τότε του

δίνεται η δυνατότητα να συμπεριλάβει συγκεκριμένους αλγόριθμους στην διαδικασία και να αποκλείσει άλλους. Οι διαθέσιμοι αλγόριθμοι για την δημιουργία ενός μοντέλου regression είναι οι εξής έντεκα:

- ▶ adaboost
- ▶ ard regression
- ▶ decision tree
- ▶ extra trees
- ▶ gaussian process
- ▶ gradien boosting
- ▶ k nearest neighboors
- ▶ liblinear svr
- ▶ libsvm svr
- ▶ random forest
- ▶ sgd

Το σχήμα 3.17 απεικονίζει την οθόνη, στην οποία ο χρήστης έχει την δυνατότητα να παραμετροποιήσει την διαδικασία δημιουργίας μοντέλου, στην περίπτωση που το πρόβλημα μηχανικής μάθησης είναι ένα πρόβλημα **classification**.



Σχήμα 3.17: Regression Οθόνη

Με την ίδια ακριβώς λογική, ο χρήστης μπορεί να επιλέξει να μην συμπεριλάβει αλγόριθμους από την διαδικασία. Η διαφορά μεταξύ της συγκεκριμένης οθόνης, σε σχέση με την οθόνη παλινδρόμησης, είναι οι διαθέσιμοι αλγόριθμοι. Σε αυτήν την περίπτωση, για την δημιουργία ενός μοντέλου classification διατίθενται οι εξής δεκατέσσερις αλγόριθμοι:

- ▶ adaboost
- ▶ bernouli\_nb
- ▶ decision\_tree
- ▶ extra\_trees
- ▶ gaussian\_nb
- ▶ gradient\_boosting

- ▶ k nearest neighbors
- ▶ lda
- ▶ liblinear svc
- ▶ multinomial nb
- ▶ passive aggressive
- ▶ random forest
- ▶ sgd
- ▶ qda

Στο σημείο Other Parameters και Resampling Arguments ο χρήστης έχει την δυνατότητα να παραμετροποιήσει όπως επιθυμεί την διαδικασία δημιουργίας μοντέλου ορίζοντας τον χρόνο εκτέλεσης, την μετρική, τις τεχνικές διαδικασίας επαλήθευσης κ.α. Στην περιοχή “Other Preferences” ο χρήστης δύναται να κλικάρει και να ενεργοποιήσει ή να απενεργοποιήσει άλλες λειτουργίες της διαδικασίας παραγωγής μοντέλων όπως την προεπεξεργασία χαρακτηριστικών την δημιουργία συνδυαστικών μοντέλων ή την λειτουργία μετα-μάθησης.

Τέλος, ο χρήστης έχει την δυνατότητα να αποθηκεύσει το μοντέλο που θα παραχθεί μέσω της επιλογής “Save Model after Creation”. Μέσω του κουμπιού έναρξης, ξεκινά η διαδικασία δημιουργίας του μοντέλου, αφότου ο χρήστης έχει θέσει τις προτιμήσεις του και έχει ρυθμίσει τις παραμέτρους του συστήματος.

# Κεφάλαιο 4

## Υλοποίηση

Στο κεφάλαιο αυτό περιγράφονται οι τεχνικές και οι διαδικασίες υλοποίησης του εργαλείου αυτοματοποιημένης μηχανικής μάθησης που αναπτύχθηκε. Επιπλέον, περιγράφεται η αρχιτεκτονική του συστήματος και η δομή του κώδικα. Στο επόμενο κεφάλαιο περιγράφεται, ελέγχεται και αξιολογείται η εφαρμογή όσον αφορά στις λειτουργίες που υλοποιεί.

### 1. Τεχνική Περιγραφή και Απαιτήσεις Συστήματος

Στην συγκεκριμένη ενότητα θα παρουσιαστούν αναλυτικά τα εργαλεία που χρησιμοποιούθηκαν για την υλοποίηση της εφαρμογής και των λειτουργιών της. Επιπλέον, θα περιγραφούν οι απαιτήσεις συστήματος και τα βήματα εγκατάστασης της εφαρμογής σε ένα νέο υπολογιστικό σύστημα. Η τεχνική περιγραφή του συστήματος στο οποίο υλοποιήθηκε η εφαρμογή βρίσκεται στον πίνακα:

Τεχνική Περιγραφή	
Λειτουργικό Σύστημα	Linux Ubuntu 20.04.2 LTS (Focal Fossa)
Επεξεργαστής	AMD® Ryzen 5 3600 6-core processor × 12
Μνήμη RAM	16GB DDR4 3200MHz
Σκληρός Δίσκος	Samsung 970 Evo Plus SSD 500GB M.2 NVMe

Οι απαιτήσεις συστήματος για την εφαρμογή είναι οι εξής

Απαιτήσεις Συστήματος	
Λειτουργικό Σύστημα	Linux (π.χ. Ubuntu)
Έκδοση Python	≥ 3.6
Compiler	GCC - C++ Compiler με υποστήριξη C++ 11
Έκδοση SWIG	3.0.* (εκδόσεις ≥ 4.0.0 δεν υποστηρίζονται)

Η λίστα των απαιτούμενων πακέτων για την ορθή λειτουργία της εφαρ-

μογής είναι η εξής:

```
_libgcc_mutex=0.1=main
auto-sklearn=0.12.6=pypi_0
bokeh=2.3.1=py38h578d9bd_0
brotlipy=0.7.0=py38h8df0ef7_1001
ca-certificates=2020.12.5=ha878542_0
certifi=2020.12.5=py38h578d9bd_1
cffi=1.14.5=py38h261ae71_0
chardet=4.0.0=py38h578d9bd_1
click=7.1.2=pyh9f0ad1d_0
cloudpickle=1.6.0=py_0
configspace=0.4.18=pypi_0
cryptography=3.4.7=py38ha5dfef3_0
cython=0.29.23=pypi_0
cytoolz=0.11.0=py38h25fe258_1
dask=2021.4.1=pyhd8ed1ab_0
dask-core=2021.4.1=pyhd8ed1ab_0
dbus=1.13.18=hb2f20db_0
distributed=2021.4.1=py38h578d9bd_0
expat=2.2.10=he6710b0_2
fontconfig=2.13.0=h9420a91_0
freetype=2.10.4=h5ab3b9f_0
fsspec=2021.4.0=pyhd8ed1ab_0
glib=2.56.2=hd408876_0
gst-plugins-base=1.14.0=hb8d80ab_1
gstreamer=1.14.0=hb453b48_1
heapdict=1.0.1=py_0
icu=58.2=he6710b0_3
idna=2.10=pyh9f0ad1d_0
jinja2=2.11.3=pyh44b312d_0
joblib=1.0.1=pyhd8ed1ab_0
jpeg=9b=habf39ab_1
lazy-import=0.2.2=pypi_0
ld_impl_linux-64=2.33.1=h53a641e_7
liac-arff=2.5.0=pypi_0
libblas=3.9.0=8_openblas
libcblas=3.9.0=8_openblas
libffi=3.3=he6710b0_2
libgcc-ng=9.1.0=hdf63c60_0
libgfortran-ng=7.5.0=h14aa051_19
libgfortran4=7.5.0=h14aa051_19
```

```
liblapack=3.9.0=8_openblas
libopenblas=0.3.12=pthreads_hb3c22a3_1
libpng=1.6.37=hbc83047_0
libstdcxx-ng=9.1.0=hdf63c60_0
libtiff=4.1.0=h2733197_1
libuuid=1.0.3=h1bed415_2
libxcb=1.14=h7b6447c_0
libxml2=2.9.10=hb55368b_3
locket=0.2.0=py38_1
lz4-c=1.9.2=he1b5a44_3
markupsafe=1.1.1=py38h8df0ef7_2
micromlgen=1.1.23=pypi_0
msgpack-python=1.0.0=py38h82cb98a_2
ncurses=6.2=he6710b0_1
numpy=1.19.4=py38hf0fd68c_1
olefile=0.46=pyh9f0ad1d_1
openssl=1.1.1k=h27cf23_0
packaging=20.9=pyh44b312d_0
pandas=1.1.3=py38he6710b0_0
partd=1.2.0=pyhd8ed1ab_0
patsy=0.5.1=py_0
pcre=8.44=he6710b0_0
pillow=7.1.2=py38hb39fc2d_0
pip=21.0.1=py38h06a4308_0
psutil=5.7.2=py38h7b6447c_0
pycparser=2.20=pyh9f0ad1d_2
pynisher=0.6.4=pypi_0
pyopenssl=20.0.1=pyhd8ed1ab_0
pyparsing=2.4.7=pyh9f0ad1d_0
pyqt=5.9.2=py38h05f1152_4
pyrfr=0.8.2=pypi_0
pysocks=1.7.1=py38h578d9bd_3
python=3.8.8=hdb3f193_5
python-dateutil=2.8.1=py_0
python_abi=3.8=1_cp38
pytz=2021.1=pyhd8ed1ab_0
pyyaml=5.3.1=py38h7b6447c_1
qt=5.9.7=h5867ecd_1
readline=8.1=h27cf23_0
requests=2.25.1=pyhd3deb0d_0
scikit-learn=0.24.2=pypi_0
scipy=1.5.3=py38h828c644_0
```

```
setuptools=52.0.0=py38h06a4308_0
sip=4.19.24=py38he6710b0_0
six=1.15.0=pyh9f0ad1d_0
smac=0.13.1=pypi_0
sortedcontainers=2.3.0=pyhd8ed1ab_0
sqlite=3.35.4=hdfb4753_0
statsmodels=0.12.1=py38h0b5ebd8_1
tblib=1.7.0=pyhd8ed1ab_0
threadpoolctl=2.1.0=pyh5ca1d4c_0
tk=8.6.10=hbc83047_0
toolz=0.11.1=py_0
tornado=6.1=py38h25fe258_0
tqdm=4.60.0=pyhd8ed1ab_0
tsfresh=0.17.0=py_0
typing_extensions=3.7.4.3=py_0
urllib3=1.26.4=pyhd8ed1ab_0
wheel=0.36.2=pyhd3eb1b0_0
xz=5.2.5=h7b6447c_0
yaml=0.2.5=h516909a_0
zict=2.0.0=py_0
zlib=1.2.11=h7b6447c_3
zstd=1.4.5=h6597ccf_2
```

## 2. Οδηγοί Εγκατάστασης

### 2..1 Περιβάλλον Υλοποίησης

Η ανάπτυξη του εργαλείου πραγματοποιήθηκε σε υπολογιστικό σύστημα με λειτουργικό σύστημα Linux Ubuntu 20.04. Επιλέχθηκε η γλώσσα προγραμματισμού Python 3.8 [36] και το ολοκληρωμένο περιβάλλον υλοποίησης Visual Studio Code .

Για την εγκατάσταση και την διαχείριση των πακέτων που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής, έγινε χρήση του εικονικού περιβάλλοντος Miniconda. Το conda είναι ένα ανοιχτού κώδικα σύστημα διαχείρισης πακέτων και περιβαλλόντων. Είναι ανεπτυγμένο σε Python και υποστηρίζεται από τα λειτουργικά συστήματα Windows, Mac OS και Linux. Αναλαμάνει την διαχείριση, την εγκατάσταση και την ενημέρωση πακέτων καθώς και των εξαρτήσεων τους με την υποστήριξη του Anaconda®.

## 2..2 Οδηγός Για Προγραμματιστές

Στην συνέχεια παρατίθενται τα βήματα για την προετοιμασία του περιβάλλοντος υλοποίησης και την εγκατάσταση των απαραίτητων πακέτων από την αρχή. Επιπλέον περιγράφεται η διαδικασία δημιουργίας εκτελέσιμου αρχείου ώστε να εφαρμογή να μπορεί να εκτελεστεί σε οποιοδήποτε υπολογιστικό σύστημα άμεσα.

**Σημείωση:** Τα παρακάτω βήματα ακολουθήθηκαν κατά την διαδικασία υλοποίησης και ανάπτυξης της εφαρμογής. Λόγω της διαρκούς αναβάθμισης των πακέτων και της Python ενδέχεται κάποιες απαιτήσεις να αλλάζουν και το περιβάλλον να μην είναι λειτουργικό για την εκτέλεση της εφαρμογής. Προτείνεται η άμεση εγκατάσταση των πακέτων μέσω του ths.yml αρχείου όπως περιγράφεται στον οδηγό, στο σημείο 2..2!

**Βήμα 1: Εγκατάσταση miniconda / Δημιουργία και ενεργοποίηση εικονικού περιβάλλοντος**

Αρχικά πρέπει να εγκαταστήσουμε στο σύστημά μας το miniconda. Κατεβάζουμε το miniconda από τον σύνδεσμο [https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\\_64.sh](https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh). Στην συνέχεια ανοίγουμε το Terminal και εκτελούμε την εντολή

```
$ bash Miniconda3-latest-Linux-x86_64.sh
```

Κλείνουμε και ανοίγουμε ξανά το Terminal ώστε να λάβουν χώρα οι αλλαγές μας. Επιβεβαιώνουμε ότι το miniconda έχει εγκατασταθεί επιτυχώς μέσω της εντολής

```
$ conda --version
```

, όπου θα μας επιστρέψει την εγκατεστημένη έκδοση του miniconda. Εφόσον έχει πραγματοποιηθεί η εγκατάσταση, θα χρειαστεί να δημιουργήσουμε ένα νέο περιβάλλον όπου θα εγκαταστήσουμε τα απαραίτητα πακέτα για την εφαρμογή.

**Σε αυτό το σημείο μπορούμε να δημιουργήσουμε ένα περιβάλλον από ένα ήδη υπάρχον, μέσω ενός yml αρχείου. Η διαδικασία αυτή περιγράφεται παρακάτω.**

Δημιουργούμε ένα νέο κενό περιβάλλον μέσω της εντολής

```
$ conda create -n my_env
```

Δίνουμε το όνομα που επιθυμούμε στο περιβάλλον μας. Μόλις δημιουργήσουμε το νέο περιβάλλον μας θα πρέπει να το ενεργοποιήσουμε μέσω της εντολής

```
$ conda activate my_env
```

Μπορούμε να επιβεβαιώσουμε την ενεργοποίηση του περιβάλλοντός μας εαν η γραμμή στην οποία γράφουμε στο terminal ξεκινάει με το όνομα του, (my\_env), και όχι με (base).

## Βίνα 2: Εγκατάσταση Python

Εφόσον έχουμε ενεργοποιήσει το περιβάλλον που δημιουργήσαμε, μπορούμε να ξεκινήσουμε την εγκατάσταση των απαραίτητων πακέτων. Ξεκινάμε με την εγκατάσταση της γλώσσας Python. Η εφαρμογή υλοποιήθηκε στην έκδοση 3.8 της Python και για να την εγκαταστήσουμε θα χρησιμοποιήσουμε την εντολή

```
$ conda install python=3.8
```

Μέσω της εντολής αυτής θα εγκατασταθεί ένα σύνολο από βασικά πακέτα. Αποδεχόμαστε την εγκατάσταση τους και περιμένουμε την ολοκλήρωση της.

## Βίνα 3: Εγκατάσταση του auto-sklearn

Μόλις η εγκατάσταση ολοκληρωθεί προχωράμε στην εγκατάσταση του auto-sklearn. Αρχικά εγκαθιστάμε όλες τις εξαρτήσεις μέσω της εντολής

```
$ sudo apt install curl  
$ sudo apt-get install build-essential  
$ curl https://raw.githubusercontent.com/automl/auto-sklearn/master/  
    ↪ requirements.txt | xargs -n 1 -L 1 pip3 install
```

Μετά την επιτυχή εγκατάσταση των παραπάνω πακέτων, εκτελούμε την εντολή για την εγκατάσταση του autosklearn

```
$ pip3 install auto-sklearn
```

Μπορούμε να επιβεβαιώσουμε την εγκατάσταση του auto-sklearn με την εντολή

```
$ conda list
```

, η οποία θα επιστρέψει όλα τα εγκατεστημένα πακέτα του περιβάλλοντός μας. Το auto-sklearn θα πρέπει να βρίσκεται στην λίστα με τα εγκατεστημένα πακέτα.

## Βίνα 4: Εγκατάσταση του PyQt5

Στην συνέχεια εγκαθιστάμε το PyQt5 μέσω της εντολής

```
$ conda install -c anaconda pyqt
```

## **Βήμα 5: Εγκατάσταση της SQLite**

Έπειτα εγκαθιστάμε το SQLite πακέτο καθώς και τον SQLite Browser για το γραφικό περιβάλλον του μέσω της εντολής:

```
$ conda install -c conda-forge sqlite  
$ sudo apt-get install sqlitebrowser
```

## **Βήμα 6: Εγκατάσταση του tsfresh**

Στην συνέχεια εγκαθιστάμε το πακέτο tsfresh με την εντολή

```
$ conda install -c conda-forge tsfresh
```

## **Βήμα 7: Εγκατάσταση του micromlgen**

Τέλος εγκαθιστάμε το πακέτο micromlgen με την εντολή

```
$ pip install micromlgen
```

Μετά την επιτυχή εγκατάσταση των παραπάνω θα είμαστε σε θέση να εκτελέσουμε την εφαρμογή.

## **Βήμα 8: Εκτέλεση της εφαρμογής**

Για την εκτέλεση της εφαρμογής πρέπει να μεταβούμε στον φάκελο που βρίσκεται το αρχείο app.py και να την εκτελέσουμε.

```
$ cd <path-to-app-folder>  
$ python app.py
```

Η εφαρμογή μας θα εκτελεστεί και το παραμήκο της αρχικής σελίδας θα εμφανιστεί στην οθόνη μας.

## **Βήμα 9: Εξαγωγή του miniconda περιβάλλοντος σε αρχείο**

Μπορούμε να εξάγουμε το περιβάλλον που δημιουργήσαμε σε ένα αρχείο yml ώστε να μπορούμε να το χρησιμοποιήσουμε σε κάποιο άλλο υπολογιστικό σύστημα μέσω της εντολής

```
$ conda env export > environment.yml
```

Βεβαιωνόμαστε ότι έχουμε ενεργοποιήσει το περιβάλλον το οποίο θέλουμε να εξάγουμε πρωτού εκτελέσουμε την εντολή. Το αρχείο yml που θα προκύψει, θα αποθηκευτεί στον Home φάκελο του συστήματος και θα περιέχει όλα τα πακέτα του περιβάλλοντός μας. Μπορούμε να χρησιμοποιήσουμε το παραγόμενο αρχείο για την δημιουργία ενός πανομοιότυπου περιβάλλοντος σε κάποιο

άλλο υπολογιστικό σύστημα άμεσα χωρίς να εγκαταστήσουμε όλα τα πακέτα ένα ένα ξανά.

#### Βήμα 10: Δημιουργία εκτελέσιμου αρχείου

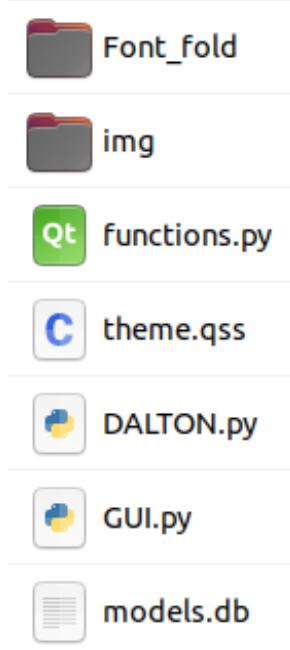
Αρχικά, ενώ βρισκόμαστε στο περιβάλλον που έχουμε δημιουργήσει, εγκαθιστάμε το πακέτο pyinstaller που θα μας βοηθήσει να κάνουμε την εφαρμογή μας εκτελέσιμη από οποιοδήποτε άλλο υπολογιστικό σύστημα χωρίς να χρειάζεται η εγκατάσταση όλων των παραπάνω πακέτων από την αρχή. Εγκαθιστάμε το pyinstaller με την εντολή

```
$ conda install -c conda-forge pyinstaller
```

Για να αποφύγουμε κάποια πιθανά μηνύματα σφάλματος εγκαθιστάμε μια παλαιότερη έκδοση της βιβλιοθήκης scipy μέσω της εντολής

```
$ conda install scipy=1.4.1
```

Στην συνέχεια βεβαιωνόμαστε ότι βρισκόμαστε στον φάκελο που περιέχει τα αρχεία python,qss, το αρχείο models.db και τους φακέλους που περιέχουν τις εικόνες και τις γραμματοσειρές που χρησιμοποιούνται σε αυτήν.



Σχήμα 4.1: Περιεχόμενα κύριου φακέλου

Στην συνέχεια εκτελούμε τις εντολές:

```
$ pip3 install opencv-python
$ sudo apt-get install libhdf5-dev
$ sudo apt-get install libhdf5-serial-dev
$ sudo apt-get install libatlas-base-dev
% Dimiourgia ektelestimou arxeiou
$ pyinstaller -w --add-data "models.db:." DALTON.py
```

Μετά την εκτέλεση της τελευταίας εντολής, θα δημιουργηθούν δύο υποφάκελοι, build και dist καθώς και ένα αρχείο spec. Στην συνέχεια, τροποποιώντας το αρχείο spec θα εισάγουμε στο εκτελέσμα αρχείο μας όλα τα απαραίτητα αρχεία όπως εικόνες και γραμματοσειρές. Επεξεργαζόματε το αρχείο DALTON.spec και προσθέτουμε στο πεδίο data τα εξής:

```
datas=[('models.db', '.'),  
       ('theme.qss', '.'),  
       ('./img/*.png', 'img'),  
       ('./Font_fold/*.ttf', 'Font_fold')  
     ],
```

Στην συνέχεια εκτελούμε την εντολή:

```
pyinstaller DALTON.spec
```

Το εκτελέσιμο αρχείο μας είναι το "DALTON" και βρίσκεται στον φάκελο dist/DALTON.

