Πανεπιστήμιο Πειραιώς
Τμήμα Ψηφιακών Συστημάτων
Π.Μ.Σ Πληροφοριακά Συστήματα & Υπηρεσίες

Ειδίκευση
Μεγάλα Δεδομένα και Αναλυτική

Διπλωματική Εργασία

Automated Supervised Machine Learning with sampling techniques

Αυτόματη Εποπτευόμενη Μηχανική Μάθηση με τεχνικές δειγματοληψίας

Σταύρος Κουρέας
Φεβρουάριος 2022

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου Χρήστο Δουλκερίδη, καθώς κατά τη διεκπεραίωση της διπλωματικής μου εργασίας και τη πορεία μου στο μεταπτυχιακό πρόγραμμα σπουδών αποκόμισα γνώσεις, συμβουλές οι οποίες με βοήθησαν στην πραγματοποίηση της παρούσας έρευνας.
Αναμφισβήτητα, θα ήθελα να ευχαριστήσω τόσο τη συμβολή του Ιωάννη Πουλάκη για τις δημιουργικές συζητήσεις επάνω στην έρευνα μου, όσο και τον Νικόλαο Κουτρουμάνη για την αέναη ανταλλαγή απόψεων κατά την διάρκεια του μεταπτυχιακού προγράμματος.
Με αυτόν τον τρόπο, δημιουργήθηκαν ιδέες και σκέψεις, οι οποίες έδρασαν ως πυλώνες για τη κάτωθι εργασία, που στόχος της είναι η εξέλιξη της μηχανικής μάθησης για την αντιμετώπιση μελλοντικών προβλημάτων.
Τέλος, θα ήθελα να ευχαριστήσω τον στενό μου κύκλο ο οποίος με στήριξε για να φέρω εις πέρας το συγκεκριμένο μεταπτυχιακό πρόγραμμα.

Περιεχόμενα

Εικόνες

Abstract

Data analysis is a sector of modern science that deals with the management and interpretation of usable information, which is now growing rapidly. In this thesis, machine learning applications will be researched in data sets of various sizes, technologies mentioned at the beginning and on the one hand the machine learning improvement technique the devices seem to meet specific specifications.

Nowadays "armed" with a multitude of algorithms and hyper-parameters we can achieve amazing results but choosing the right combinations is a difficult process. By processing large data sets daily, the demands on processing power and time increase. Most disciplines require highly accurate predictions, which requires a great deal of research in each data set.

This thesis aims to propose a new technique with sampling procedures, which can bring satisfactory results in less time with less processing power. At the same time, it "builds" a methodology for analyzing big data and dealing with general problems such as missing values, alphanumeric values and others such as unbalanced data sets.

This technique works with the method of sampling in rows and columns, is evaluated through an experimental process where several results are collected from different data sets and compared without using it. More specifically, 15 data sets were used for binary classification, 15 for multi classification and 5 for regression. All data sets are known datasets in the field of machine learning.

The results of the experimental procedure indicated that 10% is sufficient for sampling in rows and 80% is sufficient for sampling in columns based on correlation. The result seems to be satisfactory since the same selection of algorithms with the use of the sample against complete at 80%, while in the case that the selection algorithm is not the same, there is a probability that exceeds 70% on selecting an algorithm that is the next better one. This practically means that if in a smaller data set the decision to use an algorithm was made, then this algorithm is quite likely to work better in the whole data set.

Specifically, this technique is developed in python language in the form of a library, which consists of specific organized sub-procedures. Each sub-process handles specific decisions during the pre-processing stage, such as sampling management in rows, sampling management in columns, dealing with missing values, normalization but also in the modeling stage such as algorithm selection and hyperparameter optimization.

However, this library has been published in the PiPy repository under the name "**AutoMLWrapper**" (since it is a set of subsystems of special methods) and is accompanied by a relevant notebook sample. https://pypi.org/project/automlwrapper/ So distribution and execution can be done easily and quickly in a simple python environment by installing the library using pip install, so its use is direct to the end user.

Περίληψη

Η ανάλυση δεδομένων είναι ένας κλάδος της σύγχρονης επιστήμης, ο οποίος ασχολείται με τη διαχείριση και την ερμηνεία αξιοποιήσιμων πληροφοριών, οι οποίες πλέον αυξάνονται με ραγδαίους ρυθμούς. Σε αυτό το έγγραφο για τους σκοπούς της διπλωματικής εργασίας, θα ερευνηθούν αφενός εφαρμογές μηχανικής μάθησης σε σύνολα δεδομένων διαφόρων μεγεθών, τεχνολογίες που χρησιμοποιούνται κατά κόρον και αφενός οι τεχνικές βελτίωσης απόδοσης της μηχανικής μάθησης οι οποίες φαίνεται να ικανοποιούν συγκεκριμένες απαιτήσεις.