Για να μπορέσουμε να το εκτελέσουμε χωρίς προβλήματα θα πρέπει να προσθέσουμε στον φάκελο dist/DALTON κάποια νέα πακέτα που υπάρχουν στο conda περιβάλλον μας. Θα τα βρούμε στον φάκελο home/miniconda3/envs/<env\_name>/lib/python3.8/site-packages. Αντιγράφουμε όλα τα πακέτα και κάνουμε επικόλληση στον φάκελο dist/DALTON της εφαρμογής μας. Επιλέγουμε Replace για ίδια αρχεία και Merge για ίδιους φακέλους. Μόλις η διαδικασία ολοκληρωθεί μπορούμε να εκτελέσουμε την εφαρμογή μας μέσω terminal με την εντολή

```
$ ./DALTON
```

,εφόσον έχουμε πλοηγηθεί στον φάκελο dist/DALTON .

**Δημιουργία περιβάλλοντος miniconda από αρχείο yml. (Σταθερή έκδοση)**

Υπό την προϋπόθεση ότι έχουμε εγκαταστήσει επιτυχώς το miniconda σύμφωνα με το πρώτο βήμα του οδηγού, ακολουθούμε τα παρακάτω βήματα για την δημιουργία του περιβάλλοντός μας χρησιμοποιώντας ένα yml αρχείο. Ανοίγουμε το terminal, πλοηγούμαστε στον φάκελο, στον οποίο βρίσκεται το yml αρχείο μας και εκτελούμε την εντολή

```
$ conda env create -f environment.yml
```

Στην συνέχεια εγκαθιστούμε το autosklearn και το micromlgen με ρίρ, σε περίπτωση που δεν εγκατασταθούν με την παραπάνω εντολή. Χρησιμοποιούμε τις εντολές

```
$ pip3 install auto-sklearn  
$ pip3 install micromlgen
```

Το περιβάλλον μας έχει δημιουργηθεί με επιτυχία και έχει όλα τα απαραίτητα πακέτα για την εκτέλεση της εφαρμογής.

## Βήμα 2: Εκτέλεση εφαρμογής

Για την εκτέλεση της εφαρμογής πλοηγούμαστε στον κύριο φάκελο της εφαρμογής μέσω του terminal και εκτελούμε την εντολή

```
$ python DALTON.py
```

## 2..3 Οδηγός Για Χρήστες

Στον κύριο φάκελο της εφαρμογής πλοηγούμαστε στον φάκελο dist/DALTON. Στην συνέχεια κάνουμε δεξί κλικ και επιλέγουμε Open in terminal. Μόλις το τερματικό ανοίξει εκτελούμε την εντολή:

```
$ ./DALTON
```

Η εφαρμογή θα ξεκινήσει την εκτέλεση της.

## 3. Δομή κώδικα

Ο κώδικας είναι οργανωμένος σε 5 βασικά αρχεία τα οποία είναι τα εξής:

1. DALTON.py
2. functions.py
3. GUI.py
4. theme.qss
5. models.db

Το αρχείο GUI.py προκύπτει από την μετατροπή του gui.ui αρχείου, που περιλαμβάνει όλα τα χαρακτηριστικά του γραφικού περιβάλλοντος της εφαρμογής, σε python αρχείο. Έχουμε φροντίσει το συγκεκριμένο αρχείο να περιλαμβάνει όσο το δυνατόν λιγότερο κώδικα σχετικό με την λειτουργικότητα της εφαρμογής. Ο κώδικας που περιέχει το αρχείο σχετίζεται σχεδόν αποκλειστικά με το γραφικό περιβάλλον.

Το αρχείο DALTON.py περιέχει την main συνάρτηση που απαιτείται για την εκτέλεση του προγράμματος καθώς και την πλειοψηφία των βασικών συναρτήσεων που δίνουν λειτουργικότητα σε αυτό. Αποτελείται από την κλάση MainWindowUIClass που είναι υπο-κλάση της Ui\_MainWindow, όπου η τελευταία βρίσκεται στο αρχείο GUI.py. Ο λόγος που χρειάστηκε μια υποκλάση για την υλοποίηση των βασικών συναρτήσεων της εφαρμογής είναι ότι για κάθε νέο ενημερωμένο ui αρχείο, πρέπει να παραχθεί ένα νέο python αρχείο το οποίο αντικαθιστά όλον τον κώδικα του GUI.py με τον ενημερωμένο κώδικα. Επομένως εάν επιχειρούσαμε να γράψουμε τις βασικές συναρτήσεις στο αρχείο GUI.py, με κάθε ενημέρωση ο κώδικας θα χανόταν, κάτι που καθιστά απαραίτητο μία υπο-κλάση σε ένα αρχείο που θα ελέγχει και θα υλοποιεί την λειτουργικότητα του Γραφικού περιβάλλοντος της εφαρμογής.

Το αρχείο functions.py αποτελεί κατά κύριο λόγο ένα API καθώς περιέχει πολλή από την λειτουργικότητα των συναρτήσεων που καλούνται μέσω του

αρχείου DALTON.py. Ο λόγος που χρειάζεται ένα τέτοιο αρχείο είναι για την καλύτερη οργάνωση του κώδικα.

Το αρχείο theme.css περιλαμβάνει κώδικα που αφορά την εμφάνιση των στοιχείων του γραφικού περιβάλλοντός. Αποτελεί ένα stylesheet και κατά κάποιο τρόπο ένα θέμα εμφάνισης στην εφαρμογή.

Τέλος, το αρχείο models.db αποτελεί το αρχείο της SQLite βάσης δεδομένων μας και είναι απαραίτητο για την αποθήκευση των μοντέλων καθώς και για την ανάκτησή τους. Η βάση δεδομένων αποτελείται από έναν πίνακα, models, που αποτελείται από τέσσερις στήλες.

1. name: Πρόκειται για ένα αλφαριθμητικό πεδίο και αποτελεί το όνομα με το οποίο έχουμε αποθηκεύσει το μοντέλο μας στην βάση δεδομένων.
2. data: Πρόκειται για ένα αλφαριθμητικό πεδίο το οποίο περιέχει το μονοπάτι των αρχείων που περιέχουν τα κωδικοποιημένα μοντέλα στο σύστημα αρχείων του υπολογιστή.
3. timestamp: Περιέχει το χρονικό στιγμάτυπο που αντιστοιχεί στην στιγμή της δημιουργίας του μοντέλου.
4. learning type: Αποτελεί ένα αλφαριθμητικό πεδίο και αφορά στον τύπο του μοντέλου μηχανικής μάθησης. Ενδέχεται να είναι Classification ή Regression.

## 4. Βιβλιοθήκες της Python

### 4..1 csv

Ο τύπος αρχείων CSV (Comma Separated Values) είναι ο πιο σύνηθης τύπος εισαγωγής και εξαγωγής δεδομένων από βάσεις δεδομένων και υπολογιστικά φύλλα. Η βιβλιοθήκη csv δίνει την δυνατότητα εξαγωγής και εγγραφής δεδομένων σε τέτοια μορφή, όπως επίσης την εισαγωγή και ανάγνωση τους. Χρησιμοποιήθηκε για την διαχείριση των csv τύπου αρχείων που αποτελούν είσοδο στο σύστημα για την διαδικασία δημιουργίας μοντέλων.

### 4..2 pandas

Η βιβλιοθήκη pandas αποτελεί ένα ισχυρό εργαλείο για την διαχείριση και την ανάλυση real-world δεδομένων σε Python γλώσσα προγραμματισμού. Στην περίπτωση που η δεδομένα τα οποία θέλουμε να επεξεργαστούμε και να

διαχειριστούμε είναι της μορφής πίνακα, όπως για παράδειγμα csv ή υπολογιστικά φύλλα, τότε η συγκεκριμένη βιβλιοθήκη προσφέρει πολυ χρήσιμες κλάσεις και συναρτήσεις για αυτόν τον σκοπό.

### 4..3 sklearn

Η Scikit-learn [30] είναι μία από τις πιο χρήσιμες και διαδεδομένες βιβλιοθήκες μηχανικής μάθησης της Python. Η βιβλιοθήκη, αυτή, προσφέρει πολύτιμα και ισχυρά εργαλεία που αποσκοπούν στην δημιουργία στατιστικών μοντέλων, όπως classification, regression, clustering και άλλα. [4]

### 4..4 auto-sklearn

Η βιβλιοθήκη auto-sklearn είναι βασισμένη στην sklearn και παρέχει λειτουργίες για μηχανική μάθηση με επίβλεψη. Η καινοτομία της συγκεκριμένης βιβλιοθήκης είναι το γεγονός ότι δημιουργεί στατιστικά μοντέλα, **αυτοματοποιώντας** την αναζήτηση των κατάλληλων αλγορίθμων και την βελτιστοποίηση των υπερπαραμέτρων του. Σκοπός της βιβλιοθήκης είναι η διευκόλυνση της διαδικασίας και η απαλλαγή των αναλυτών από τις παραπάνω χρονοβόρες διαδικασίες. [22]

Η έκδοση που χρησιμοποιήθηκε είναι η 0.12.3. Στην συνέχεια παρουσιάζονται και αναλύονται οι κλάσεις και συναρτήσεις με τις παραμέτρους που χρησιμοποιήθηκαν από την βιβλιοθήκη.

#### AutoSklearnClassifier/AutoSklearnRegressor

- ▶ time\_left\_for\_this\_task Με την συγκεκριμένη παράμετρο ορίζεται η χρονική διάρκεια κατά την οποία θα γίνεται αναζήτηση βελτιστοποίηση και δημιουργία του τελικού ensemble.
- ▶ initial\_configurations\_via\_metalearning Μέσω της συγκεκριμένης παραμέτρου, δίνεται η δυνατότητα ενεργοποίησης και απενεργοποίησης της διαδικασίας του meta learning. Η διαδικασία αυτή περιγράφηκε γενικά στην ενότητα 2 που αφορά στο θεωρητικό υπόβαθρο της διπλωματικής εργασίας, όμως εδώ θα περιγραφεί αναλυτικά η τεχνική που χρησιμοποιείται συγκεκριμένα από την βιβλιοθήκη auto-sklearn. Στην ουσία, γίνεται συλλογή δεδομένων σχετικών με την απόδοση, καθώς και μετα-δεδομένων, για παράδειγμα χαρακτηριστικά που αφορούν το dataset, τα οποία μπορούν να μας δώσουν σημαντικές πληροφορίες ώστε να επιλέξουμε τον αλγόριθμο που θα πρέπει να χρησιμοποιηθεί σε ένα νέο dataset. Η τεχνική αυτή αποτελεί συμπλήρωμα της Bayesian Optimization και αντίστροφα,

υπό την έννοια ότι μπορεί να ανακαλύψει και να προτείνει κάποια διάταξη των διαδικασιών μηχανικής μάθησης που να είναι πιθανό να παράγει ένα αποδοτικό μοντέλο γρηγορότερα από αυτήν. Το αντίστροφο ισχύει καθώς, αντιθέτως με την Bayesian Optimization, η τεχνική του meta learning δεν μπορεί να βελτιστοποιήσει την απόδοση του μοντέλου που θα προτείνει. Συμπερασματικά, το auto sklearn χρησιμοποιεί ένα πλήθος από προτεινόμενες διατάξεις που παρήχθησαν μέσω του meta-learning ως είσοδο στην Bayesian Optimization. Αναλυτικά ο τρόπος με τον οποίο λειτουργεί η συγκεκριμένη τεχνική είναι ο εξής:

1. Εξάγονται και αξιολογούνται μεταδεδομένα από κάθε dataset ενός συνόλου από datasets που βρίσκεται στο OpenML repository.
  2. Μέσω της BO επιλέγεται μια διάταξη με την πιο ισχυρή εμπειρική απόδοση.
  3. Για ένα νέο dataset, υπολογίζονται τα μεταδεδομένα του.
  4. Τα υπάρχοντα datasets κατατάσσονται σύμφωνα με την απόσταση τους από το νέο dataset κατά αύξουσα σειρά.
  5. Επιλέγονται οι διατάξεις για τα 25 datasets τα οποία βρίσκονται πιο κοντά στο καινούριο στον χώρο των μετα-χαρακτηριστικών και ξεκινάει η BO για την βελτιστοποίηση των αποτελεσμάτων τους.
- ▶ ensemble\_size Μέσω της συγκεκριμένης μεταβλητής ορίζεται το πλήθος των μοντέλων από τα οποία θα αποτελείται το τελικό ensemble. Στην περίπτωση που είναι ίσο με τη μονάδα τότε η λειτουργία του ensembling απενεργοποιείται. Εκτός από την λειτουργία του meta-learning, το auto sklearn διαθέτει και την λειτουργία του ensembling. Μέσω της BO, ανακαλύπτουμε τις βέλτιστες ρυθμίσεις των υπερ-παραμέτρων με την καλύτερη απόδοση. Παρόλα αυτά παρατηρείται πως όλα τα μοντέλα που εκπαιδεύονται κατά την αναζήτηση του καλύτερου συνδυασμού υπερ-παραμέτρων μένουν ανεκμετάλλευτα ανεξάρτητα από την απόδοσή τους. Ενδέχεται πολλά από αυτά τα μοντέλα να είναι σχεδόν τόσο αποδοτικά, όσο και το βέλτιστο μοντέλο στο οποίο έχει καταλήξει η BO. Το auto sklearn χρησιμοποιεί αυτά τα μοντέλα για να δημιουργήσει ένα ensemble, με στόχο την αποφυγή του overfitting και την αύξηση της ευρωστίας της διαδικασίας. Τα μοντέλα που απαρτίζουν τελικό ensemble διαθέτουν διαφορετικά βάρον που υπολογίζονται μέσω της τεχνικής ensemble selection, καθώς είναι γρηγορότερη από τις τεχνικές stacking και numerical optimization
  - ▶ include\_estimators Μέσω της παραμέτρου αυτής ορίζονται ποιοι αλγόριθμοι μηχανικής μάθησης θα συμπεριληφθούν στην διαδικασία δημιουργίας του τελικού μοντέλου.

- `include_preprocessors` Εαν η συγκεκριμένη παράμετρος ορίστεί ως `none`, τότε η λειτουργία του αυτοματοποιημένου `preprocessing` απενεργοποιείται. Διαφορετικά, το `autosklearn` αναλαμβάνει να πραγματοποιήσει κάποιες διαδικασίες προεπεξεργασίας των δεδομένων εισόδου. Η διαδικασία του `preprocessing` διακρίνεται σε :

1. **Data preprocessing:** Περιλαμβάνει τις τεχνικές προεπεξεργασίας δεδομένων:

- Balancing
- Imputation
- One-Hot-Encoding
- Rescaling
- Variance Threshold

Η συγκεκριμένη διαδικασία του `data preprocessing` δεν είναι δυνατόν να απενεργοποιηθεί, επομένως με την απενεργοποίηση της παραμέτρου `include preprocessors` αναφερόμαστε αποκλειστικά στην απενεργοποίηση της διαδικασίας `Feature preprocessing`.

2. **Feature preprocessing:** Αποτελείται από τεχνικές που χρησιμοποιούνται για παράδειγμα για την αλλαγή του χώρου των χαρακτηριστικών των δεδομένων (π.χ. η τεχνική `PCA`) ή τεχνικές επιλογής χαρακτηριστικών.

- `resampling_strategy` Μέσω της συγκεκριμένης παραμέτρου είναι δυνατόν να ορίστεί η μέθοδος `resampling` που μπορεί να είναι `cross validation` ή `holdout`.
- `resampling_strategy_arguments` Ανάλογα με την μέθοδο που επιλέγεται υπάρχει η δυνατότητα ορισμού κάποιων επιπλέον παραμέτρων για την λειτουργία της μεθόδου.
- `metric`: Οι μετρικές που χρησιμοποιούνται για την αξιολόγηση κατά την δημιουργία του τελικού μοντέλου. Συγκεκριμένα έγινε χρήση των:

#### 4..5 tsfresh

Πρόκειται για μια βιβλιοθήκη της Python η οποία επεξεργάζεται και εξαγεί χρήσιμα χαρακτηριστικά από ένα πλήθος χρονοσειρών. Μέσω συναρτήσεων και κλάσεων της συγκεκριμένης βιβλιοθήκης υλοποιείται η διαδικασία `rolling` των χρονοσειρών καθώς και η διαδικασία εξαγωγής και επιλογής των πιο χρήσιμων χαρακτηριστικών από αυτές. Επιπλέον μέσω παραμέτρων ορίζονται οι προτιμήσεις που αφορούν την εξαγωγή χαρακτηριστικών. [3]

## 4..6 micromlgen

Η βιβλιοθήκη *micromlgen* είναι μία προσπάθεια υλοποίησης αλγορίθμων μηχανικής μάθησης σε μορφή που υποστηρίζουν μικρο-ελεγκτές. Η βιβλιοθήκη υποστηρίζει την μετατροπή διάφορων αλγορίθμων μηχανικής μάθησης στην κατάλληλη μορφή για εφαρμογή σε μικρο-ελεγκτές και έγινε η χρήση της για αυτόν ακριβώς τον σκοπό. Σε επόμενο κεφάλαιο περιγράφεται η δημιουργία, η εξαγωγή και χρήση μοντέλων μηχανικής μάθησης σε μικρο-ελεγκτές. [20]

## 4..7 PyQt5

Η βιβλιοθήκη αυτή υποστηρίζεται από την Python 3. Εμπεριέχει και παρέχει πάνω από 620 κλάσεις. Επιλέχθηκε στην παρούσα εργασία λόγω των δυνατοτήτων που παρέχει για την δημιουργία γραφικού περιβάλλοντος για εφαρμογές αναπτυγμένες σε Python . Μέσω των λειτουργίων της PyQt αναπτύχθηκε το γραφικό περιβάλλον του εργαλείου, το οποίο αποτελεί βασικό χαρακτηριστικό της υλοποίησης του. Η βιβλιοθήκη επιπλέον υποστηρίζει το εργαλείο PyQt Designer όπου αποτελεί ένα γραφικό περιβάλλον με δυνατότητες drag and drop για την διευκόλυνση της διαδικασίας δημιουργίας γραφικού περιβάλλοντος χρήστη.

## 4..8 pickle

Η βιβλιοθήκη αυτή είναι πολύ χρήσιμη, καθώς προσφέρει συναρτήσεις που επιτρέπουν την κωδικοποίηση (serialization) και την αποκωδικοποίηση αντικειμένων της Python. Πιο συγκεκριμένα η ιεραρχία ενός αντικειμένου μετατρέπεται σε μια ροή από bytes. Με αυτόν τον τρόπο, ένα αντικείμενο μπορεί να αποθηκευτεί σε τύπο ροής bytes, μέσω της διαδικασίας που είναι γνωστή ως pickling και να ανακτηθεί με την αντίστροφη διαδικασία (unpickling).

- ▶ **dump():** Μέσω της κλήσης της συγκεκριμένης συνάρτησης, το Python αντικείμενο που θέτουμε ως παράμετρο, γίνεται serialized και εξάγεται σε αρχείο μορφής pickle.
- ▶ **load():** Μέσω της συνάρτησης αυτής γίνεται η αντίστροφη διαδικασία με αποτέλεσμα να ανακτήσουμε το αρχικό αντικείμενο Python από ένα συγκεκριμένο αρχείο μορφής pickle.

## 4..9 sqlite3

Η συγκεκριμένη βιβλιοθήκη προσφέρει δυνατότητες διαχείρισης και δημιουργίας βάσεων δεδομένων που δεν απαιτούν ξεχωριστό server αλλά είναι

βασισμένες και στημένες στον δίσκο του υπολογιστή. Η διαχείριση των δεδομένων της βάσης που δημιουργείται μέσω της sqlite γίνεται με εντολές της γλώσσας προγραμματισμού SQL.

# Κεφάλαιο 5

## Έλεγχος και Αξιολόγηση

Στο κεφάλαιο αυτό θα περιγραφεί η διαδικασία που ακολουθήθηκε για τον έλεγχο και την αξιολόγηση της εφαρμογής.

### 1. Γενική μεθοδολογία ελέγχου

Ο έλεγχος του συστήματος αυτού πραγματοποιήθηκε με τη χρήση σεναρίων λειτουργίας. Στα σενάρια αυτό θεωρούμε ότι ένας τελικός χρήστης επιθυμεί να περιηγηθεί στην εφαρμογή, να επιλέξει αρχεία και να αξιοποιήσει τις λειτουργίες του εργαλείου, εξασφαλίζοντας την ορθή και ομαλή ροή τους. Παρακάτω περιγράφονται βηματικά οι διαδικασίες δημιουργίας μοντέλων για διαφορετικούς τύπους προβλημάτων. Οι οδηγοί αυτοί αποτελούν χρήσιμο εργαλείο για τους νέους χρήστες της εφαρμογής και ταυτόχρονα περιγράφουν την διαδικασία που ακολουθήθηκε για τον έλεγχο της λειτουργίας της.

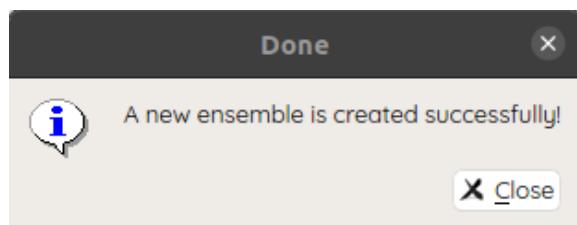
Ο χρήστης δεν πρέπει να συναντήσει προβλήματα και συμπεριφορές στην εφαρμογή που δεν προβλέπονταν κατά την υλοποίησή της. Στην περίπτωση που εμφανιστεί κάποιο σφάλμα, θα πρέπει η ροή του προγράμματος να συνεχίζεται, δίνοντας στον χρήστη οδηγίες για την αντιμετώπισή τους και πληροφορίες για την αιτία πρόκλησής τους. Ο απότομος και απρόβλεπτος τερματισμός της εφαρμογής λόγω οποιουδήποτε σφάλματος πρέπει να αποφευχθεί και αυτό επιτυγχάνεται μέσω αμυντικού προγραμματισμού. Κάθε σφάλμα στον κώδικα έχει αναγνωριστεί και έχει γίνει σωστή διαχείριση, έτσι ώστε να εξασφαλίζεται με κάθε τρόπο η συνέχεια της σωστής ροής εκτέλεσης του προγράμματος. Ιδιαίτερη σημασία δίνεται σε ακραίες περιπτώσεις όπου ο κώδικας ενδέχεται να περιέχει αδυναμίες και να εμφανίζει σφάλματα, για παράδειγμα σε κενές εισόδους δεδομένων, σε λάθος επιλογές τύπου αρχείου ή λάθος επιλογές παραμέτρων. Για παράδειγμα, μια τέτοια συμπεριφορά παρατηρείται στην περίπτωση που ο χρήστης ορίσει το πρόβλημα μηχανικής μάθησης ως

classification, αλλά επιλέξει ένα dataset που ανήκει στην κατηγορία regression. Αμέσως ένα σφάλμα θα προκύψει στον κώδικα το οποίο πρέπει να διαχειριστούμε ώστε να εξασφαλίσουμε την συνέχεια της ροής του προγράμματος.

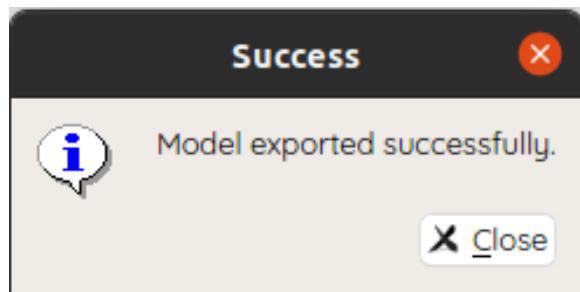
Επιπλέον, Είναι απαραίτητο να βεβαιωθούμε ότι ο χρήστης δεν καταλήγει σε αδιέξοδα ή σε ατέρμονες βρόχους κατά την πλοϊγησή του στην εφαρμογή. Για να επιτευχθεί αυτό πρέπει να διασφαλίσουμε ότι τα κουμπιά περιήγησης (Next, Back, Home) οδηγούν τον χρήστη στις οθόνες που επιθυμούμε σε όλα τα διαφορετικά σενάρια πλοϊγησης, καθώς επίσης και ότι ο κώδικας λειτουργεί ορθά σε κάθε διαφορετική περίπτωση.

Για να εξασφαλιστεί η σωστή ροή της εφαρμογής είναι απαραίτητο να περιηγηθούμε σε αυτήν χρησιμοποιώντας όλους τους πιθανούς συνδυασμούς περιήγησης μέσω των επιλογών Next, Back, Home. Πρέπει να διασφαλιστεί η εκκαθάριση τυχόν μεταβλητών καθώς και η εκ νέου αρχικοποίηση τους ιδιαίτερα όταν πλοηγούμαστε προς τα πίσω, έτσι ώστε κάθε διαδικασία που υλοποιείται από τις οθόνες να μπορεί να ξεκινήσει από την αρχή και να ολοκληρωθεί επιτυχώς, αποφεύγοντας την χρήση δεδομένων από προηγούμενες εκτελέσεις και την αρχικοποίηση διπλότυπων μεταβλητών. Για παράδειγμα, μια τέτοια συμπεριφορά παρατηρείται στην περίπτωση που δεν εκκαθαρίσουμε τα δεδομένα των λιστών όταν περιηγηθούμε προς τα πίσω και στην συνέχεια προς τα μπροστά. Το αποτέλεσμα είναι να γεμίσουμε ξανά τις λίστες με αντικείμενα, χωρίς να έχουμε εκκαθαρίσει τα προηγούμενα. Τελικά, καταλήγουμε σε λίστες που αποτελούνται από διπλότυπες ή περιττές επιλογές.

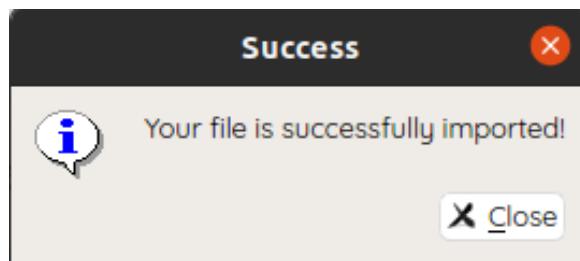
Ο χρήστης κατά την διάρκεια χρήσης της εφαρμογής ενημερώνεται για σωστές ή εσφαλμένες ενέργειες μέσω μηνυμάτων που εμφανίζονται στην οθόνη του. Τα μηνύματα που ενδέχεται να εμφανιστούν στον χρήστη είναι τα παρακάτω.



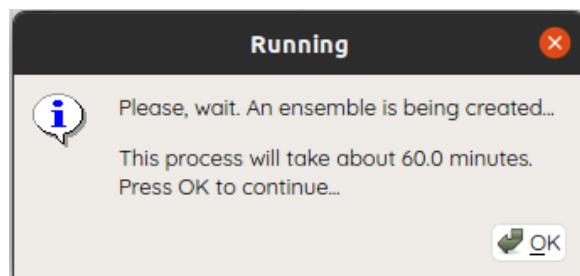
Σχήμα 5.1: Εμφανίζεται μετά την επιτυχή δημιουργία μοντέλου



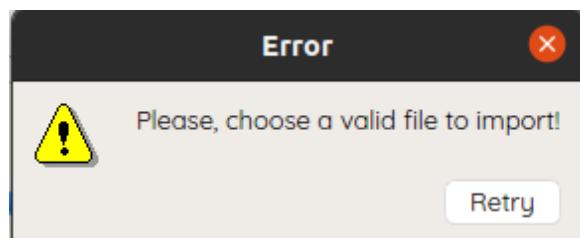
Σχήμα 5.2: Εμφανίζεται με την επιτυχή εξαγωγή ενός MicroML μοντέλου



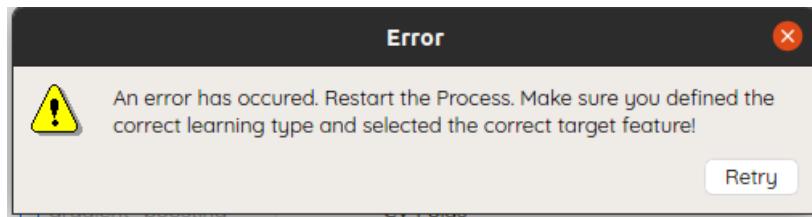
Σχήμα 5.3: Εμφανίζεται με την επιτυχή εισαγωγή ενός αρχείου δεδομένων στην εφαρμογή



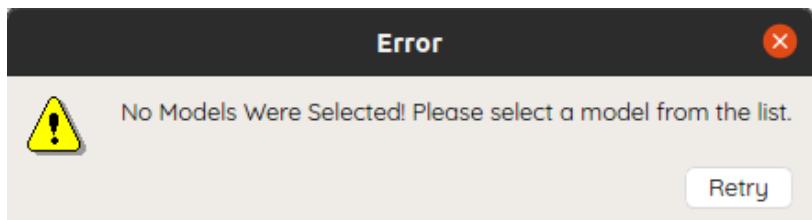
Σχήμα 5.4: Εμφανίζεται κατά την εκκίνηση της μοντέλοποίησης. Ενημερώνει τον χρήστη για τον χρόνο που θα διαρκέσει.



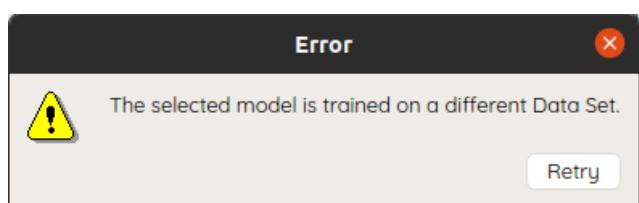
Σχήμα 5.5: Εμφανίζεται με την εισαγωγή ενός μη έγκυρου αρχείου δεδομένων στην εφαρμογή



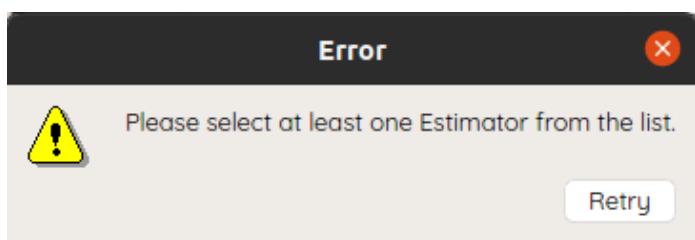
Σχήμα 5.6: Εμφανίζεται κατά την εκκίνηση της διαδικασίας μοντελοποίησης όταν κάποιο λάθος ανιχνευθεί



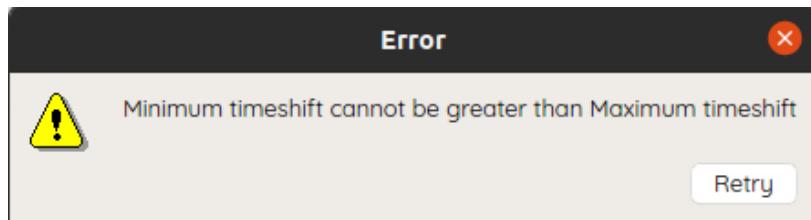
Σχήμα 5.7: Εμφανίζεται όταν δεν επιλεχθεί κάποιο μοντέλο για να φορτωθεί στην εφαρμογή



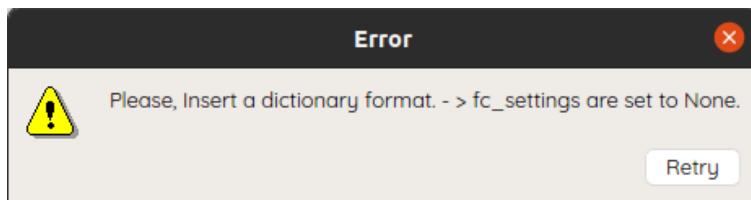
Σχήμα 5.8: Εμφανίζεται όταν το μοντέλο με το οποίο επιχειρούμε να πραγματοποιήσουμε προβλέψεις έχει εκπαιδευτεί σε διαφορετικό σύνολο δεδομένων από αυτό που έχουμε εισάγει στην εφαρμογή



Σχήμα 5.9: Εμφανίζεται στην περίπτωση που δεν έχει επιλεχθεί κανένας αλγόριθμος από την λίστα στην οθόνη παραμετροποίησης



Σχήμα 5.10: Εμφανίζεται όταν το Minimum timeshift είναι αριθμός μεγαλύτερος του Maximum timeshift



Σχήμα 5.11: Εμφανίζεται όταν η μορφή του Custom Parameters πεδίου δεν είναι τύπου Python Dictionary

Στην συνέχεια ακολουθούν πρακτικά παραδείγματα που αποδεικνύουν την ορθή λειτουργία της εφαρμογής και επιβεβαιώνουν την ολοκληρωμένη διεκπεραίωση όλων των στόχων που ορίστηκαν κατά τον σχεδιασμό της. Η αξιολόγηση της εφαρμογής θα γίνει αφενός παρουσιάζοντας την ορθή διεκπεραίωση των λειτουργιών που υλοποιεί και αφετέρου εξετάζοντας την ικανοποιητική απόδοση της, όσον αφορά τα αποτελέσματα που παράγει.

## 2. Δημιουργία Classification Μοντέλου Μηχανικής Μάθησης

### 2..1 Σύνολο Δεδομένων Classification

Για την αξιολόγηση της δημιουργίας Classification μοντέλου Μηχανικής μάθησης έγινε χρήση του κλασσικού συνόλου δεδομένων "Iris".<sup>[19]</sup> Το σύνολο δεδομένων αποτελείται από 5 στίλες, εκ των οποίων η τελευταία αποτελεί το target χαρακτηριστικό. Συγκεκριμένα τα χαρακτηριστικά είναι τα εξής:

1. sepal length - Μήκος Σεπτάλου (εκ.)
2. sepal width - Πλάτος Σεπτάλου (εκ.)
3. petal length - Μήκος Πετάλου (εκ.)

4. petal width - Πλάτος Πετάλου (εκ.)

5. class - Κατηγορία Iris

(α') Iris Setosa

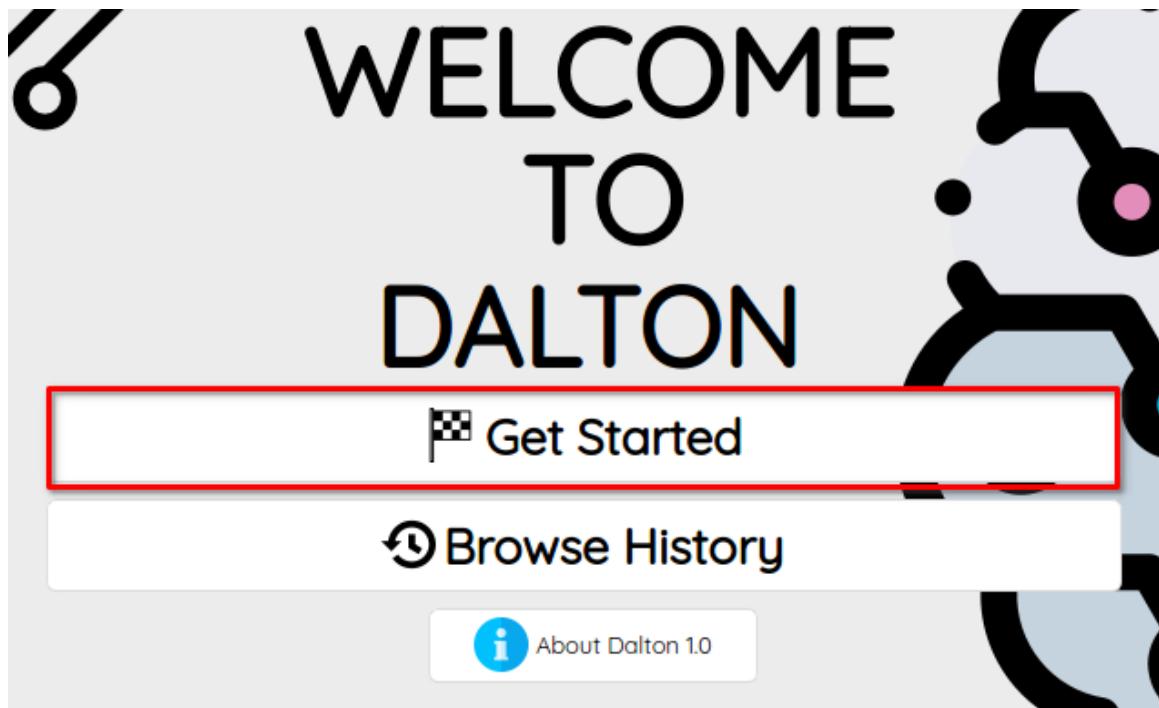
(β') Iris Versicolour

(γ') Iris Virginica

Σκοπός είναι να ταξινόμηση iris λουλουδιών στις 3 διακριτές κατηγορίες (Setosa, Versicolour, Virginica).

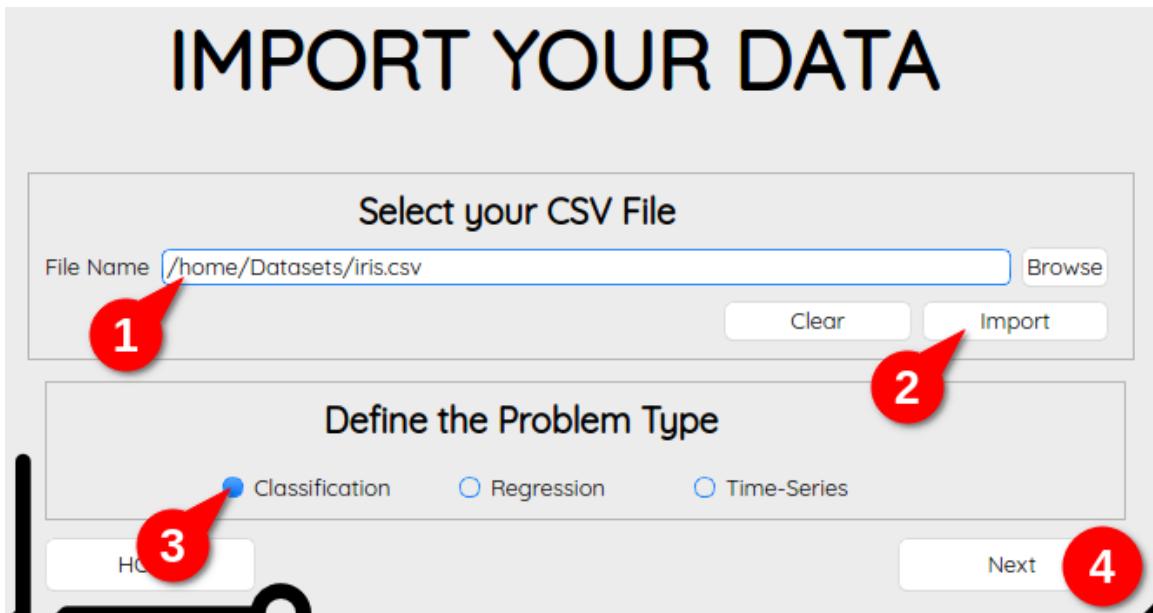
## 2..2 Περιγραφή Διαδικασίας

**Βήμα 1:** Ξεκινώντας, στην αρχική οθόνη του εργαλείου επιλέγουμε Get Started όπως φαίνεται στην εικόνα 5.12.



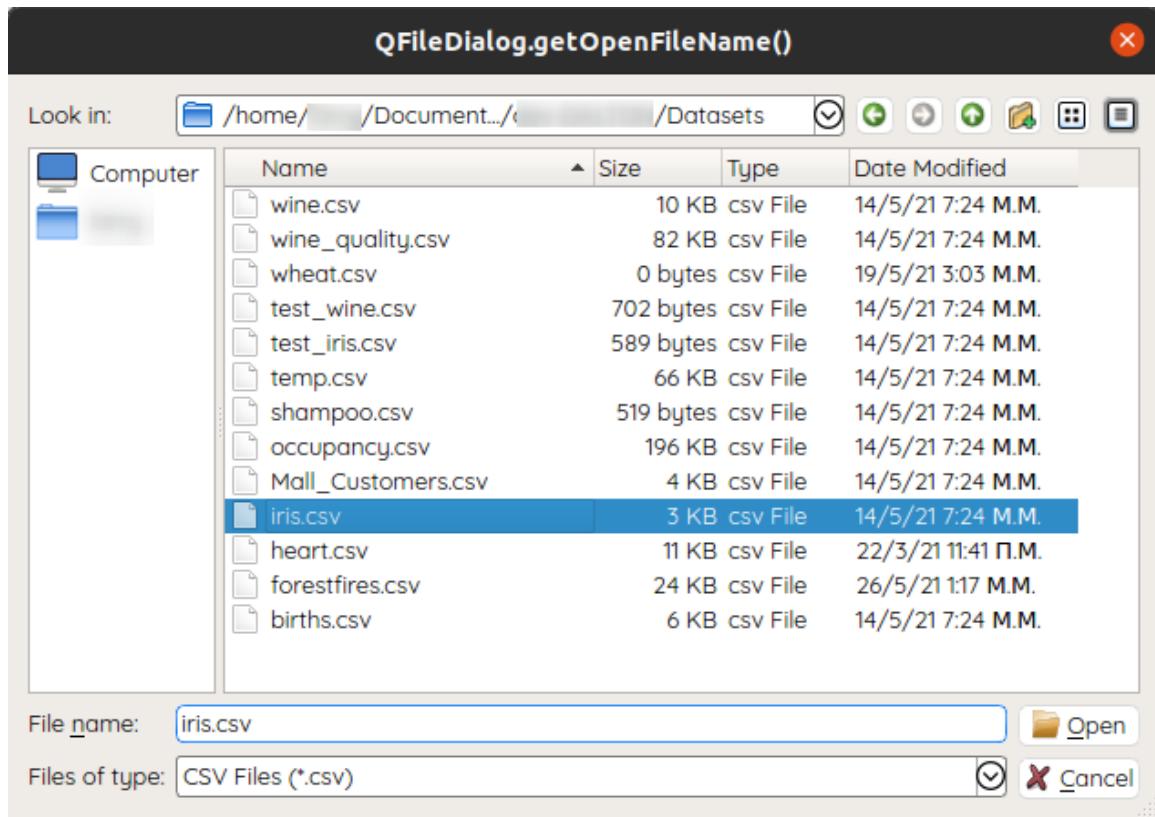
Σχήμα 5.12: Εκκίνηση

**Βήμα 2:** Στο δεύτερο βήμα της διαδικασίας ακολουθούμε την σειρά βημάτων που παρουσιάζονται στο σχήμα 5.13.



Σχήμα 5.13: Εισαγωγή αρχείου - Επιλογή τύπου προβλήματος

1. Παρέχουμε το μονοπάτι στο οποίο βρίσκεται το αρχείο iris.csv με το σύνολο δεδομένων μας. Εναλλακτικά μέσω του Browse κουμπιού αναζητάμε το αρχείο και το εισάγουμε από το σύστημα αρχείων μας.