Στην σημερινή εποχή «οπλισμένοι» με ένα πλήθος αλγορίθμων και υπερ-παραμέτρων μπορούμε να επιτύχουμε καταπληκτικά αποτελέσματα, όμως η κατάλληλη επιλογή τους είναι μια δύσκολη διαδικασία. Επεξεργαζόμενοι καθημερινά μεγάλα σύνολα δεδομένων οι απαιτήσεις σε επεξεργαστική ισχύ και σε χρόνο αυξάνονται. Οι περισσότεροι επιστημονικοί κλάδοι απαιτούν προβλέψεις με υψηλή ακρίβεια, κάτι το οποίο χρειάζεται αρκετή έρευνα για κάθε σύνολο δεδομένων προκειμένου να επιτευχθεί.

Η παρούσα διπλωματική εργασία προτείνει μια νέα τεχνική με δειγματοληπτικές διαδικασίες, η οποία μπορεί να επιφέρει ικανοποιητικά αποτελέσματα σε λιγότερο χρόνο με μικρότερη επεξεργαστική ισχύ. Παράλληλα «χτίζει» μια μεθοδολογία για την ανάλυση μεγάλων δεδομένων και αντιμετώπισης γενικών προβλημάτων όπως οι ελλιπείς τιμές, οι αλφαριθμητικές τιμές αλλά και άλλα όπως τα ανισόρροπα σύνολα δεδομένων.

Η τεχνική αυτή, λειτουργεί με την μέθοδο της δειγματοληψίας σε γραμμές αλλά και στις στήλες, αξιολογείται μέσω μιας πειραματικής διαδικασίας όπου συγκεντρώνονται αρκετά αποτελέσματα από διάφορα σύνολα δεδομένων και συγκρίνεται χωρίς την χρήση αυτής. Πιο συγκεκριμένα χρησιμοποιήθηκαν 15 σύνολα δεδομένων για binary classification, 15 για multi classification και 5 για regression. Όλα τα σύνολα δεδομένων είναι γνωστά σύνολα στον τομέα της μηχανικής μάθησης.

Τα αποτελέσματα της πειραματικής διαδικασίας υπέδειξαν πως ένα 10% είναι αρκετό για δειγματοληψία στις γραμμές και ένα 80% είναι αρκετό για δειγματοληψία στις στήλες με βάση την συσχέτιση. Το αποτέλεσμα φαίνεται να είναι ικανοποιητικό αφού η ίδια επιλογή των αλγορίθμων με χρήση του sample έναντι του complete φτάνει στο 80%, ενώ στην περίπτωση που ο αλγόριθμος επιλογής δεν είναι ο ίδιος υπάρχει πιθανότητα που ξεπερνά το 70% να είναι ο αμέσως καλύτερος. Αυτό πρακτικά σημαίνει πως αν σε ένα μικρότερο σύνολο δεδομένων ληφθεί η απόφαση για την χρήση ενός αλγορίθμου τότε αυτός ο αλγόριθμος είναι αρκετά πιθανό να λειτουργήσει καλύτερα και σε όλο το σύνολο δεδομένων.

Συγκεκριμένα η τεχνική αυτή έχει υλοποιηθεί σε γλώσσα python υπό μορφή βιβλιοθήκης, η οποία αποτελείται από αρκετές υπο-διαδικασίες με συγκεκριμένη σειρά και οργάνωση. Κάθε υπο-διαδικασία χειρίζεται συγκεκριμένες αποφάσεις κατά το στάδιο της προ-επεξεργασίας, όπως την διαχείριση δειγματοληψίας στις γραμμές, την διαχείριση δειγματοληψίας στις στήλες, την αντιμετώπιση ελλιπών τιμών, την κανονικοποίηση αλλά και στο στάδιο της μοντελοποίησης όπως την επιλογή αλγορίθμου και την βελτιστοποίηση υπερ-παραμέτρων.