Σχήμα 5.14: Αναζήτηση αρχείων

2. Εισάγουμε το αρχείο μέσω του Import κουμπιού.
3. Επιλέγουμε Classification ως τον τύπο προβλήματος μηχανικής μάθησης.
4. Επιλέγουμε Next.

**Βήμα 3:** Επιλέγουμε το Target χαρακτηριστικό μας από την dropdown λίστα όπως φαίνεται στο σχήμα 5.15. Συγκεκριμένα το χαρακτηριστικό που θέλουμε να προβλέψουμε είναι η στήλη Variety.

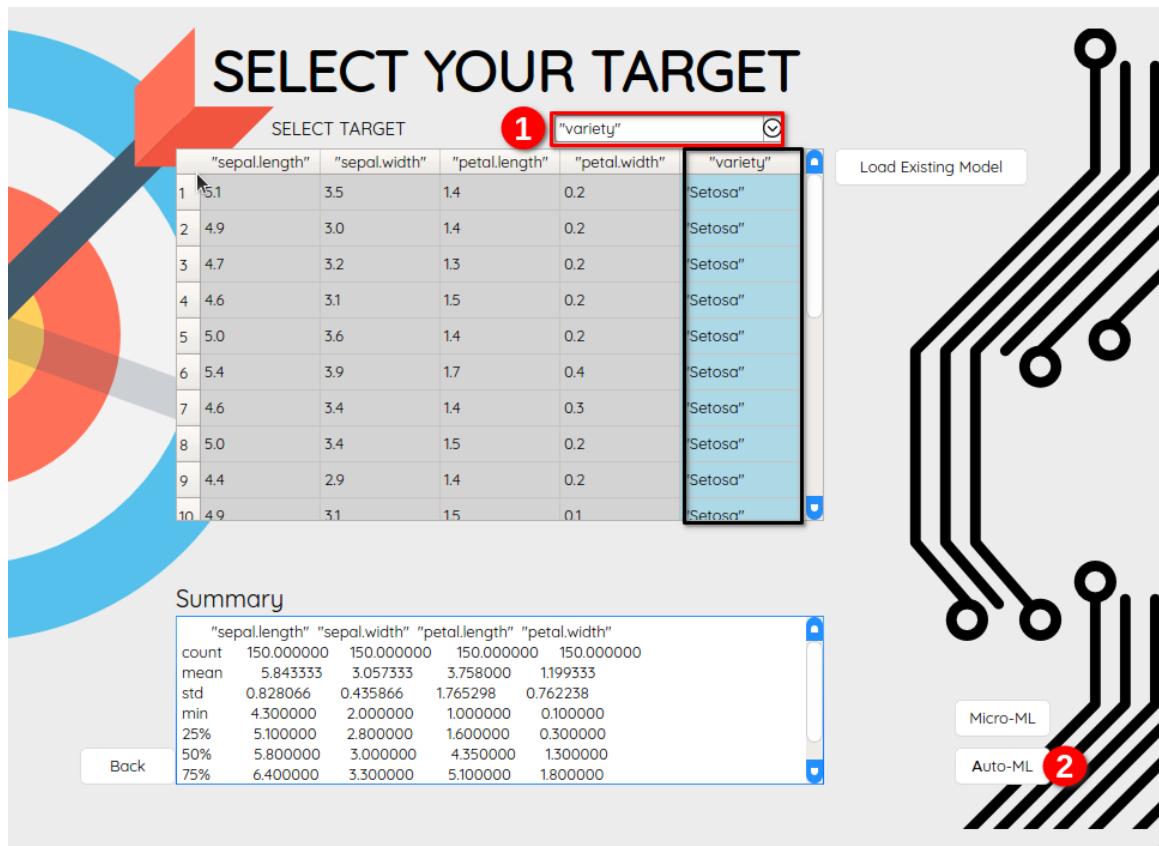
**SELECT TARGET**

	"sepal.length"	"sepal.width"	"petal.length"		
1	5.1	3.5	1.4	0.2	"Setosa"
2	4.9	3.0	1.4	0.2	"Setosa"
3	4.7	3.2	1.3	0.4	"Setosa"
4	4.6	3.1	1.5	0.3	"Setosa"
5	5.0	3.6	1.4	0.2	"Setosa"
6	5.4	3.9	1.7	0.2	"Setosa"
7	4.6	3.4	1.4	0.2	"Setosa"
8	5.0	3.4	1.5	0.2	"Setosa"
9	4.4	2.9	1.4	0.1	"Setosa"
10	4.9	3.1	1.5		

Σχήμα 5.15: Επιλογή Target Μεταβλητής

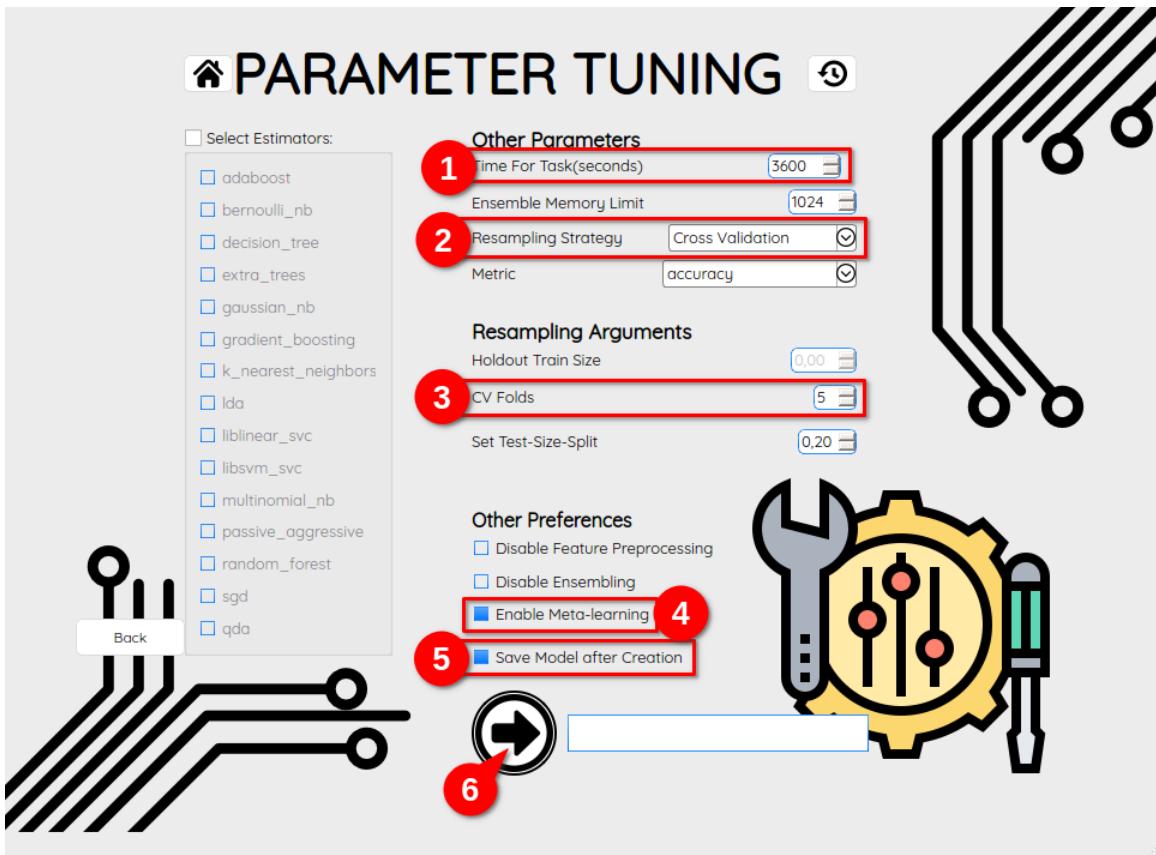
Στο σχήμα 5.16 φαίνονται τα βήματα της διαδικασίας.

1. Επιλέγουμε το Target χαρακτηριστικό.
2. Επιλέγουμε Auto-ML.



Σχήμα 5.16: Επιλογή Target

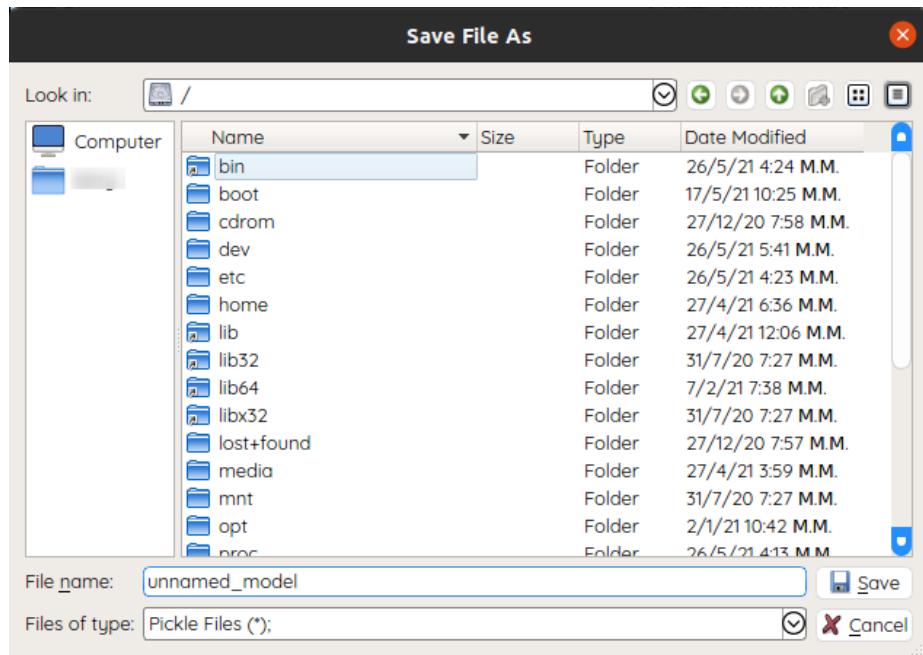
**Βήμα 4:** Στο βήμα 4 βρισκόμαστε πλέον στην οθόνη παραμετροποίησης και δημιουργίας του μοντέλου. Παραμετροποιούμε την διαδικασία μοντελοποίησης όπως φαίνεται στο σχήμα 5.17:



Σχήμα 5.17: Ρύθμιση Παραμέτρων

1. Θέτω τον χρόνο εύρεσης μοντέλων στην 1 ώρα.
2. Θέτω την Resampling στρατηγική σε Cross Validation.
3. Επιλέγω τα Cross Validation Folds να είναι 5.
4. Ενεργοποιώ την επιλογή Meta-Learning.
5. Επιλέγω να αποθηκευτεί το μοντέλο για την αξιοποίηση του αργότερα.
6. Ξεκινάω την διαδικασία.

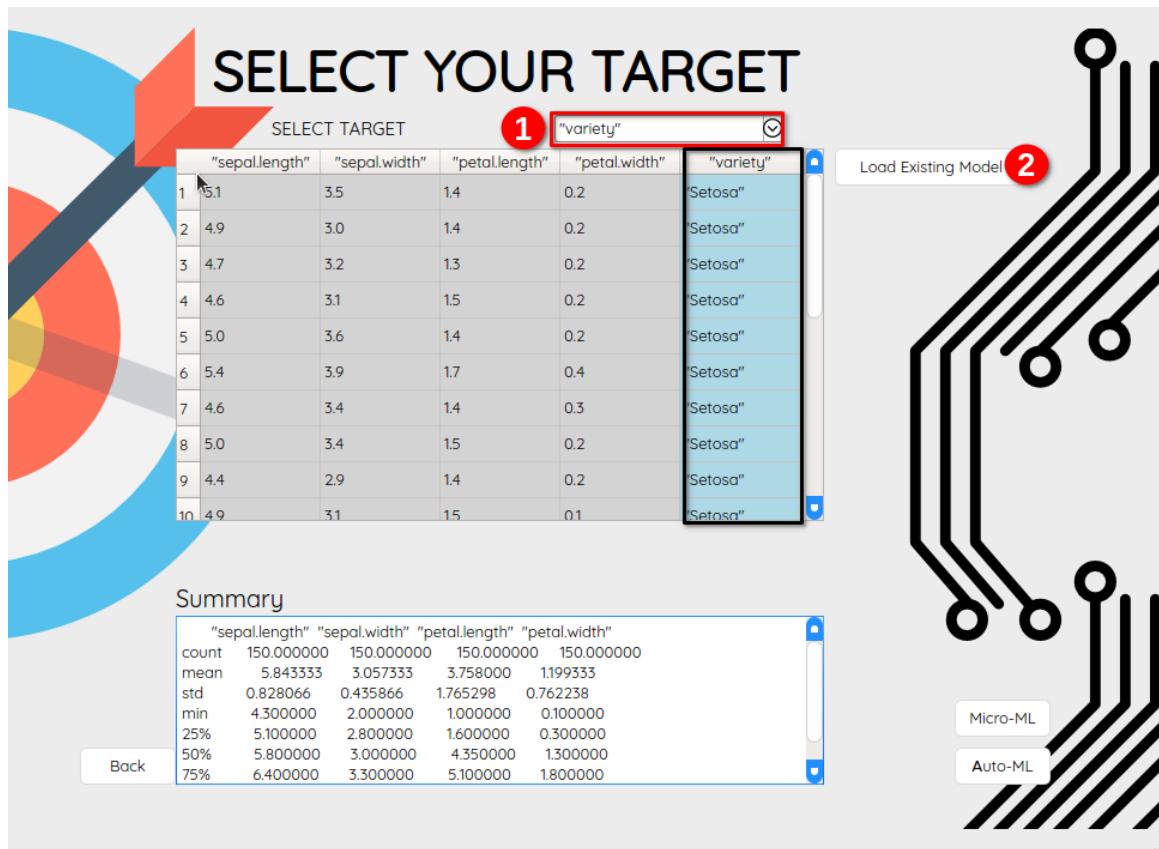
Περιμένουμε να ολοκληρωθεί η διαδικασία και να δημιουργηθεί το συνδυαστικό μοντέλο auto-sklearn. Μόλις η διαδικασία ολοκληρωθεί θα κληθούμε να αποθηκεύσουμε το μοντέλο μας στο σύστημα αρχείων του υπολογιστή μας όπως φαίνεται στο σχήμα 5.18. Μετά την αποθήκευση το μονοπάτι που αποθηκεύτηκε το μοντέλο μας θα εισαχθεί και στην βάση δεδομένων μας μαζί με το όνομα που δώσαμε στο αρχείο.



Σχήμα 5.18: Αποθήκευση Μοντέλου

### 2..3 Ανάκτηση Μοντέλου και Αξιολόγηση Αποτελέσματος

Για την αξιολόγηση του αποτελέσματος του μοντέλου μας ακολουθούμε τα βήματα 1, 2 και 3 της διαδικασίας 2.2. Στην συνέχεια επιλέγουμε Load Existing Model όπως φαίνεται στο σχήμα 5.19.



Σχήμα 5.19: Ανάκτηση από την Βάση Δεδομένων

Στην οθόνη που προκύπτει ακολουθούμε τις ενέργειες που φαίνονται στο σχήμα.

**MODELS**

Model Name	TimeStamp
1_model	2021-05-18 10:40:20.000
2 unnamed_model	2021-05-19 18:40:13.739
3 unnamed_model	2021-05-20 12:50:58.790
4 iris_model	2021-05-20 13:00:28.497
5 wine_cls	2021-05-20 13:41:07.790
6 wine-10fold-30sec	2021-05-20 14:52:59.427
7 iris_1hr	2021-05-20 17:00:00.039
8 heart_def	2021-05-21 13:13:37.713
9 heart_disease_1hr	2021-05-21 15:05:18.956
10 unnamed_model	2021-05-22 11:49:09.600
11 mse	2021-05-22 12:51:05.060
12 wine_quality_1hr	2021-05-26 20:37:25.470
13 car_eval_1hr	2021-05-27 00:32:35.714
14 occupancy_1hr	2021-05-27 01:41:47.087
15 iris_1hr_5folds	2021-05-28 09:28:29

Load Model 2

auto-sklearn results:  
Dataset name: iris  
Metric: accuracy  
Best validation score: 0.983333  
Number of target algorithm runs: 385  
Number of successful target algorithm runs: 381  
Number of crashed target algorithm runs: 4  
Number of target algorithms that exceeded the time limit: 0  
Number of target algorithms that exceeded the memory limit: 0

PREDICTION RESULTS

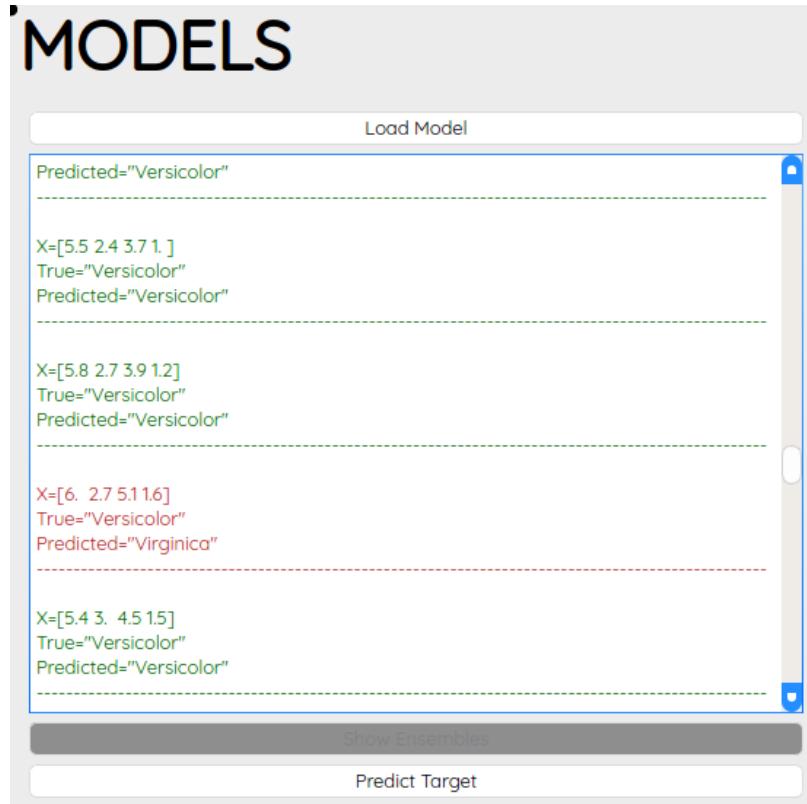
```
Accuracy: 0.9933333333333333
Balanced Acc: 0.9933333333333333
Precision Micro: 0.9933333333333333
Precision Macro: 0.9934640522875817
Precision Weighted: 0.9934640522875816
Recall Micro: 0.9933333333333333
Recall Macro: 0.9933333333333333
Recall Weighted: 0.9933333333333333
F1 Micro: 0.9933333333333333
```

Show Ensemble 3

Predict Target

Σχήμα 5.20: Ανάκτηση Iris Μοντέλου

1. Επιλέγουμε το μοντέλο μας από την λίστα.
2. Επιλέγουμε Load Model. Μέσω αυτής της επιλογής βλέπουμε μερικές πληροφορίες για το μοντέλο μας.
3. Επιλέγουμε Predict Target. Μέσω της επιλογής αυτής βλέπουμε αναλυτικά τις προβλέψεις του μοντέλου μας στο σύνολο δεδομένων που αναζήσαμε καθώς και την απόδοση του σύμφωνα με διάφορες μετρικές.



Σχήμα 5.21: Αναλυτικές προβλέψεις του Iris Μοντέλου

Συμπερασματικά βλέπουμε πως η απόδοση του μοντέλου μας είναι πολύ καλή καθώς πετυχαίνει υψηλό ποσοστό για όλες τις μετρικές που είναι εμφανείς στο σχήμα 5.20. Οι μετρικές υπολογίζονται συγκρίνοντας τις προβλέψεις του μοντέλου μας πάνω στο σύνολο δεδομένων *iris* με τις πραγματικές κλάσεις των δειγμάτων.

### 3. Δημιουργία Regression Μοντέλου Μηχανικής Μάθησης

#### 3..1 Σύνολο Δεδομένων Regression

Για την αξιολόγηση της διαδικασίας δημιουργίας μοντέλου μηχανικής μάθησης για Regression προβλήματα έγινε χρήση του συνόλου δεδομένων "Wine Quality". [19] [16]

**Περιγραφή:** Το συγκεκριμένο σύνολο δεδομένων αφορά στην αξιολόγηση της ποιότητας κρασιών, βαθμολογώντας το από 0 έως 10, σύμφωνα με πλήθος χαρακτηριστικών τους. Το σύνολο δεδομένων περιλαμβάνει 12 στήλες εκ των οποίων η τελευταία αποτελεί το target χαρακτηριστικό μας, δηλαδή την ποιότητα του κρασιού. Συγκεκριμένα οι στήλες αποτελούν τα παρακάτω χαρακτηριστικά:

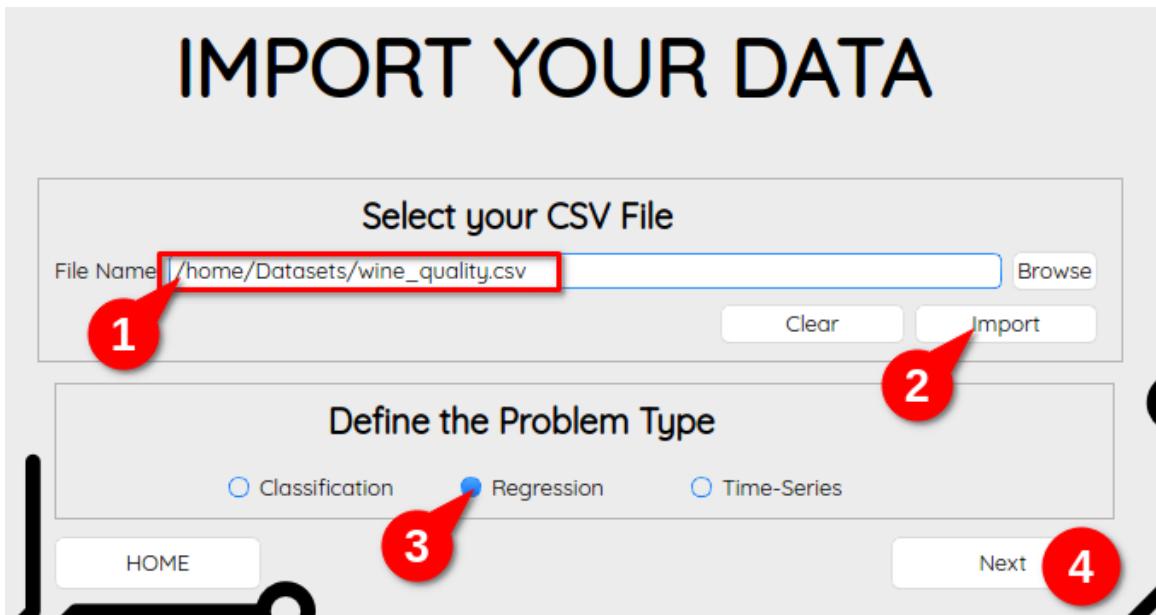
1. fixed acidity
2. fixed acidity
3. volatile acidity
4. citric acid
5. residual sugar
6. chlorides
7. free sulfur dioxide
8. total sulfur dioxide
9. density
10. pH
11. sulphates
12. alcohol
13. Ποιότητα: quality

Σκοπός είναι η δημιουργία μοντέλου Regression που θα προσεγγίζει όσο το δυνατόν καλύτερα την ποιότητα του κρασιού λαμβάνοντας υπόψιν τα χαρακτηριστικά του.

### 3..2 Περιγραφή Διαδικασίας

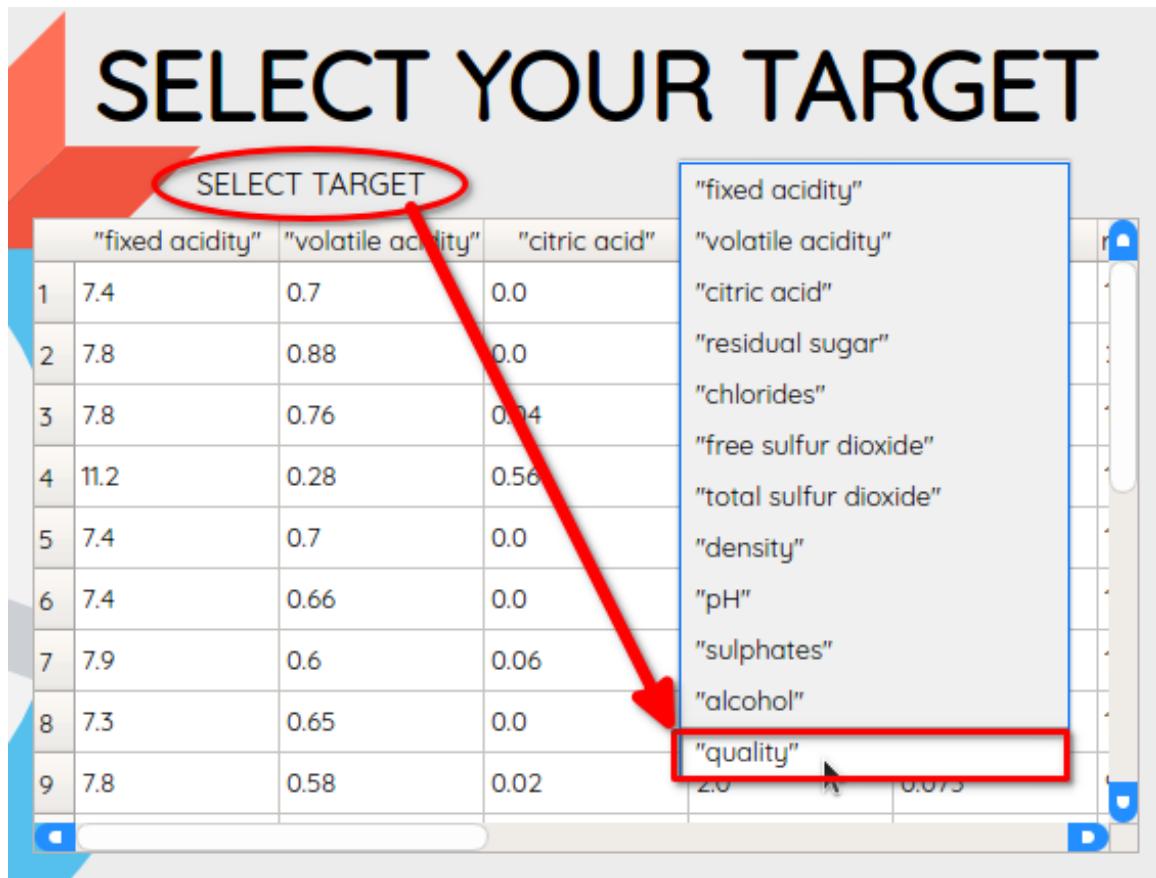
**Βήμα 1:** Το πρώτο βήμα ταυτίζεται με το Βήμα 1 της υπο-ενότητας [2.2](#).

**Βήμα 2:** Το βήμα 2 αποτελεί παρόμοια διαδικασία με το βήμα 2 της υπο-ενότητας 2.2. Στην περίπτωση αυτή θα επιλέξουμε το αρχείο wine\_quality.csv και ως τύπο προβλήματος θα επιλέξουμε Regression όπως φαίνεται στο παρακάτω σχήμα, 5.22. Στην συνέχεια επιλέγουμε Next.



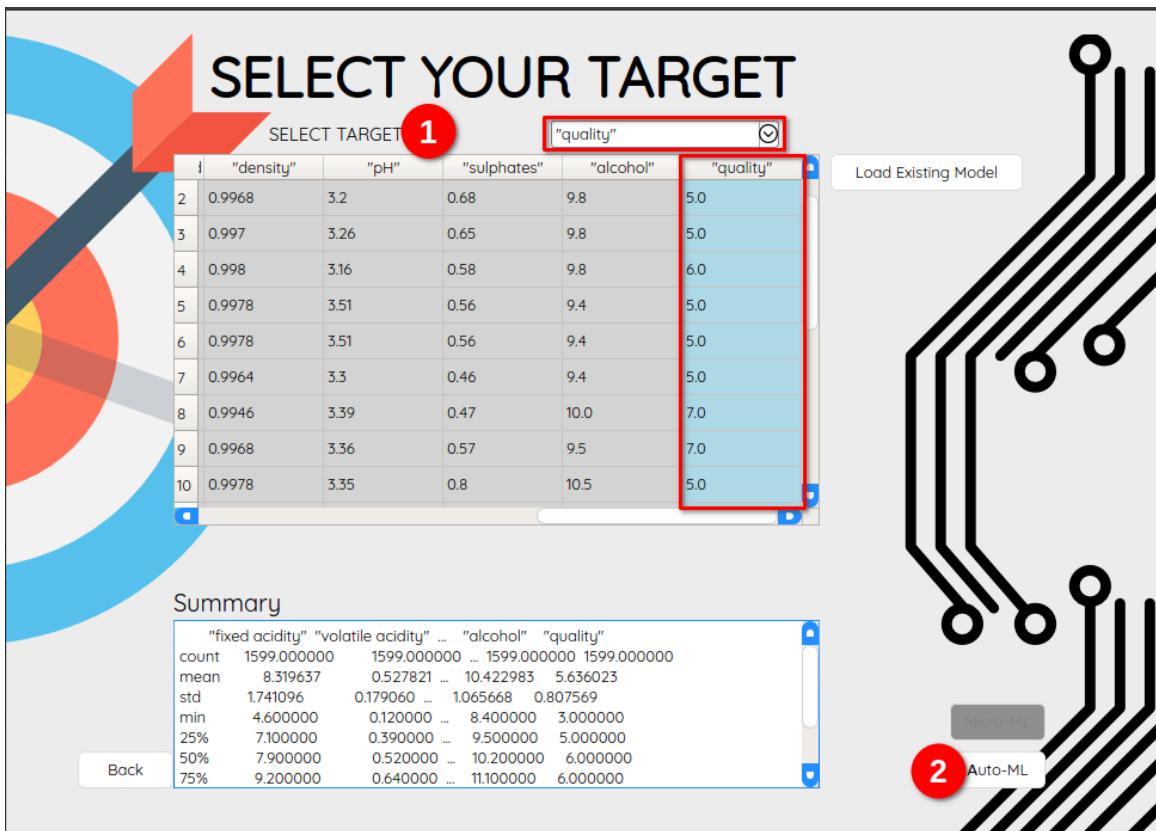
Σχήμα 5.22: Ανάκτηση από την Βάση Δεδομένων

**Βήμα 3:** Όπως και στην περίπτωση Classification που περιγράφεται στην υπο-ενότητα 2.2 μεταφερόμαστε στην οθόνη επιλογής Target χαρακτηριστικού.



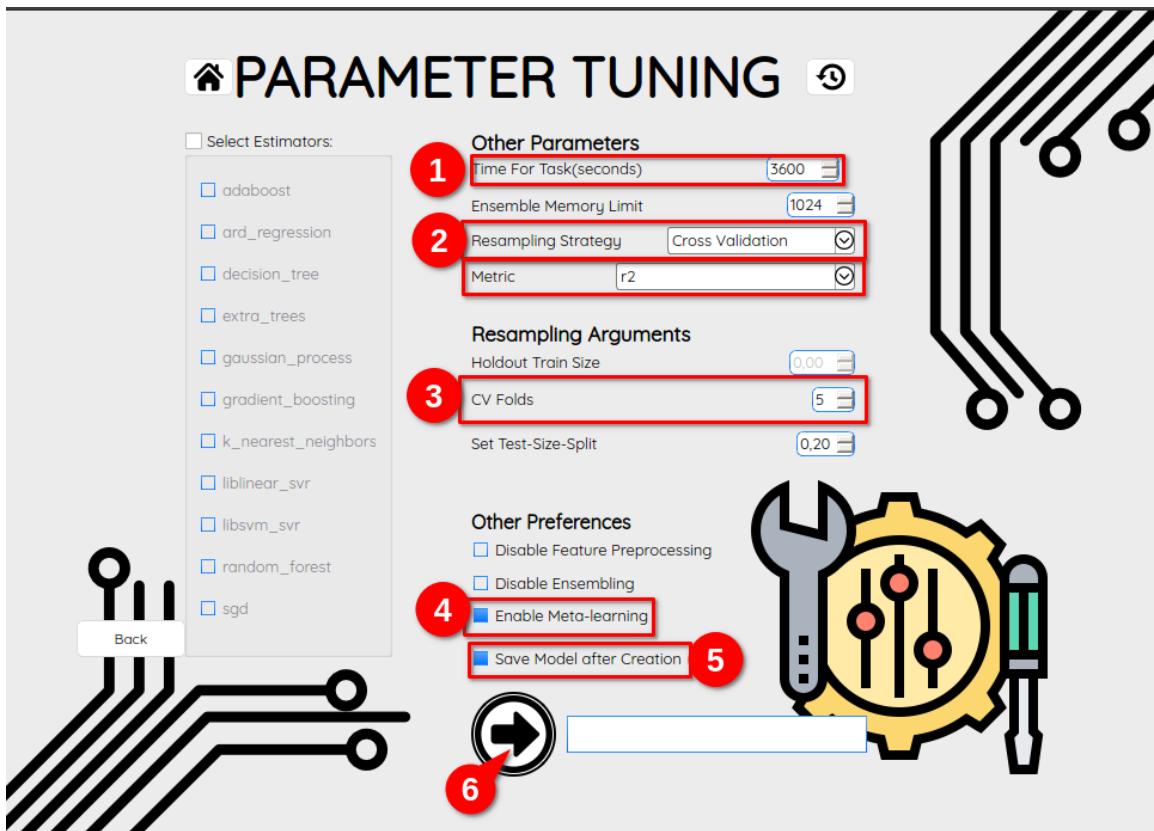
Σχήμα 5.23: Επιλογή Target Μεταβλητής

Στην περίπτωση αυτή επιλέγουμε ως target χαρακτηριστικό την στήλη quality και στην συνέχεια επιλέγουμε AutoML όπως φαίνεται στα σχήματα 5.23 και 5.24.



Σχήμα 5.24: Επιλογή Target Μεταβλητής

**Βήμα 4:** Στο βήμα 4 βρισκόμαστε στην οθόνη ρύθμισης των παραμέτρων και την δημιουργία του συνδυαστικού μοντέλου μπχανικής μάθησης. Ορίζουμε τις παραμέτρους όπως φαίνεται στο παρακάτω σχήμα, 5.25.



Σχήμα 5.25: Επιλογή Target Μεταβλητής

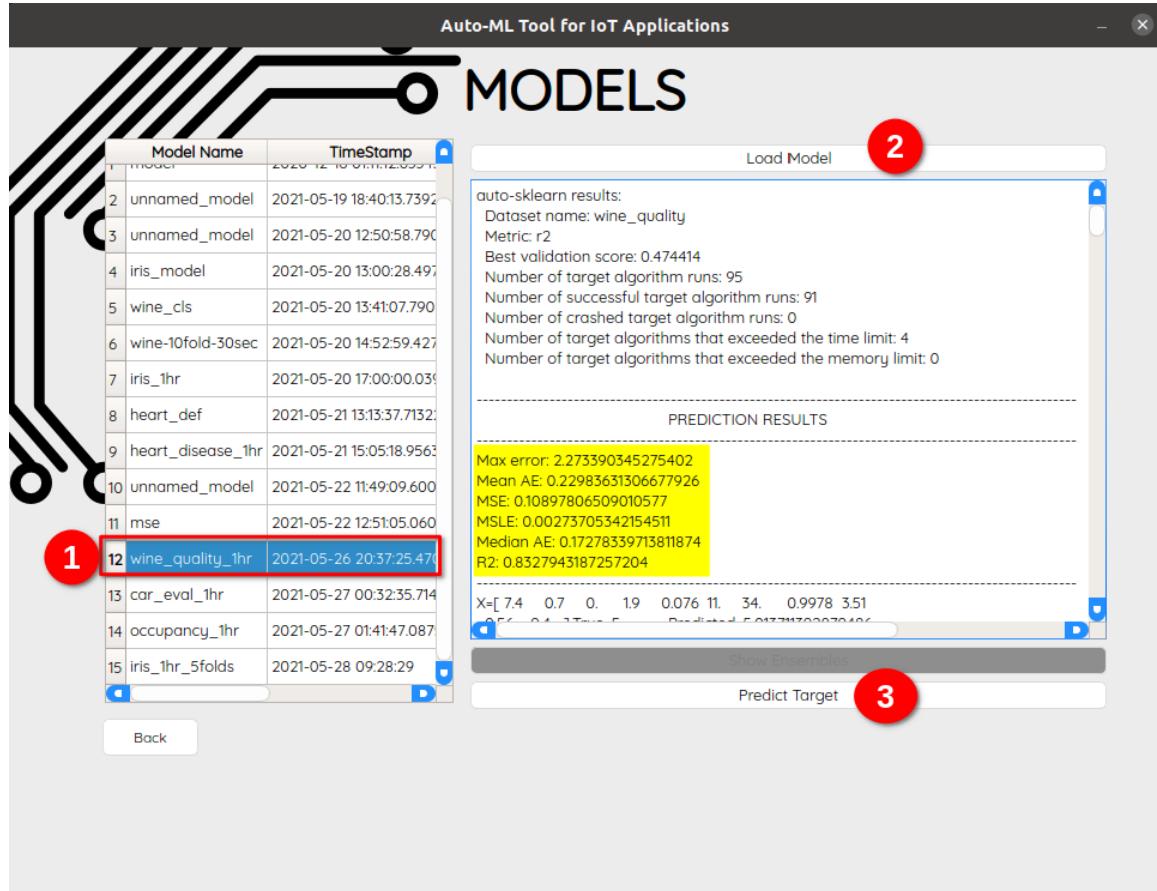
1. Ορίζουμε τον χρόνο εκτέλεσης της διαδικασίας στην 1 ώρα.
2. Ορίζουμε το Resampling Strategy ως Cross Validation και την Μετρική ως r2.
3. Ορίζουμε το πλήθος των Folds του Cross Validation να είναι 5.
4. Ενεργοποιούμε την επιλογή Meta Learning.
5. Επιλέγουμε να αποθηκευτεί το μοντέλο μετά την δημιουργία.
6. Ξεκινάμε την διαδικασία.

Μόλις ολοκληρωθεί η δημιουργία του μοντέλου μηχανικής μάθησης, μια ώρα αργότερα, καλούμαστε να αποθηκεύσουμε το μοντέλο στο σύστημα αρχείων του υπολογιστή μας, όπως φαίνεται στο σχήμα 5.18. Ταυτόχρονα το μονοπάτι του αρχείου θα αποθηκευτεί στην βάση δεδομένων μας μαζί με το όνομα που έχουμε δώσει κατά την διαδικασία αποθήκευσης.

### 3..3 Ανάκτηση Μοντέλου και Αξιολόγηση Αποτελέσματος

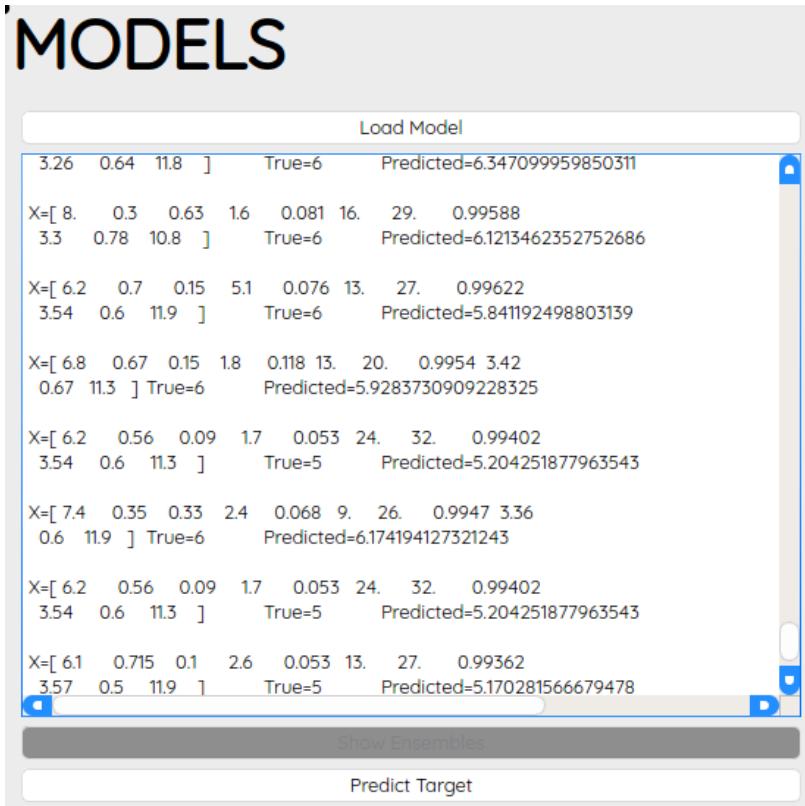
Για την αξιολόγηση του αποτελέσματος του μοντέλου μας ακολουθούμε τα βήματα 1, 2 και 3 της διαδικασίας 3.2. Στην συνέχεια επιλέγουμε Load Existing Model και όχι Auto-ML ή Micro-ML (βλ. 5.19).

Στην οθόνη που προκύπτει ακολουθούμε τις ενέργειες που φαίνονται στο σχήμα.



Σχήμα 5.26: Ανάκτηση Wine Quality Μοντέλου

1. Επιλέγουμε το μοντέλο μας από την λίστα.
2. Επιλέγουμε Load Model. Μέσω αυτής της επιλογής βλέπουμε μερικές πληροφορίες για το μοντέλο μας.
3. Επιλέγουμε Predict Target. Μέσω της επιλογής αυτής βλέπουμε αναλυτικά τις προβλέψεις του μοντέλου μας στο σύνολο δεδομένων που αναρτήσαμε καθώς και την απόδοση του σύμφωνα διάφορες μετρκές.