Ακόμα, αυτή η βιβλιοθήκη έχει δημοσιοποιηθεί στο repository PiPy με το όνομα "**AutoMLWrapper**" (αφού πρόκειται για ένα σύνολο υπο-διαδικασιών ειδικότερων μεθόδων) και συνοδεύεται από σχετικό notebook sample. https://pypi.org/project/automlwrapper/ Έτσι η διανομή και εκτέλεση μπορεί να γίνει εύκολα και γρήγορα σε ένα οποιοδήποτε περιβάλλον python απλά εγκαθιστώντας την βιβλιοθήκη με την χρήση του pip install ώστε η χρήση της να είναι άμεση από τον τελικό χρήστη.

1. Εισαγωγή

Στην σημερινή εποχή υπάρχει ένας καταιγισμός από δεδομένα διαφορετικής προέλευσης, διαφορετικής μορφής, τα οποία διανέμονται σε μεγάλο ρυθμό με αποτέλεσμα την δύσκολη ανάλυση και την εξαγωγή συμπερασμάτων αλλά ακόμη πιο δύσκολη την ανάλυση τους σε πραγματικό χρόνο. Η αλματώδης ανάπτυξη των υπολογιστικών συστημάτων (τόσο σταθερών όσο και κινητών) σε συνάρτηση με την ολοένα και μεγαλύτερη διείσδυση των ασύρματων και των ενσύρματων δικτύων έχουν ως συνέπεια την δημιουργία πολύ μεγάλων όγκων δεδομένων σε καθημερινή βάση. Η αποτελεσματική ανάλυση των δεδομένων μπορεί να προσφέρει ουσιαστικές λύσεις και να βοηθήσει στη λήψη αποφάσεων σε διαφόρους κλάδους (Thomas Dietterich, 2017).

Σε αυτήν την εποχή της τεχνητής νοημοσύνης, η μηχανική μάθηση είναι ένα μεγάλο θέμα. Η υπολογιστική όραση (computer vision) και η προβλεπτική αναλυτική (predictive analytics) ανοίγουν νέους δρόμους που κανείς δεν μπορούσε να προβλέψει. Παρατηρείται πως η μηχανική μάθηση χρησιμοποιείται ολοένα και περισσότερο στην καθημερινή μας ζωή. Για παράδειγμα η αναγνώριση προσώπου σε smartphone, το λογισμικό μετάφρασης γλώσσας, τα αυτο-οδηγούμενα αυτοκίνητα αλλά και σε συστήματα προτάσεων, σε διαδικασίες ανίχνευσης ανωμαλίας και πολλά ακόμη. Κατά το πέρας του χρόνου δημιουργήθηκαν πολλές τεχνικές ανάλυσης, εκπαίδευσης και πρόβλεψης με διαφορετικά χαρακτηριστικά, πλεονεκτήματα αλλά και μειονεκτήματα. Μερικά από τα χαρακτηριστικά αυτά είναι ο χρόνος εκπαίδευσης, το ποσοστό επιτυχίας εκπαίδευσης, το ποσοστό επιτυχίας πρόβλεψης και πολλές ακόμη μετρικές αξιολόγησης.

Ανάλογα το πρόβλημα, υπάρχουν διαφορετικά πεδία ανάλυσης, όπως της κατηγοριοποίησης, της παλινδρόμησης, ομαδοποίησης, της εξαγωγής συσχετίσεων αλλά και της πρόβλεψης χρόνο-σειρών, όπου το καθένα χρησιμοποιείται με συγκεκριμένη μεθοδολογία. Τα τελευταία χρόνια σε πολλά από τα προβλήματα της ανάλυσης δεδομένων που ανακύπτουν, η απάντησή τους εξαρτάται κατά κύριο λόγο από την ενυπάρχουσα ικανότητα διαχείρισης του μεγάλου όγκου των δεδομένων, την επιλογή μεθόδων επεξεργασίας, ανάλυσης αλλά και την χρήση κατάλληλων παραμέτρων. Για τον λόγο αυτό ανάλογα το πεδίο ανάλυσης υπάρχει ποικιλία αλγορίθμων που επιτυγχάνουν τον ίδιο στόχο αλλά με διαφορετικό ποσοστό επιτυχίας, χρόνου εκπαίδευσης και πολλά περισσότερα τα οποία πρέπει να ληφθούν υπόψιν. Ωστόσο, για κάθε αλγόριθμο υπάρχει ένα διαφορετικό σύνολο παραμέτρων οι οποίες αποκαλούνται «υπερ-παράμετροι» και οι οποίες διαχειρίζονται την συμπεριφορά του εκάστοτε αλγορίθμου.

Η χρήση της μηχανικής μάθησης δεν είναι εύκολη καθώς απαιτεί έρευνα και πειραματική διαδικασία προκειμένου να επιλεχθούν κατάλληλοι τρόποι προ-επεξεργασίας αλλά και επιλογή αλγορίθμων και παραμέτρων. Για αυτόν το λόγο κρίνεται απαραίτητη η χρήση μιας μεθοδολογίας αυτοματοποιημένης μηχανικής μάθησης η οποία αφενός θα μπορεί να προ-επεξεργαστεί τα δεδομένα κατάλληλα αντιμετωπίζοντας θέματα όπως κενές, ή κατηγορικές τιμές και αφετέρου θα επιτρέπει την αυτοματοποιημένη επιλογή αλγορίθμου (από ένα σύνολο αλγορίθμων) και την παραμετροποίησή του (από ένα εύρος πιθανών παραμέτρων) με χρήση ενός δείγματος μόνο των δεδομένων. Η αποτίμηση δείχνει ότι χρησιμοποιώντας ένα μικρότερο σύνολο δεδομένων μπορούν να ληφθούν αποφάσεις που αντιπροσωπεύουν όλο το σύνολο δεδομένων. Επομένως, η προτεινόμενη μέθοδος είναι εφαρμόσιμη σε μεγάλα σύνολα δεδομένων, μειώνοντας δραστικά το χρόνο επιλογής του καταλληλότερου αλγόριθμου. Σήμερα υπάρχουν λιγοστές αυτοματοποιημένες λύσεις που όμως δίνουν ικανοποιητικά αποτελέσματα. Κάποιες στοχεύουν περισσότερο στο κομμάτι της προ-επεξεργασίας (προετοιμασία δεδομένων) και κάποιες άλλες στο κομμάτι της μοντελοποίησης (επιλογή αλγορίθμου και υπερ-παραμέτρων).

2. Συναφείς Ερευνητικές Εργασίες

Τα τελευταία χρόνια έχουν γίνει πολλές έρευνες για τον τρόπο προ-επεξεργασίας των δεδομένων, τον τρόπο επιλογής κατάλληλων αλγορίθμων αλλά και για τον τρόπο επιλογής υπερ-παραμέτρων. Συγκεκριμένα, ενδιαφέρον παρουσιάζουν οι παρακάτω έρευνες οι οποίες είναι και μέρος της συλλογής «The Springer Series on Challenges in Machine Learning» στην οποία οι Frank Hutter, Lars Kotthoff και Joaquin Vanschoren εξηγούν υφιστάμενες έρευνες AutoML όπως το πρόβλημα βελτιστοποίησης των υπερ-παραμέτρων, τεχνικές learn to learn για συνεχή μάθηση, τεχνικές NAS για την σχεδίαση νευρωνικών δικτύων, το γνωστό εργαλείο Auto-Weka, τις γνωστές βιβλιοθήκες Hyperopt-sklearn και Auto-sklearn αλλά και την Auto-net όπως και το σύστημα TPOT. Τέλος, γίνεται μία επισκόπηση στις προκλήσεις του AutoML (Frank Hutter, et al., 2019).

Μια ενδιαφέρουσα έρευνα από τους Lars Kotthoff, Chris Thornton, Holger H. Hoos, Frank Hutter και Kevin Leyton-Brown με τίτλο «Auto-WEKA: Automatic Model Selection and Hyperparameter Optimization in WEKA» στην οποία περιγράφουν την βιβλιοθήκη Auto-WEKA η οποία λειτουργεί με επιλογή χαρακτηριστικών, στην συνέχεια εφαρμόζει μια μεθοδολογία CASH (Combined Algorithm Selection and Hyperparameter Optimization) όπου η επιλογή αλγορίθμου και υπερ-παραμέτρων γίνεται ταυτόχρονα και την τεχνική SMAC (Sequential Model-Based Algorithm Configuration) για την αναζήτηση υπερ-παραμέτρων η οποία συνδυάζει Bayesian Optimization με Random Forests. Ακόμα η συγκεκριμένη τεχνική εφαρμόζει αλγορίθμους ανάλογα με το σύνολο δεδομένων όπως για παράδειγμα εφαρμόζει αλγορίθμους που μπορούν να χειριστούν ελλιπείς τιμές (Lars Kotthoff, και συν., 2019).