Σχήμα 5.27: Αναλυτικές προβλέψεις του Wine Quality Μοντέλου

Συμπερασματικά, παρατηρούμε πως το παραγόμενο μοντέλο έχει πολύ καλή απόδοση σύμφωνα με πλήθος μετρικών που είναι εμφανείς στο σχήμα 5.26. Οι μετρικές υπολογίζονται συγκρίνοντας τις πραγματικές τιμές των δειγμάτων με τις προβλέψεις του μοντέλου μας. Βλέπουμε πως η απόκλιση των προβλέψεων είναι μικρή παρατηρώντας τις αναλυτικές προβλέψεις του μοντέλου και τις πραγματικές τιμές.

## 4. Μετατροπή Προβλήματος Time Series σε Κλασσικό Πρόβλημα Μηχανικής Μάθησης

### 4..1 Σύνολο Δεδομένων Timeseries

Για την δοκιμή της λειτουργίας μετατροπής προβλήματος Time Series σε κλασσικό πρόβλημα Classification έγινε χρήση του συνόλου δεδομένων "Occupancy Detection".[29]

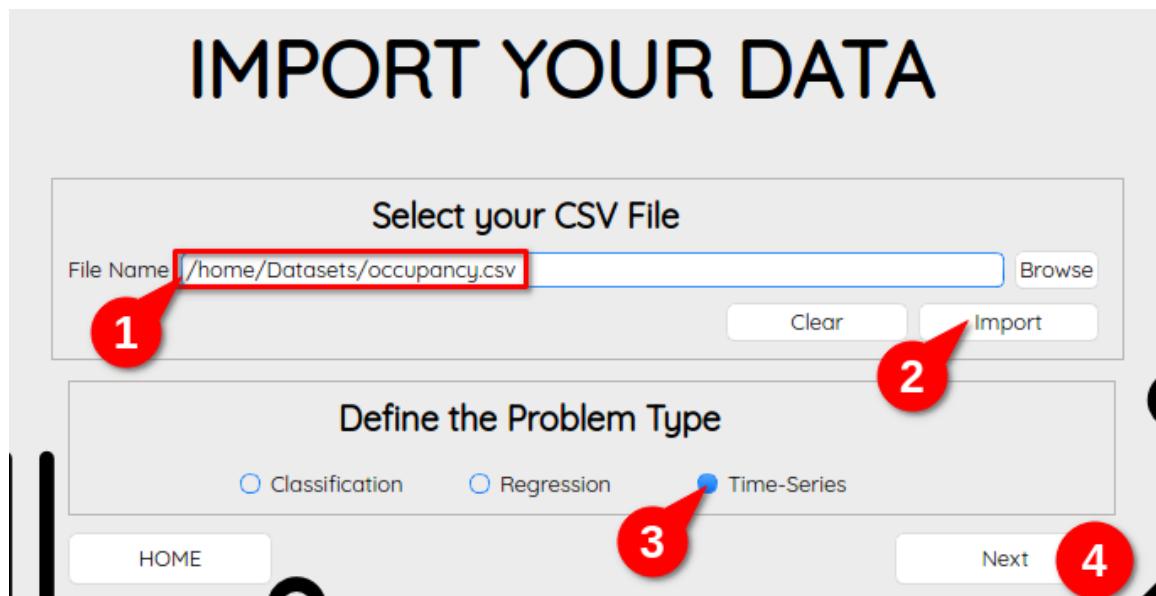
**Περιγραφή:** Το σύνολο δεδομένων αποτελείται από 5 στήλες εκ των οποίων η τελευταία αποτελεί την Target μεταβλητή μας. Οι υπόλοιπες 4 στήλες είναι οι εξής: Abstract: Experimental data used for binary classification (room occupancy) from Temperature, Humidity, Light and CO<sub>2</sub>. Ground-truth occupancy was obtained from time stamped pictures that were taken every minute.

1. Temperature - Θερμοκρασία δωματίου
2. Humidity - Υγρασία δωματίου
3. Light - Φώς
4. CO<sub>2</sub>
5. Occupancy (0 ή 1)

Σκοπός είναι η πρόβλεψη του αν ένα δωμάτιο είναι κατειλημμένο ή όχι.

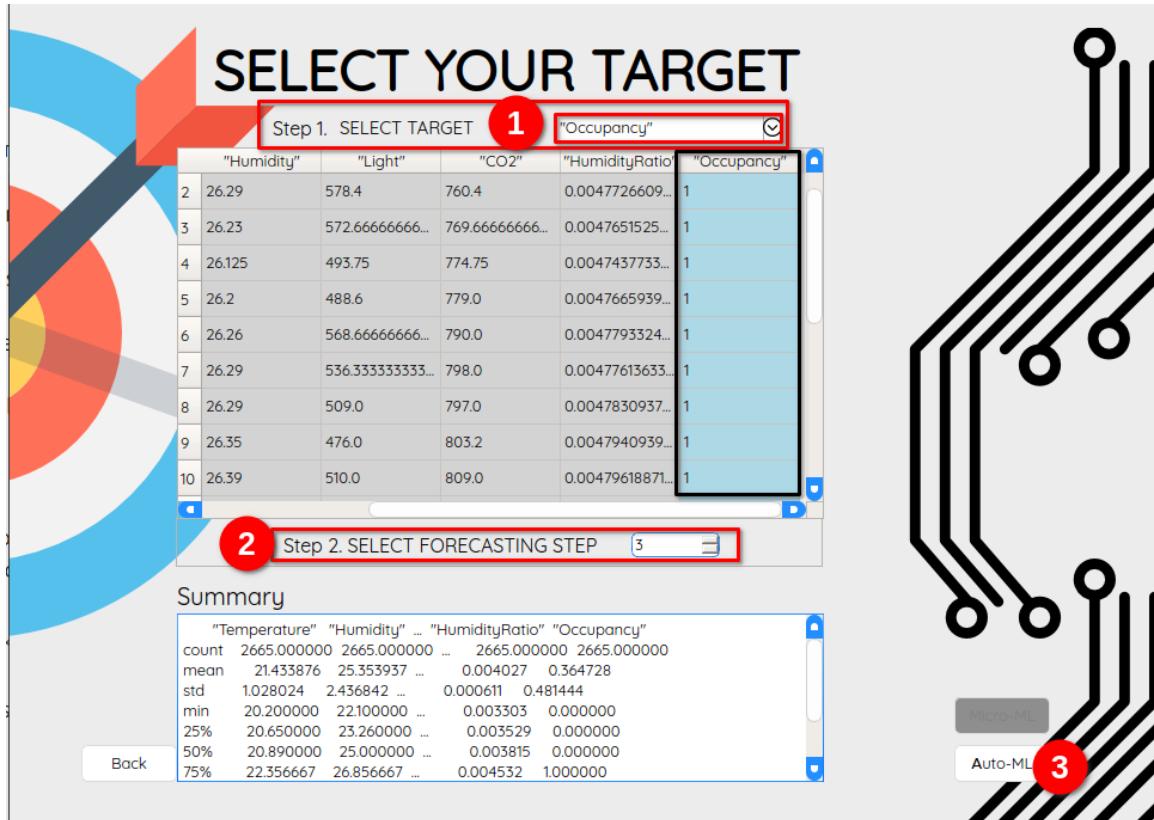
#### 4..2 Περιγραφή Διαδικασίας

**Βήμα 1:** Το πρώτο βήμα ταυτίζεται με το Βήμα 1 της υπο-ενότητας [2.2](#). Εισάγουμε το επιθυμητό αρχείο στην εφαρμογή, επιλέγουμε Import και στην συνέχεια επιλέγουμε ως τύπο προβλήματος Time-Series. Στην συνέχεια επιλέγουμε Next.



Σχήμα 5.28: Εισαγωγή συνόλου δεδομένων και επιλογή τύπου προβλήματος

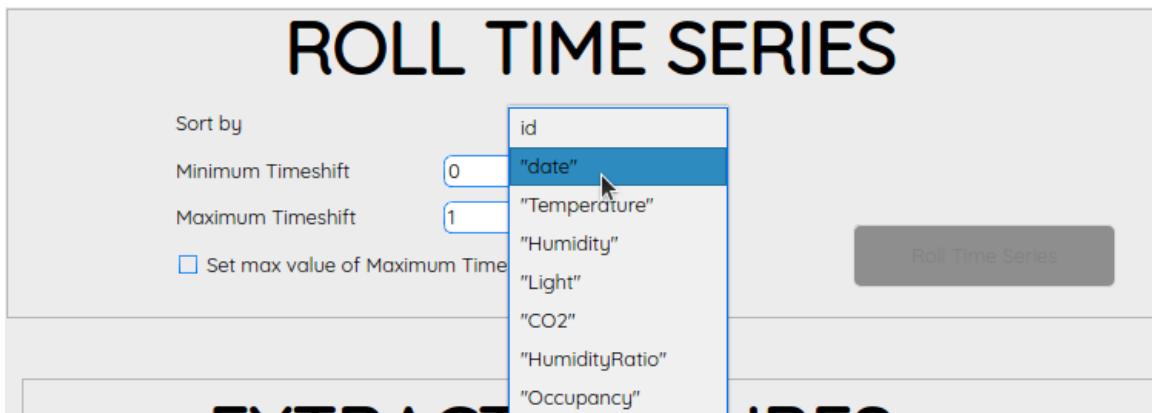
**Βήμα 2:** Επιλέγουμε αρχικά το Target χαρακτηριστικό όπως φαίνεται και στην συνέχεια θα επιλέξουμε το βήμα πρόβλεψης να είναι 3 όπως αναπαρίσταται στο σχήμα 5.29.



Σχήμα 5.29: Επιλογή Target Μεταβλητής και Βήματος Πρόβλεψης

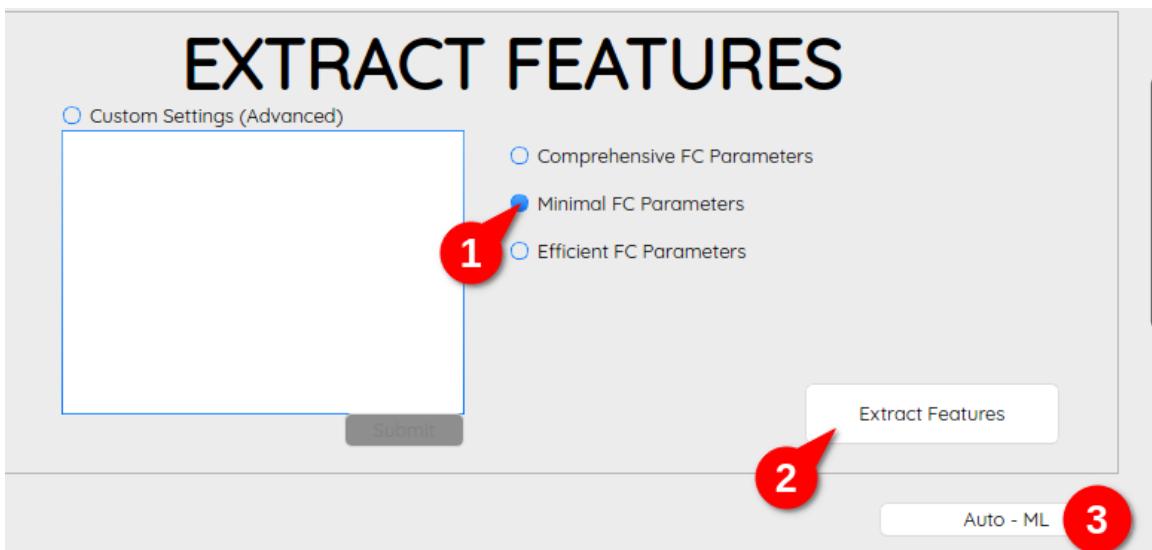
Τέλος επιλέγουμε Auto-ML.

**Βήμα 3:** Στο βήμα 3 μεταφερόμαστε στην οθόνη που υλοποιούνται οι διαδικασίες Roll και Extract. Αρχικά επιλέγουμε την στήλη σύμφωνα με την οποία τα δεδομένα μας ταξινομούνται στον χρόνο. Στην περίπτωσή μας η στήλη αυτή είναι η date. στην συνέχεια επιλέγουμε Roll Time Series.



Σχήμα 5.30: Rolling Διαδικασία

Έπειτα από το επιτυχημένο Roll, επιλέγουμε Minimal FC Parameters για την εξαγωγή των βασικών χαρακτηριστικών από το σύνολο δεδομένων που έχουμε εισάγει στο εργαλείο και στην συνέχεια επιλέγουμε Extract Features οπως φαίνεται στο σχήμα 5.31.



Σχήμα 5.31: Extract Διαδικασία

Τέλος επιλέγουμε Auto-ML.

**Βήμα 5:** Καλούμαστε να επιλέξουμε τον τύπο προβλήματος και ταυτόχρονα βλέπουμε ένα δείγμα του παραγόμενου συνόλου δεδομένων στην οθόνη μας

(Σχ. 5.32). Επιλέγουμε Classification εφόσον η Target μεταβλητή μας είναι μία δυαδική τιμή. Τέλος επιλέγουμε Next.

The screenshot shows the KNIME 'PREVIEW' interface. At the top, it says 'PREVIEW'. Below that, there are two tables: 'PREDICTORS ( extracted features )' and 'TARGET VARIABLE'.

**PREDICTORS ( extracted features )**

	upancy"__min	"light"__minimur	atio"__standarc	idityRatio"__var	nperature"__
1	1.0	585.2	0.0	0.0	23.7
2	1.0	578.4	4.2489839805...	1.80538648667...	23.70900000
3	1.0	572.66666666...	3.794312876611...	1.43968102056...	23.71600000
4	1.0	493.75	1.07146961650...	1.14804713909...	23.717625
5	1.0	488.6	9.8029649038...	9.6098120906...	23.7249
6	1.0	488.6	1.09352342416...	1.19579347919...	23.73075
7	1.0	488.6	1.08134189992...	1.16930030453...	23.73064285
8	1.0	488.6	1.14561335308...	1.31242995476...	23.7335625
9	1.0	476.0	1.33990050374...	1.79533335994...	23.73583333
10	1.0	476.0	1.468671391587...	2.15699565646...	23.73585
11	1.0	476.0	1.72043168770...	2.9598851920...	23.73668181E
12	1.0	476.0	1.97633960369...	3.90591822913...	23.733625

**TARGET VARIABLE**

	"Occupancy"
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1

**Define the Problem Type**

Classification  Regression

**Buttons at the bottom:**

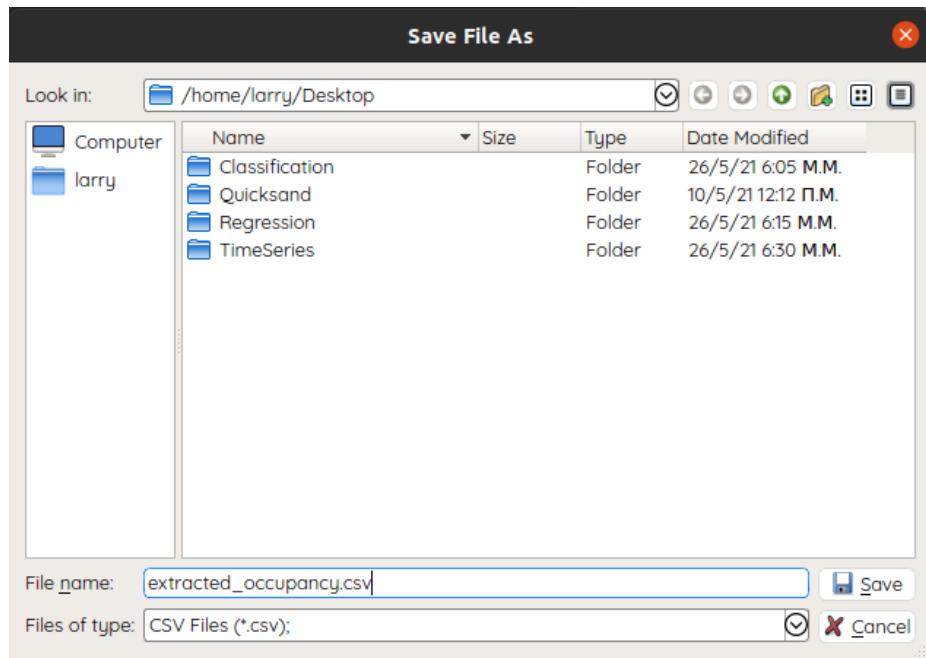
- Save Dataset (circled with red circle 3)
- Back
- Next (circled with red circle 2)

Σχήμα 5.32: Εμφάνιση predictors και target στηλών

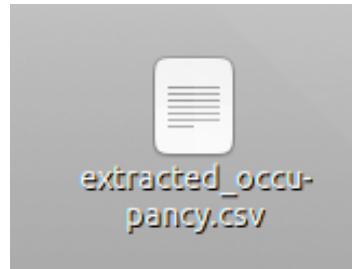
**Βήμα 6:** Αντιμετωπίζουμε το πρόβλημα μας ως ενα κοινό πρόβλημα Μηχανικής Μάθησης εφόσον βρισκόμαστε πλέον στην θύμιση των παραμέτρων για την μοντελοποίηση των δεδομένων μας.

#### 4..3 Εξαγωγή Συνόλου Δεδομένων

Μέσω της επιλογής 3 που φαίνεται στην εικόνα 5.32 έχουμε την δυνατότητα αποθήκευσης του παραγόμενου συνόλου δεδομένων μετά την εξαγωγή χαρακτηριστικών. Η διαδικασία φαίνεται στις εικόνες 5.33, 5.34 και 5.35.



Σχήμα 5.33: Αποθήκευση αρχείου με το σύνολο δεδομένων



Σχήμα 5.34: Αποθηκευμένο αρχείο τύπου csv

Σχήμα 5.35: Μορφή του εξαγόμενου συνόλου δεδομένων

Μπορούμε να χρησιμοποιήσουμε το αρχείο με σκοπό να δημιουργήσουμε μοντέλα μηχανικής μάθησης όπως ακριβώς περιγράφεται στην υποενότητα 2.2.

## 5. Ανάκτηση και Αξιοποίηση Αποθηκευμένων Μοντέλων Μηχανικής Μάθησης

Για την επιβεβαίωση της σωστής λειτουργίας ανάκτησης και επαναχρησιμοποίησης μοντέλων θα χρησιμοποιήσουμε ως παράδειγμα τα μοντέλα που δημιουργήσαμε στις προηγούμενες υποενότητες.

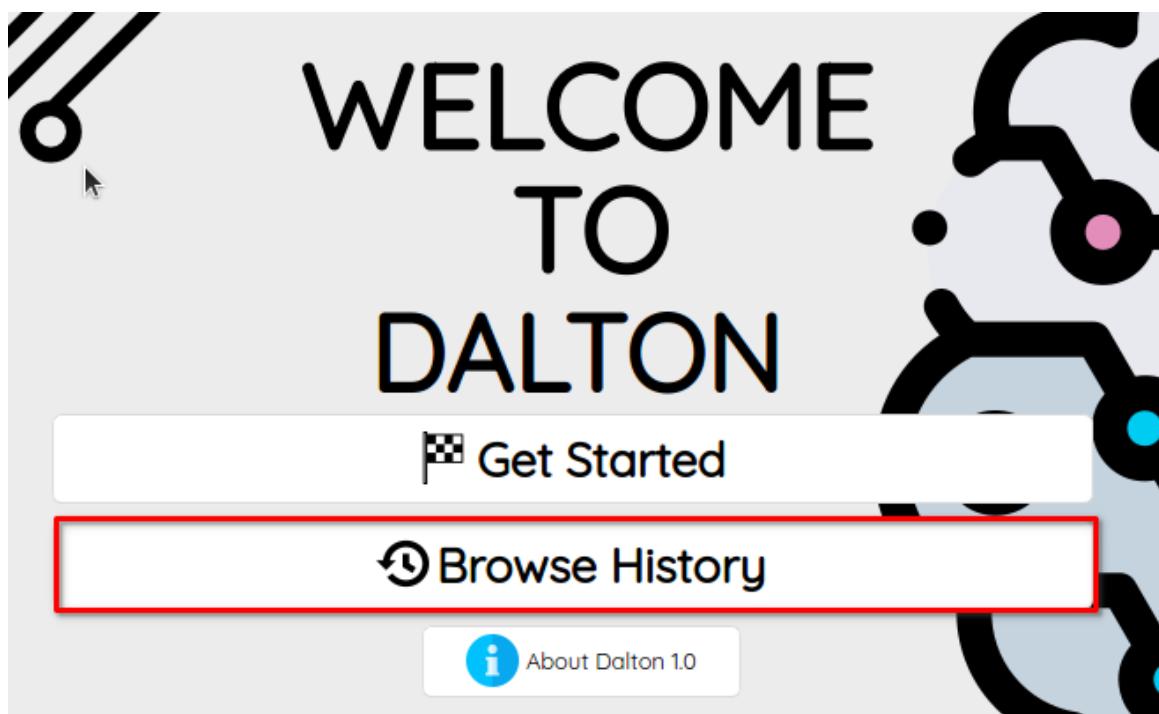
## 5..1 Ανάκτηση Μοντέλου μέσω της Βάσης Δεδομένων

### 5..1.1 Ανάκτηση μετά την διαδικασία επιλογής Target μεταβλητής

Αυτός ο τρόπος ανάκτησης περιγράφεται αναλυτικά στην διαδικασία αξιολόγησης του μοντέλου (βλ. 2..3)

### 5..1.2 Ανάκτηση μέσω του Ιστορικού

Η ανάκτηση των μοντέλων μπορεί επίσης να πραγματοποιηθεί μέσω του Ιστορικού. Από την αρχική οθόνη του εργαλείου επιλέγουμε Browse Model History όπως φαίνεται στο σχήμα 5.36.