Σε άλλη έρευνα αυτόματης μηχανικής μάθησης και επιλογής υπερ-παραμέτρων, οι Brent Komer, James Bergstra και Chris Eliasmith στο Paper «Hyperopt-Sklearn» περιγράφουν την βιβλιοθήκη Hypetopt SKLearn (για την προ-επεξεργασία) η οποία χρησιμοποιεί την Hyperopt (για την αναζήτηση υπερ-παραμέτρων) η οποία έχει την δυνατότητα να αναζητήσει και παράλληλα χρησιμοποιώντας υπολογιστικές μονάδες της MongoDB ή Spark σε ένα κατανεμημένο σύστημα ελαχιστοποιώντας μια συνάρτηση κόστους. Η βιβλιοθήκη δέχεται και αυτή pipelines με τις απαραίτητες ενέργειες προ-επεξεργασίας και μοντέλων προς αναζήτηση (Brent Komer, και συν., 2019).

Αντίθετα στο πρόβλημα CASH , όπου αλγόριθμος και υπερ-παράμετροι αναζητούνται μαζί, οι Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, Frank Hutter στο Paper «Efficient and Robust Automated Machine Learning» περιγράφουν μια ακόμη βιβλιοθήκη, την AutoSKLearn, η οποία είναι ενισχυμένη με ένα πρώτο Meta-Learning επίπεδο το οποίο επιλέγει τον αλγόριθμο μηχανικής μάθησης χρησιμοποιώντας την γνώση από προηγούμενα σύνολα δεδομένων που έχουν δοκιμαστεί, το Bayesian Optimization όπου γίνεται η επιλογή υπερ-παραμέτρων με επαναληπτικό τρόπο βάση της πιθανότητας βελτίωσης και τέλος ένα automated ensemble construction επίπεδο με το οποίο φαίνεται να είναι αποδοτικότερο σε ταχύτητα εκπαίδευσης αλλά και ικανότητα πρόβλεψης (Matthias Feurer, et al., 2019).

Μια πιο εξειδικευμένη έρευνα, η οποία περιγράφει την χρήση νευρωνικών δικτύων και SGD διαδικασιών για την βελτιστοποίηση της απόδοσης με τίτλο Towards Automatically-Tuned Deep Neural Networks από τους Hector Mendoza, Aaron Klein, Matthias Feurer, Jost Tobias Springenberg, Matthias Urban, Michael Burkart, Maximilian Dippel, Marius Lindauer, και Frank Hutter στην οποία παρουσιάζουν το Auto-Net το οποίο και αυτό λύνει το πρόβλημα CASH αλλά με την χρήση μια ενισχυμένης μεθοδολογίας SMAC. (Hector Mendoza, και συν., 2019).

Ακόμα μερικές έρευνες οι οποίες παρουσιάζουν ενδιαφέρον καθώς αναλύουν συγκεκριμένες διαδικασίες οι οποίες χρησιμοποιούνται κατά κόρον από τις νεότερες εφαρμογές αυτόματης μηχανικής μάθησης όπως το Meta-Learning (δηλαδή τον περιορισμό αλγορίθμων και υπερ-παραμέτρων από προηγούμενη εμπειρία) και το Bayesian Optimization (δηλαδή την βελτιστοποίηση υπερ-παραμέτρων βάση της πιθανότητας βελτίωσης σε κάθε επανάληψη):

Οι Frank Hutter, Holger H. Hoos and Kevin Leyton-Brown στο Paper «Sequential Model-Based Optimization for General Algorithm Configuration» στο οποίο γίνεται προσπάθεια ενίσχυσης του απλού γενικού SMBO (Sequential Model Based Optimization) με το Random Online Aggressive Racing (ROAR) και στην συνέχεια με το SMAC όπου από την πειραματική μελέτη φαίνεται να υπερ-νικά πάντα τα προηγούμενα δύο καθώς το τελευταίο λειτουργεί σειριακά με Bayesian Optimization (Frank Hutter, et al., 2011).

Οι Jasper Snoek, Hugo Larochelle and Ryan P. Adams στο Paper «Practical Bayesian Optimization of Machine Learning» όπου το Bayesian Optimization βρίσκει καλύτερες υπερ-παραμέτρους σημαντικά πιο γρήγορα από τις προηγούμενες προσεγγίσεις Grid (Σειριακή αναζήτηση), RandomGrid (Τυχαία αναζήτηση), GP EI (Expected Improvement) και GP EI MCMC (Expected Improvement Markov Chain Monte Carlo) (Jasper Snoek, και συν., 2012).

Ο Joaquin Vanschoren στο Paper «Meta-Learning: A Survey» το οποίο εξηγεί την διαδικασία meta-learning με την οποία διάφορες μετρήσεις συγκεντρώνονται από διάφορα σύνολα δεδομένων οι οποίες θα είναι παρόμοιες σε όμοια σύνολα δεδομένων. Το meta-learning μπορεί να καλύψει κάθε τύπου μηχανική μάθηση με βάση προηγούμενη εμπειρία. Στόχος είναι αυτή η εμπειρία να βελτιώνεται με κάθε σύνολο δεδομένων χωρίς αυτή να μένει στάσιμη. Χρησιμοποιώντας τέτοιες τεχνικές δεν χρειάζεται ποτέ να ξεκινάει μια διαδικασία αναζήτησης από την αρχή αλλά έχοντας μια βάση από προηγούμενη εμπειρία (Joaquin Vanschoren, 2019).

Οι Yang Li, Jiawei Jiang, Jinyang Gao, Yingxia Shao, Ce Zhang, Bin Cui στο Paper «Efficient Automatic CASH via Rising Bandits Rising Bandits» στο οποίο προτείνεται μια μέθοδος επίλυσης του προβλήματος CASH σε έναν μικρό χώρο υπερ-παραμέτρων με την χρήση MAB (Multi Armed Bandits). Στην συγκεκριμένη έρευνα η απόδοση της τεχνικής συγκρίνεται με τις νέες έρευνες όπως AutoWeka, AutoSKLearn, HyperoptSKLearn και άλλες όπου στα περισσότερα σύνολα δεδομένων υπερίσχυε. Τέλος η κατευθυντήρια μελλοντική γραμμή είναι η χρήση meta-learning για την επιτάχυνση του CASH καθώς θα περιοριστεί το εύρος αλγορίθμων προς αξιολόγηση (Yang Li, et al., 2020).

Οι Hadi S. Jomaa, Lars Schmidt-Thieme, Josif Grabocka στο Paper «Dataset2Vec: learning dataset meta-features» στο οποίο προτείνεται ένας meta-feature εξαγωγέας με όνομα Dataset2Vec χρησιμοποιώντας νευρωνικά δίκτυα. Τα meta-features είναι διάφορα χαρακτηριστικά του συνόλου δεδομένων όπως, ο αριθμός των δειγμάτων, ο αριθμός των χαρακτηριστικών, ο αριθμός των κλάσεων και άλλα. Με αυτόν τον τρόπο εκτιμώνται ποια σύνολα δεδομένων είναι παρόμοια και ποια δεν σχετίζονται. Η συγκεκριμένη τοποθέτηση είναι βασισμένη στο θεώρημα Kolmogorov-Arnold η οποία εφαρμόζεται για την εύρεση ημι-περιοδικών λύσεων για την μη γραμμική εξίσωση Schrödinger (Hadi S. Jomaa, et al., 2021).

Τέλος οι Maroua Bahri, Flavia Salutari, Andrian Putina και Mauro Sozio στο Paper «AutoML: State Of The Art With A Focus On Anomaly Detection, Challenges, And Research Directions» συγκεντρώνουν μερικές ολοκληρωμένες λύσεις αυτόματης μηχανικής μάθησης σε διάφορα πεδία όπως Supervised, Unsupervised και Semi-Supervised. Ειδικότερα παρουσιάζουν και θέτουν το ερώτημα, ότι αν η χρήση meta-learning μπορεί να εφαρμοστεί και σε Unsupervised προβλήματα το λεγόμενο meta-clustering (Maroua Bahri, και συν., 2022).

3. Μηχανική Μάθηση

Η μηχανική μάθηση είναι μια τεχνολογία που υπάρχει αρκετά χρόνια. Παρόλα αυτά δεν είχε διαδοθεί όσο σήμερα καθώς οι απαιτήσεις σε υπολογιστική ισχύ απαγορεύαν την χρήση της. Σήμερα, με διαθέσιμη υπολογιστική ισχύ, η μηχανική μάθηση είναι πραγματικότητα. Μηχανική μάθηση ονομάζουμε την δυνατότητα υπολογιστικών συστημάτων να μαθαίνουν χωρίς να έχουν προγραμματιστεί με συγκεκριμένους κανόνες. Διάφοροι αλγόριθμοι μπορούν να εξάγουν σημαντική πληροφορία από μεγάλα σύνολα δεδομένων προκειμένου να κάνουν τις απαραίτητες προβλέψεις η να πάρουν συγκεκριμένες αποφάσεις. Ο τομέας της μηχανικής μάθησης είναι στενά συνδεδεμένος με τον προγραμματισμό αλλά και την στατιστική αφού όλα τα μοντέλα εκφράζονται από μαθηματικές συναρτήσεις.