Σχήμα 5.36: Ανάκτηση από την Βάση Δεδομένων

Μέσω αυτής της επιλογής μεταφερόμαστε στην οθόνη Model History όπου ακολουθούμε τις ενέργειες που φαίνονται στο σχήμα 5.37 για να ανακτήσουμε το επιθυμητό μοντέλο και να δούμε αναλυτικές πληροφορίες για αυτό.

The screenshot shows a 'MODEL HISTORY' interface. At the top, there are tabs for 'Classification Models' and 'Regression Models'. The 'Regression Models' tab is selected and highlighted with a red box and a red number '1' above it. Below the tabs is a table with columns 'Model Name' and 'TimeStamp'. The table contains four rows:

Model Name	TimeStamp
1 model	2020-12-18 01:11:12.633437
2 unnamed_model	2021-05-20 12:50:58.790554
3 mse	2021-05-22 12:51:05.060547
4 wine_quality_1hr	2021-05-26 20:37:25.470961

A red box surrounds the fourth row, and a red number '2' is placed to its left. To the right of the table is a yellow box containing the text 'auto-sklearn results:' followed by various dataset statistics. A red box surrounds this yellow box, and a red number '3' is placed above it. At the bottom right of the yellow box is a 'Show more...' button with a red number '4' placed above it.

Σχήμα 5.37: Επιλογή μοντέλου Wine Quality από την λίστα

1. Επιλέγουμε την Regression καρτέλα ώστε να εμφανιστούν τα διαθέσιμα αποθηκευμένα μοντέλα.
2. Επιλέγουμε το μοντέλο που επιθυμούμε από την λίστα.
3. Επιλέγουμε Show Model Summary για την εμφάνιση μιας σύντομης περιγραφής του μοντέλου στην οθόνη μας.
4. Επιλέγουμε Show more... για την εμφάνιση περισσότερων πληροφοριών για τα μοντέλα που αποτελούν το συνδυαστικό μοντέλο μας όπως φαίνεται στην εικόνα 5.38

The screenshot shows a 'MODEL HISTORY' interface. At the top, there's a decorative graphic of several parallel lines with circles at their ends. Below it, a 'MODEL LIST' title is centered. A table follows, with columns labeled 'Weight' and 'Information'. The first row contains a weight of '1' and a value of '0.34'. The 'Information' column displays a complex Python code snippet representing a 'SimpleRegressionPipeline' object. The code includes various preprocessing steps like 'category\_coalescer', 'quantile\_transformer', and 'feature\_preprocessor', along with a 'KNeighborsRegressor' component. At the bottom of the table is a 'Back' button.

	Weight	Information
1	0.34	<pre>SimpleRegressionPipeline({'data_processing:categorical_transformer:categorical_encoding:__choice__:': 'no_encoding', 'data_processing:categorical_transformer:category_coalescer:__choice__:': 'minority_coalescer', 'data_processing:numerical_transformer:imputation:strategy': 'median', 'data_processing:numerical_transformer:rescaling:__choice__:': 'quantile_transformer', 'feature_processor:__choice__:': 'select_rates_regression', 'regressor:__choice__:': 'k_nearest_neighbors', 'data_processing:categorical_transformer:category_coalescer:minimum_fraction': 0.41842854801422075, 'data_processing:numerical_transformer:rescaling:quantile_transformer:n_quantiles': 317, 'data_processing:numerical_transformer:rescaling:quantile_transformer:output_distribution': 'uniform', 'feature_processor:select_rates_regression:alpha': 0.26207813302454075, 'feature_processor:select_rates_regression:mode': 'fdr', 'feature_processor:select_rates_regression:score_func': 'f_regression', 'regressor:k_nearest_neighbors:n_neighbors': 41, 'regressor:k_nearest_neighbors:p': 1, 'regressor:k_nearest_neighbors:weights': 'distance'}, dataset_properties={ 'task': 4, 'sparse': False, 'multioutput': False, 'target_type': 'regression', 'signed': False})</pre>

Σχήμα 5.38: Αναλυτικές πληροφορίες Συνδυαστικού Μοντέλου Wine Quality

## 5..2 Ανάκτηση Μοντέλου μέσω του Αποθηκευμένου Αρχείου

Μπορούμε επιπλέον να ανακτήσουμε το μοντέλο μας μέσω του αποθηκευμένου αρχείου μας χρησιμοποιώντας τον απαραίτητο κώδικα. Παρακάτω φαίνεται ένα παράδειγμα ανάκτησης του μοντέλου μέσω του pickle αρχείου που παράγεται κατα την δημιουργία και αποθήκευση του μοντέλου μηχανικής μάθησης. Συγκεκριμένα, θα ανακτήσουμε και θα επαναχρησιμοποιήσουμε το μοντέλο `iris` που δημιουργήσαμε κατά την διαδικασία που ακολουθήσαμε στην υποενότητα 2.2. Ακολουθεί ο κώδικας με τα απαραίτητα σχόλια για την κατάνοηση της λειτουργίας του. Θεωρούμε δεδομένο πως το αρχείο python κώδικα βρίσκεται στον ίδιο φάκελο με το μοντέλο που έχουμε παράξει με την βοήθεια του εργαλείου.

```

import pickle
from autosklearn.classification import AutoSklearnClassifier
import pandas as pd
import sklearn

# Ορίσμας path epithumhtou arxeiou
fileName = "./iris.csv"

# Anagnwsh arxeiou iris.csv
df = pd.read_csv(fileName, sep=", ;",
                  engine='python')

# Metavlth sthn opoia orizetai h sthlh target
tar_idx = 4
# Ορίσμας ths target sthlhs

target = df.iloc[:, (tar_idx)]

# Ορίσμας xarakthristikwn problepshs
predtictors = df.iloc[:, df.columns != df.columns[tar_idx]]

# Fortwsh tou montelou apo to pickle arxeio
model = pickle.load( open( "iris_1hr_5folds", "rb" ) )

# Pragmatopoihsh proulepsewn
predictions = model.predict(predtictors)

# Emfanhsh apodoshs me thn metrikh Accuracy
print (f"Accuracy: {sklearn.metrics.accuracy_score(target, predictions
    ↪ )}"))

# Accuracy: 0.9933333333333333

```

Μετά την εκτέλεση του κώδικα μπορούμε να δούμε στο Τερματικό τα αποτελέσματα μας, επιβεβαιώνοντας πως τα παραγόμενα μοντέλα του εργαλείου μας μπορούν να επαναχρησιμοποιηθούν ανά πάσα στιγμή.

## 6. Δημιουργία Μοντέλου Micro Μηχανικής Μάθησης

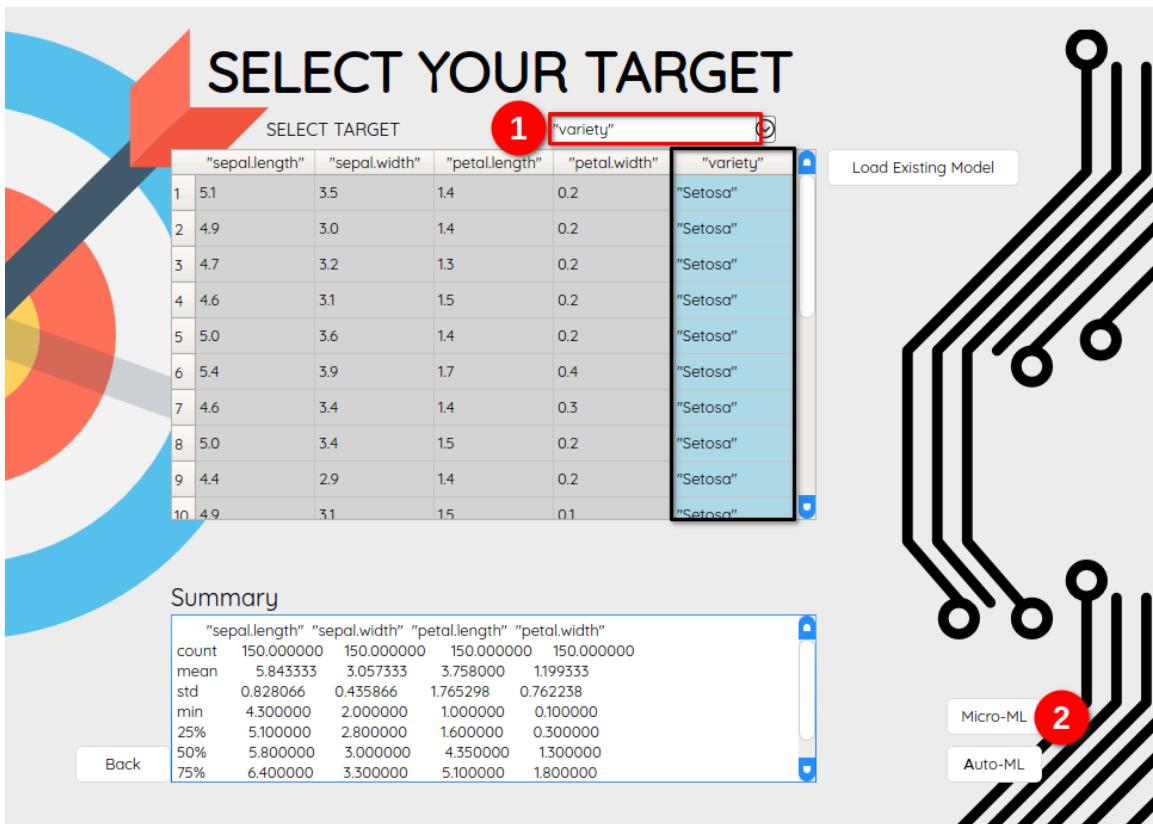
### 6..1 Σύνολο Δεδομένων Classification

Για την δημιουργία μοντέλου Micro Μηχανικής Μάθησης έγινε χρήση του μοντέλου *iris* όπως και στην υποενότητα [2.2](#).

### 6..2 Δημιουργία Classification Μοντέλου για Μικροελεγκτή

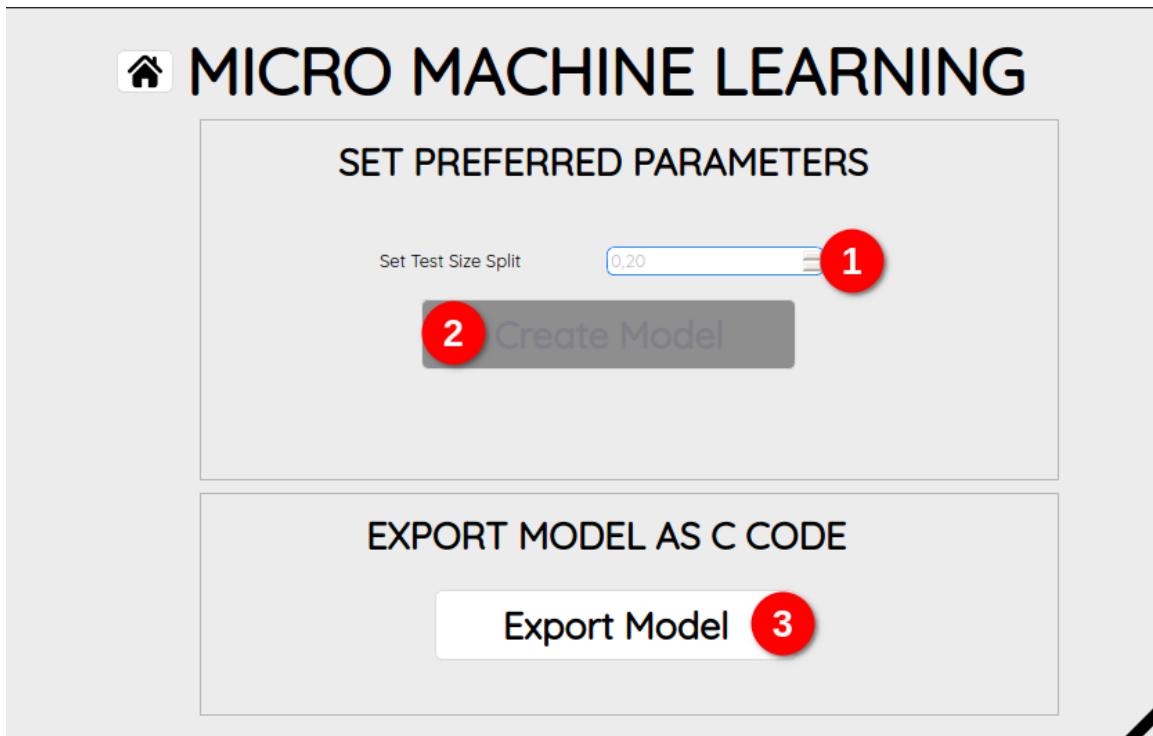
Παρακάτω περιγράφονται τα βήματα για την δημιουργία μοντέλου μηχανικής μάθησης για Classification προβλήματα με σκοπό την εξαγωγή του σε μιορφή υποστηριζόμενη από μικρο-ελεγκτές. Το παραγόμενο μοντέλο εφαρμόζεται στην συνέχεια στον μικρο-ελεγκτή UNO R3 ATmega328P CH340. Η διαδικασία εφαρμογής του μοντέλου στον μικρο-ελεγκτή περιγράφεται στον επόμενο οδηγό.

Αρχικά ακολουθούμε τα βήματα [1](#), [2](#) και [3](#) όπως περιγράφονται στην ενότητα [2.2](#). Στο βήμα [3](#), μετά την επιλογή της target μεταβλητής μας, θα επιλέξουμε Micro ML όπως φαίνεται στο σχήμα [5.39](#).

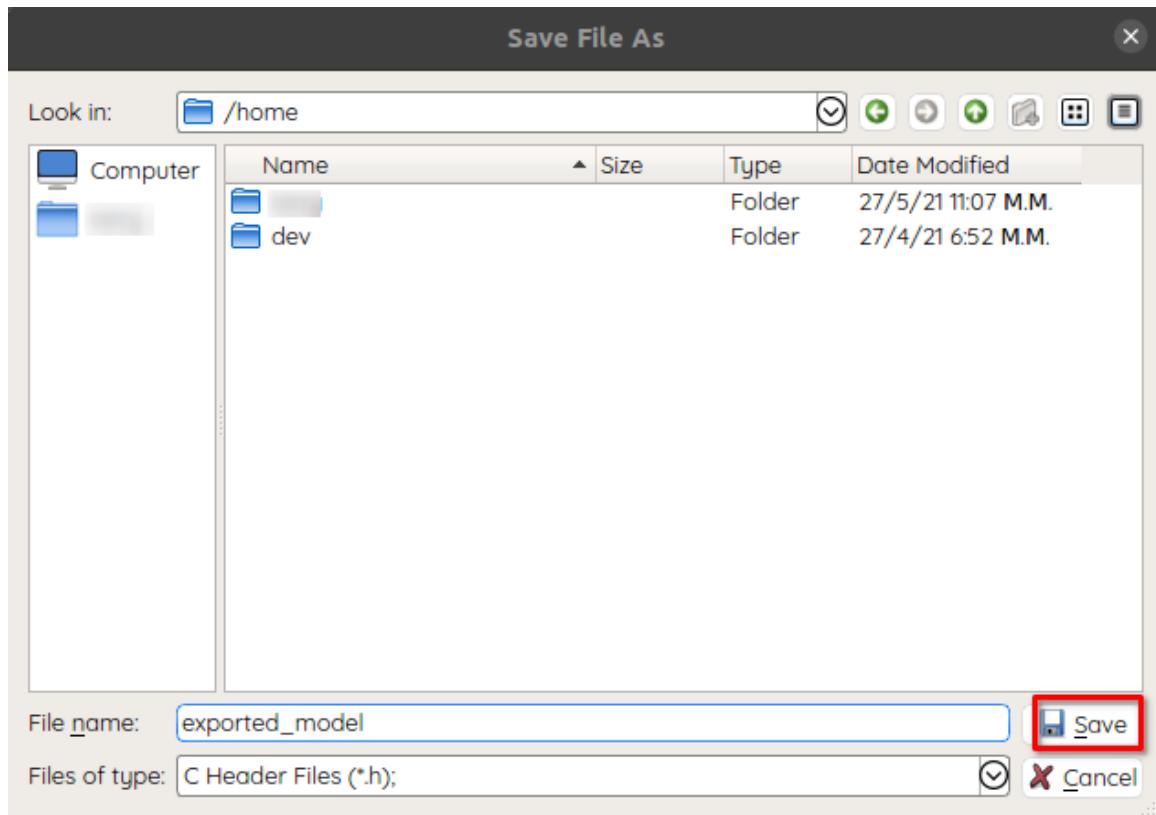


Σχήμα 5.39: Επιλογή MicroML

**Βήμα 4:** Στην οδόντ που ακολουθεί θέτουμε, αρχικά τις παραμέτρους που επιθυμούμε και επιλέγουμε Create Model. Περιμένουμε για την δημιουργία του μοντέλου ώσπου να γίνει ενεργή η επιλογή Export. Στην συνέχεια επιλέγουμε Export και στο αναδυόμενο παράθυρο επιλέγουμε την τοποθεσία που επιθυμούμε να αποθηκεύσουμε το αρχείο που εξάγεται. Οι παραπάνω ενέργειες απεικονίζονται στο σχήμα 5.40 και στο σχήμα 5.41.



Σχήμα 5.40: Δημιουργία και Εξαγωγή μοντέλου



Σχήμα 5.41: Αποθήκευση Μοντέλου MicroML

Συμπληρώνουμε το όνομα που επιθυμούμε και πατάμε Save. Στην τοποθεσία που επιλέξαμε θα βρίσκεται το αρχείο μας με κατάλογο .h το οποίο μπορούμε να χρησιμοποιήσουμε για την υλοποίηση μηχανικής μάθησης σε μικροελεγκτές. Η διαδικασία αυτή περιγράφεται παρακάτω.

### 6..3 Εφαρμογή Classification Μοντέλου στον Μικρο-ελεγκτή UNO R3 ATmega328P

Το παρακάτω παράδειγμα υλοποιήθηκε σε λειτουργικό σύστημα Windows 10, χρησιμοποιώντας την πλακέτα UNO R3 ATmega328P CH340, η οποία είναι συμβατή με Arduino.



Σχήμα 5.42: Μικροελεγκτής UNO R3 ATmega328P

Στην συνέχεια παρατίθενται βίντα για την εγκατάσταση και την χρήση του.

### Βήμα 1: Εγκατάσταση Drivers

Για την συγκεκριμένη πλακέτα ήταν απαραίτητη η εγκατάσταση drivers για τον CH340 ώστε να αναγνωριστεί η συσκευή επιτυχώς και να λειτουργήσει σωστά. Με μια απλή αναζήτηση στο διαδίκτυο μπορούμε να βρούμε τους κατάλληλους οδηγούς σε εκτελέσιμο αρχείο και να τους εγκαταστήσουμε στον υπολογιστή μας.

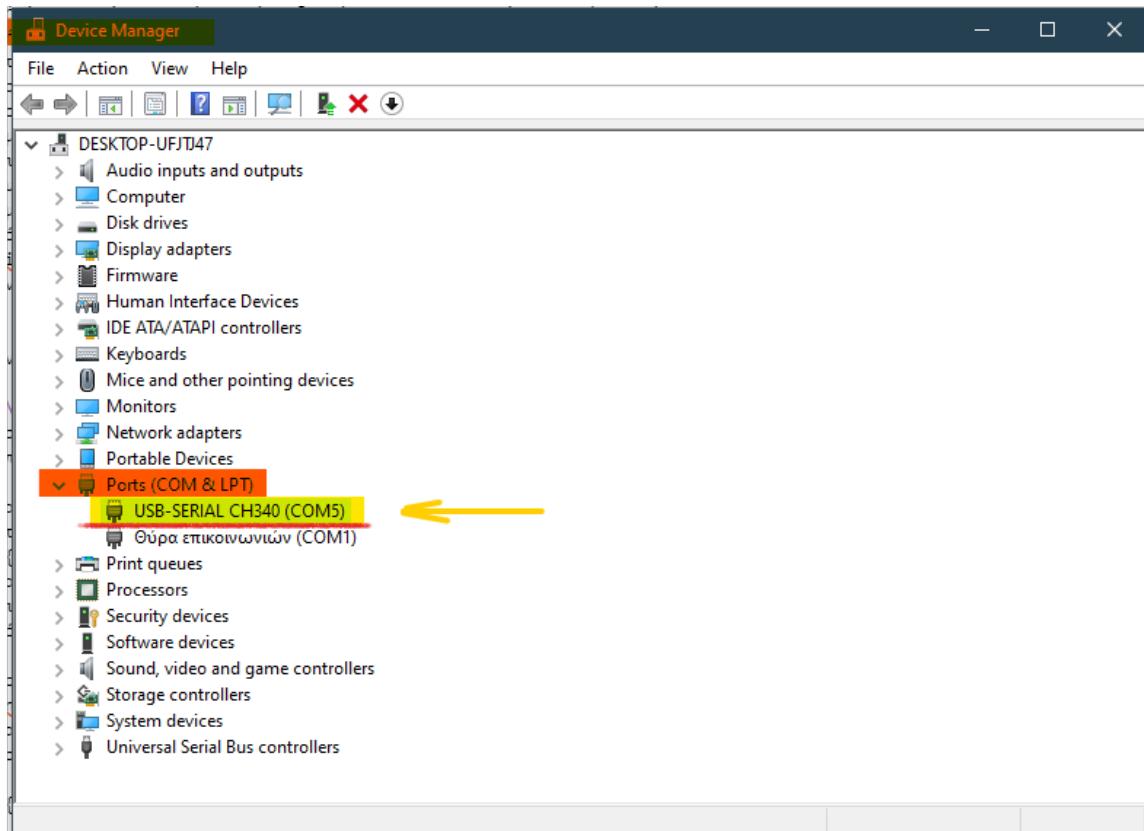
Συνδέουμε τον μικρο-ελεγκτή σε μία θύρα USB του υπολογιστή μας μέσω USB καλωδίου τύπου A/B.