Η Μηχανική μάθηση έχει πολλές εφαρμογές και συγκεκριμένα πραγματοποιείται σε περιπτώσεις όπου μια ακολουθία κανόνων θα ήταν ανέφικτη. Παραδείγματα εφαρμογών αποτελούν τα spam φίλτρα (spam filtering), η πρόβλεψη κινδύνου (risk analysis), η αναγνώριση χαρακτήρων (OCR), οι εφαρμογές ανάλυσης εικόνων όπως κλινικές ακτινογραφίες αλλά και έως μηχανές αναζήτησης (Iqbal H. Sarker, 2021).

Πολλοί επιστήμονες υπολογιστών έδωσαν σαφείς καθοριστικούς όρους που χρησιμοποιούνται μέχρι και σήμερα:

- Tom M. Mitchell χρησιμοποίησε έναν ορισμό που μέχρι σήμερα αντιπροσωπεύει την μηχανική μάθηση: «Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από εμπειρία E ως προς μια κλάση εργασιών T και ένα μέτρο επίδοσης P, αν η επίδοσή του σε εργασίες της κλάσης T, όπως αποτιμάται από το μέτρο P, βελτιώνεται με την εμπειρία E» (Tom M. Mitchell, 1997).
- Ο Alan Turing στην εργασία του «Υπολογιστικές μηχανές και Νοημοσύνη», έθεσε το ερώτημα αν μπορούν οι μηχανές να σκεφτούν και αν είναι εφικτή η προσομοίωση λειτουργίας αυτών των συστημάτων που φτάνουν τον ανθρώπινο νου (Alan M. Turing, 1990).

Στο πεδίο της ανάλυσης δεδομένων, η μηχανική μάθηση είναι μια μέθοδος που χρησιμοποιείται για την επινόηση πολύπλοκων μοντέλων και αλγορίθμων που οδηγούν στην πρόβλεψη. Συγκεκριμένα θα εξεταστούν συγκεκριμένες έννοιες της μηχανικής μάθησης με την παρακάτω δομή προκειμένου να γίνει κατανοητό το επιστημονικό υπόβαθρο της μηχανικής μάθησης:

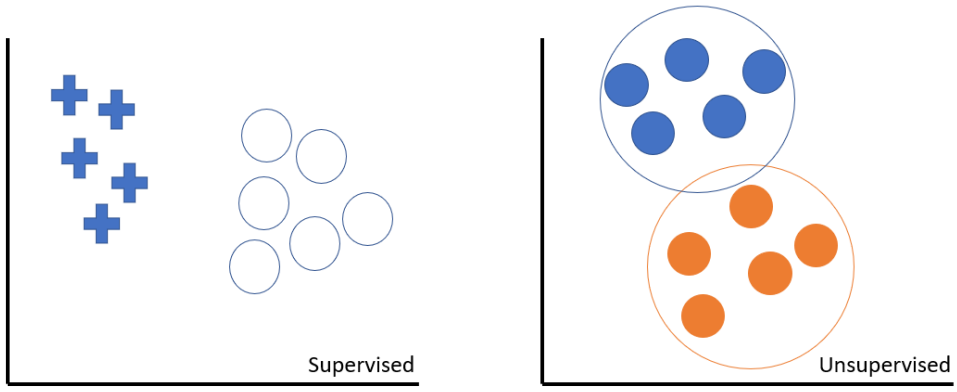
- Πεδία μηχανικής μάθησης, ποια είναι τα πεδία μηχανικής μάθησης, πως διαφέρουν και πότε επιλέγεται το κάθε πεδίο.
- Κατηγορίες μηχανικής μάθησης, ποιες είναι οι κατηγορίες μηχανικής μάθησης, πως διαφέρουν και πότε επιλέγεται η κάθε κατηγορία.
- Αλγόριθμοι μηχανικής μάθησης, ποιοι αλγόριθμοι μηχανικής μάθησης υπάρχουν, ποιοι παρουσιάζουν ομοιότητες και πως διαφέρουν.
- Υπερ-παράμετροι μηχανικής μάθησης, ποιες υπερ-παράμετροι μηχανικής μάθησης υπάρχουν, ποιες παρουσιάζουν ομοιότητες και πως διαφέρουν.
- Τεχνικές επιλογής αλγορίθμου, πως γίνεται η επιλογή κατάλληλου αλγορίθμου για συγκεκριμένο σύνολο δεδομένων και ποια ερευνητική πορεία πίσω από την επιλογή.
- Τεχνικές επιλογής υπερ-παραμέτρων, πως γίνεται η επιλογή κατάλληλων υπερ-παραμέτρων βάση αλγορίθμου για συγκεκριμένο σύνολο δεδομένων και ποια ερευνητική πορεία πίσω από την επιλογή.