Σχήμα 5.43: A/B καλώδιο

Στην περίπτωση που η εγκατάσταση πραγματοποιήθει σωστά, τότε η πλακέτα μας θα αναγνωριστεί από τα Windows ως συσκευή με όνομα USB-SERIAL CH340 (COM5). Μπορούμε να επιβεβαιώσουμε την συγκεκριμένη πληροφορία

εάν πλοηγηθούμε στο Device Manager των Windows, στην περιοχή Ports όπως φαίνεται στο στιγμότυπο.



Σχήμα 5.44: Device Manager - CH340

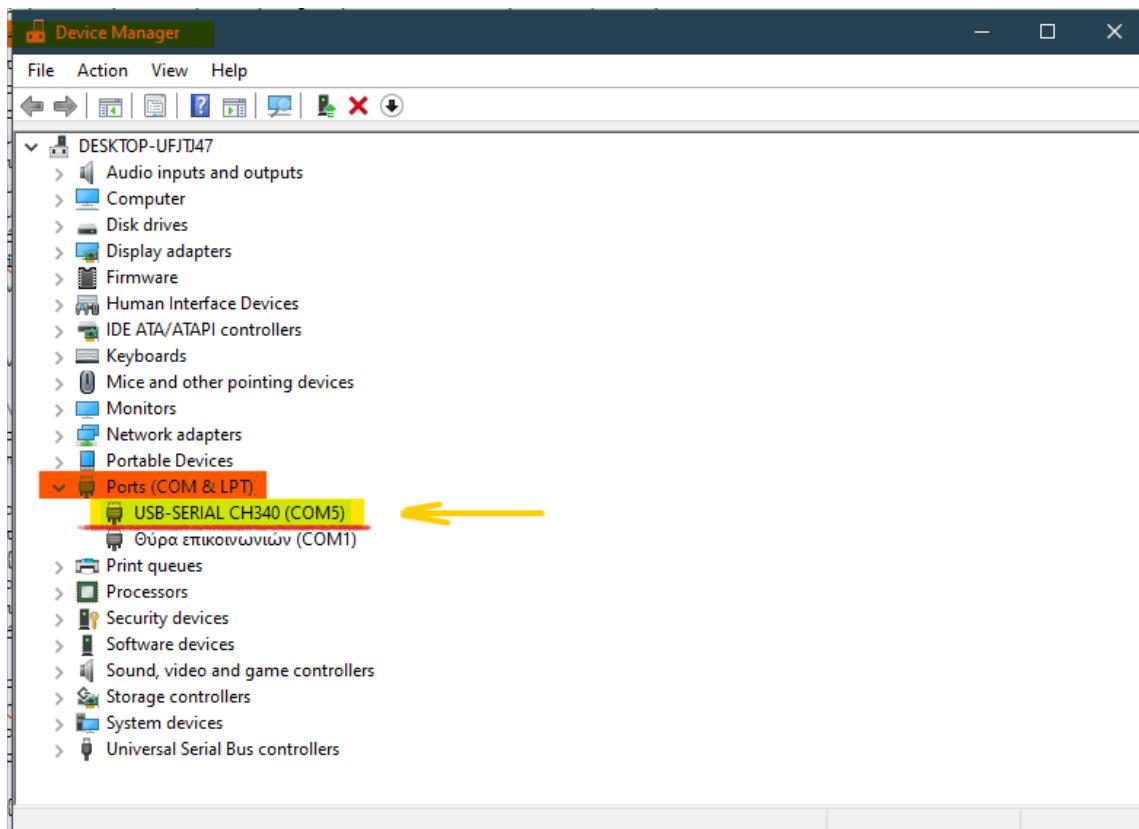
## Βήμα 2: Εγκατάσταση Λογισμικού

Για να προγραμματίσουμε την πλακέτα μας θα πρέπει να κατεβάσουμε και να εγκαταστήσουμε την εφαρμογή Arduino IDE μέσω του συνδέσμου: <https://www.microsoft.com/el-gr/p/arduino-ide/9nb1gggh4rsd8?ocid=badge&rtc=1#activetab=pivot:overviewtab> ή άμεσα μέσω του Microsoft Store. Μόλις ολοκληρωθεί η εγκατάσταση, ανοίγουμε το Arduino IDE και πραγματοποιούμε τις εξής ρυθμίσεις.

1. Επιλέγουμε, στην περίπτωση που δεν είναι επιλεγμένος, τον τύπο της πλακέτας μας: **Tools > Board > Arduino Uno**
2. Επιλέγουμε το κατάλληλο port για την πλακέτα μας: **Tools > Port > COM5**
3. Επιλέγουμε **Tools > Serial Monitor** και στο παράθυρο που προκύπτει ρυθμίζουμε το **baud rate** στο **115200** για τον συγκεκριμένο μικροελεγκτή.

### Βήμα 3: Αρχεία .ino και .h

Στο Arduino IDE επιλέγουμε File > New και δημιουργούμε ένα νέο αρχείο τύπου ino.



Σχήμα 5.45: Device Manager - CH340

Σε αυτό το σημείο βεβαιωνόμαστε ότι το .h αρχείο που έχει προκύψει από το Tool, βρίσκεται στον ίδιο φάκελο με το ino αρχείο μας ώστε να το συμπεριλάβουμε στον κώδικά μας. Το .ino αρχείο μας έχει την παρακάτω μορφή. Συμπεριλαμβάνουμε στην αρχή το αρχείο που έχει παραχθεί από το Tool με το σωστό όνομα. Στο συγκεκριμένο παράδειγμα πραγματοποιούνται τρεις προβλέψεις σύμφωνα με το σύνολο δεδομένων iris.

```
#include "iris.h"
Eloquent::ML::Port::SVM clf;
void setup() {
    Serial.begin(115200);
    delay(2000);

    Serial.println("Begin");
```

```

}

void loop() {
    Serial.println("Working...");

    float irisSample[4] = {5.0,3.3,1.4,0.2}; //setosa
    float irisSample1[4] = {7.0,3.2,4.7,1.4}; //versicolor
    float irisSample2[4] = {6.7,3.1,5.6,2.4}; //virginica

    Serial.print("Labels -> (0 = setosa, 1 = versicolor, 2 = virginica
    → ) \n");
    Serial.print("Prediction #1: ");
    Serial.println(clf.predict(irisSample));
    Serial.print("Prediction #2: ");
    Serial.println(clf.predict(irisSample1));
    Serial.print("Prediction #3: ");
    Serial.println(clf.predict(irisSample2));
    Serial.print("Exiting...");

    delay(1000);
    exit(0);
}

```

#### Βίμα 4 : Προγραμματισμός της Πλακέτας

Όταν ολοκληρώσουμε τα παραπάνω βήματα μπορούμε να αναρτήσουμε τον κώδικα στην πλακέτα μας μέσω του Upload κουμπιού από το περιβάλλον του Arduino IDE. Για να επιβεβαιώσουμε ότι το μοντέλο πραγματοποιεί προβλέψεις μπορούμε να ανοίξουμε το Serial Monitor και να δούμε την έξοδο στην οθόνη μας. Οι προβλέψεις που πραγματοποιούνται είναι πράγματι ακριβείς.

The screenshot shows the Arduino IDE interface. On the left, the code for `microml_1` is displayed, including the header `#include "iris.h"` and the main loop which prints out the results of an SVM prediction. On the right, the Serial Monitor window titled "COM4" shows the printed text.

```

microml_1 | Arduino 1.8.13
File Edit Sketch Tools Help
microml_1 iris.h
#include "iris.h"

Eloquent::ML::Port::SVM clf;

void setup() {
    Serial.begin(115200);
    delay(2000);

    Serial.println("Begin");
}

void loop() {
    Serial.println("Working...");

    float irisSample[4] = {5.0,3.3,1.4,0.2}//setosa
    float irisSample1[4] = {7.0,3.2,4.7,1.4}//versicolor
    float irisSample2[4] = {6.7,3.1,5.6,2.4}//virginica

    Serial.print("Labels -> (0 = setosa, 1 = versicolor, 2 = virginica) \n");
    Serial.print("Prediction #1: ");
    Serial.println(clf.predict(irisSample));
    Serial.print("Prediction #2: ");
    Serial.println(clf.predict(irisSample1));
    Serial.print("Prediction #3: ");
    Serial.println(clf.predict(irisSample2));

    Serial.print("Exiting...");

    delay(1000);
    exit(0);
}

```

Serial Monitor Output:

```

Begin
Working...
Labels -> (0 = setosa, 1 = versicolor, 2 = virginica)
Prediction #1: 0
Prediction #2: 1
Prediction #3: 2
Exiting...

```

Σχήμα 5.46: Arduino IDE και Serial Monitor

The screenshot shows the Serial Monitor window titled "COM5". A yellow box highlights the printed text, and a red arrow points downwards from the end of the highlighted text towards the bottom of the window.

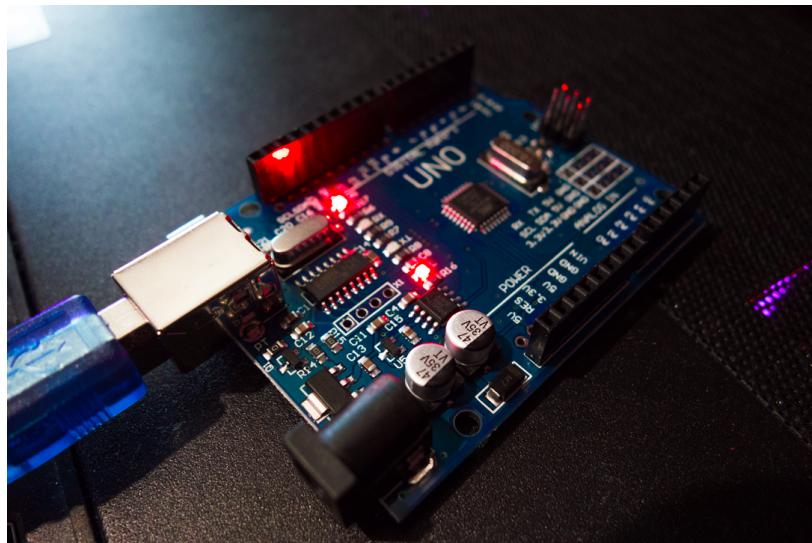
Serial Monitor Output:

```

Begin
Working...
Labels -> (0 = setosa, 1 = versicolor, 2 = virginica)
Prediction #1: 0
Prediction #2: 1
Prediction #3: 2
Exiting...

```

Σχήμα 5.47: Serial Monitor



Σχήμα 5.48: Μικροελεγκτής εν λειτουργία

## 7. Μελλοντικές Επεκτάσεις

Στην παρόύσα διπλωματική εργασία σχεδιάστηκε και αναπτύχθηκε εργαλείο Automated Machine Learning για IoT εφαρμογές με δυνατότητες επεξεργασίας δεδομένων, παραγωγής και αποθήκευσης μοντέλων. Οι δυνατότητες του εργαλείου περιγράφονται στην ενότητα 1. καθώς και στην υπο-ενότητα "Συμπεράσματα" που ακολουθεί. Η βιβλιογραφική έρευνα αποτελεί σημαντική συνεισφορά για νέους ερευνητές όσον αφορά την Μηχανική Μάθηση και την Αυτοματοποιημένη Μηχανική Μάθηση. Το εργαλείο είναι σχεδιασμένο και υλοποιημένο με τέτοιον τρόπο ώστε να προσφέρει την δυνατότητα επέκτασής του με σκοπό να υλοποιεί ακόμα περισσότερες λειτουργίες. Για την επέκταση της εφαρμογής, μπορεί να τροποποιηθεί ο υπάρχων κώδικας και να προστεθούν επιπλέον λειτουργίες, είτε μέσω του σχεδιασμού και της ανάπτυξης νέων οδονών, είτε με την επεξεργασία των υπαρχουσών οδονών και της λειτουργικότητας τους στο back-end.

Συγκεκριμένα, περαιτέρω έρευνα και ανάπτυξη μπορεί να πραγματοποιηθεί στα παρακάτω σημεία.

- ▶ Προσθήκη δυνατότητας δημιουργίας μοντέλων μέσω νευρωνικών δικτύων.
- ▶ Προσθήκη δυνατότητας εκπαίδευσης μοντέλων μέσω βιβλιοθηκών ή εργαλείων εκτός της auto-sklearn βιβλιοθήκης.
- ▶ Περαιτέρω έρευνα και ανάπτυξη όσον αφορά στο Automated Machine

Learning με σκοπό την επέκταση της παραγωγής μοντέλων για μικροεπεξεργαστές και edge devices (για παράδειγμα Arduino).

- ▶ Δυνατότητα διαχείρισης συνόλων δεδομένων πολλαπλών χρονολογικών σειρών.
- ▶ Έρευνα και ανάπτυξη όσον αφορά την αποδοτική αποθήκευση των μοντέλων που παράγονται από την εφαρμογή.

## 8. Συμπεράσματα

Παρατηρώντας τα αποτελέσματα και την αξιολόγηση των λειτουργιών της αναπτυγμένης εφαρμογής, μπορούμε να εξάγουμε το συμπέρασμα πως οι αρχικοί στόχοι της παρούσας διπλωματικής εργασίας επιτεύχθηκαν. Η εφαρμογή δεν παρουσιάζει προβλήματα κατά την εκτέλεση και πετυχαίνει αποδοτικά τους αρχικούς σκοπούς της, αποτελώντας ένα φιλικό προς το χρήστη εργαλείο Αυτοματοποιημένης Μηχανικής Μάθησης για την δημιουργία, αποθήκευση και εξαγωγή μοντέλων με δυνατότητα εφαρμογής σε μικρο-ελεγκτές, την πραγματοποίηση προβλέψεων σε σύνολα δεδομένων, την εξαγωγή χαρακτηριστικών, την επεξεργασία χρονολογικών σειρών και την μετατροπή προβλημάτων καπνογορίας χρονολογικών σειρών σε κλασσικά προβλήματα μηχανικής μάθησης.

Το εργαλείο είναι απλό, κατανοητό, εύχρονο και ενθαρρύνει απλούς χρήστες να δημιουργήσουν τα δικά τους μοντέλα μηχανικής μάθησης, απαλλάσσοντάς τους από περίπλοκες και χρονοβόρες διαδικασίες που απαιτούν εξειδικευμένες γνώσεις.

Συμπερασματικά, μπορούμε να πούμε πως το εργαλείο που υλοποιήθηκε πετυχαίνει τον αρχικό κύριο στόχο του, που ήταν το να φέρει περισσότερους ανθρώπους πιο κοντά στον τομέα της Μηχανικής Μάθησης, μέσω της απλότητας, της ευχρονστίας και της αποδοτικότητας του. Επιπλέον, η βιβλιογραφική έρευνα θα αποτελέσει σημαντική συνεισφορά για νέους ερευνητές, στο πεδίο της Αυτοματοποιημένης Μηχανικής Μάθησης.

# Βιβλιογραφία

- [1] Machine learning crash course - google - roc auc. <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc>.
- [2] Sklearn metrics. [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average\\_precision\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html).
- [3] tsfresh 0.17.1 documentation. <https://tsfresh.readthedocs.io/en/latest/index.html>.
- [4] Scikit-learn, <https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/>, Jan 2015.
- [5] Beginner's guide to reinforcement learning and its implementation in python, Jan 2017.
- [6] Linear classifier. [https://en.wikipedia.org/w/index.php?title=Linear\\_classifier&oldid=979744680](https://en.wikipedia.org/w/index.php?title=Linear_classifier&oldid=979744680), Sep 2020.
- [7] Adaboost. <https://en.wikipedia.org/w/index.php?title=AdaBoost&oldid=1021676776>, May 2021.
- [8] Linear discriminant analysis. [https://en.wikipedia.org/w/index.php?title=Linear\\_discriminant\\_analysis&oldid=1021740456](https://en.wikipedia.org/w/index.php?title=Linear_discriminant_analysis&oldid=1021740456), May 2021.
- [9] Overfitting. <https://en.wikipedia.org/w/index.php?title=Overfitting&oldid=1016721642>, Apr 2021.
- [10] Training, validation, and test sets. [https://en.wikipedia.org/w/index.php?title=Training,\\_validation,\\_and\\_test\\_sets&oldid=1021890103](https://en.wikipedia.org/w/index.php?title=Training,_validation,_and_test_sets&oldid=1021890103), May 2021.
- [11] Eijaz Allibhai. Holdout vs. cross-validation in machine learning. <https://medium.com/@eijaz/holdout-vs-cross-validation-in-machine-learning-7637112d3f8f>, Oct 2018.

- [12] Ankit Bisht. ML | classification vs regression 2019. <https://www.geeksforgeeks.org/ml-classification-vs-regression>, Jan 2019.
- [13] Jan Bodnar. Events and signals in pyqt5. <https://zetcode.com/gui/pyqt5/eventssignals/>.
- [14] Vivien Cabannes, Alessandro Rudi, and Francis Bach. Fast rates in structured prediction, 2021.
- [15] cartacioS. What is automated ml? automl - azure machine learning. <https://docs.microsoft.com/en-us/azure/machine-learning/concept-automated-ml>.
- [16] Paulo Cortez, Antônio Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009. Smart Business Networks: Concepts and Empirical Evidence.
- [17] cwells. Event-driven programming. <https://www.technologyuk.net/computing/software-development/software-design/event-driven-programming.shtml>.
- [18] Ayon Dey. Machine learning algorithms : A review. 2016.
- [19] Dheeru Dua and Casey Graff. UCI machine learning repository. <http://archive.ics.uci.edu/ml>, 2017.
- [20] eloquentarduino. *Introducing MicroML*. May 2021.
- [21] World Leaders in Research-Based User Experience. Wizards: Definition and design recommendations. <https://www.nngroup.com/articles/wizards/>.
- [22] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, and Frank Hutter. *Auto-sklearn: Efficient and Robust Automated Machine Learning*, page 113–134. The Springer Series on Challenges in Machine Learning. Springer International Publishing, 2019.
- [23] Chelsea Finn. *Learning to Learn with Gradients*. PhD thesis, EECS Department, University of California, Berkeley, Aug 2018.
- [24] Prashant Gupta. Decision trees in machine learning. <https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052>, Nov 2017.

- [25] Onel Harrison. Machine learning basics with the k-nearest neighbors algorithm. <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>, Jul 2019.
- [26] Xin He, Kaiyong Zhao, and Xiaowen Chu. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, Jan 2021.
- [27] Bernard J. Jansen. The graphical user interface. *ACM SIGCHI Bulletin*, 30(2):22–26, Apr 1998.
- [28] Dr Robert Kübler. Learning by implementing: Gaussian naive bayes. <https://towardsdatascience.com/learning-by-implementing-gaussian-naive-bayes-3f0e3d2c01b2>, Jan 2021.
- [29] Stephen Makonin. ODDs: Occupancy Detection Dataset. <https://doi.org/10.7910/DVN/2K9FFE>, 2015.
- [30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [31] RedHat. What is event-driven architecture? <https://www.redhat.com/en/topics/integration/what-is-event-driven-architecture>.
- [32] Stuart J. Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Pearson series in artificial intelligence. Pearson, fourth edition edition, 2021.
- [33] Annina Simon, Mahima Deo, Venkatesan Selvam, and Ramesh Babu. An overview of machine learning and its applications. *International Journal of Electrical Sciences Engineering*, Volume:22–24, 01 2016.
- [34] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms, 2012.
- [35] Kevin Swersky, Jasper Snoek, and Ryan Prescott Adams. Freeze-thaw bayesian optimization, 2014.
- [36] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.
- [37] Joaquin Vanschoren. *Meta-Learning*, page 35–61. The Springer Series on Challenges in Machine Learning. Springer International Publishing, 2019.

- [38] Quanming Yao, Mengshuo Wang, Yuqiang Chen, Wenyuan Dai, Yu-Feng Li, Wei-Wei Tu, Qiang Yang, and Yang Yu. Taking Human out of Learning Applications: A Survey on Automated Machine Learning. *arXiv:1810.13306 [cs, stat]*, December 2019. arXiv: 1810.13306.
- [39] Tony Yiu. Understanding random forest. <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>, Aug 2019.
- [40] Χρήστος Π. Κωτσαλένη. *ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ “ΜΑΘΗΜΑΤΙΚΑ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΤΩΝ ΑΠΟΦΑΣΕΩΝ”*. PhD thesis, ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ, May 2017.