3.1 Πεδία Μηχανικής Μάθησης

Στον χώρο της μηχανικής μάθησης υπάρχουν συγκεκριμένα πεδία ανάλυσης δεδομένων τα οποία προέκυψαν κατά την εξέλιξη της μηχανικής μάθησης (Sah, S, 2020), όπως φαίνεται παρακάτω:

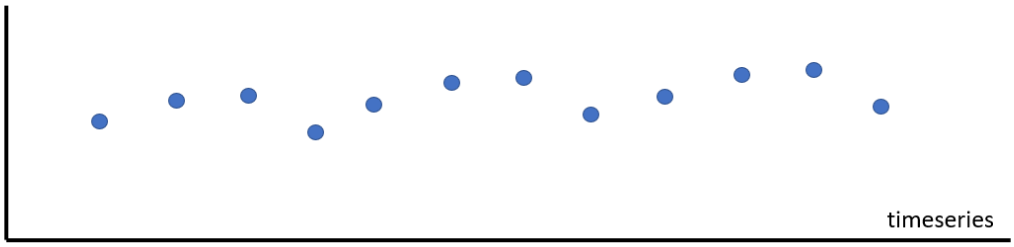
Εποπτευόμενη (Supervised), είναι η διαδικασία στην οποία αλγόριθμοι μπορούν να εκπαιδευτούν διότι γνωρίζουν εκ ‘των προτέρων (apriori) το αποτέλεσμα. Με αυτόν τον τρόπο μπορεί να γίνει επαλήθευση της εκπαίδευσης σε δεδομένα τα οποία γνωρίζουν ή σε άγνωστα δεδομένα για τους αλγορίθμους. Τέλος, μπορεί να γίνει η πιστοποίηση των προβλέψεων ανάμεσα στις προβλέψεις και στις πραγματικές τιμές.

Μη-Εποπτευόμενη (Un-Supervised), είναι η διαδικασία στην οποία αλγόριθμοι δεν εκπαιδεύονται διότι δεν γνωρίζουν το αποτέλεσμα αλλά μπορούν να εξάγουν πληροφορία για τα δεδομένα χωρίς την χρήση κάποιας εκπαίδευσης κατά το δοκούν. Συχνά ονομάζονται και νωθροί (Lazy).



Εικόνα 1. Εποπτευόμενη και Μη-εποπτευόμενη μάθηση

Άλλες τεχνικές πρόβλεψης που μπορεί να μην ανήκουν σε κάποιο πεδίο από την εποπτευομένη και την μη-εποπτευόμενη μάθηση διότι μπορεί να εμπίπτουν και στα δύο πεδία ή σε κανένα.



Εικόνα 2. Άλλα πεδία μηχανικής μάθησης

Συνοπτικά τα πεδία ανάλυσης διακρίνονται σε Εποπτευόμενη και Μη-Εποπτευόμενη όμως υπάρχουν και διαδικασίες μου μπορεί να εμπίπτουν και στις δύο κατηγορίες ή σε καμία από αυτές. Άλλες μορφές μηχανικής μάθησης. Σε αυτήν την εργασία θα γίνει μια εμβάθυνση στο πεδίο ανάλυσης της εποπτευόμενης μάθησης. Υπάρχουν δύο βασικές κατηγορίες ανάλυσης σε αυτό το πεδίο ανάλυσης όπως θα γίνει εμφανές παρακάτω.

3.2 Κατηγορίες Μηχανικής Μάθησης

Στον χώρο της μηχανικής μάθησης υπάρχουν πολλές κατηγορίες ανάλυσης δεδομένων καθώς αυτά ανήκουν σε κάποιο από τα προαναφερθείσα πεδία (Iqbal H. Sarker, 2021), όπως φαίνεται παρακάτω:

- Κατηγοριοποίησης (classification), είναι μια κατηγορία ανάλυσης που στόχο έχει την πρόβλεψη της κλάσης / ετικέτας ή αλλιώς ενός αλφαριθμητικού. Η ταξινόμηση, μπορεί να λύσει, εκτός από το κλασικό δυαδικό πρόβλημα (binary classification), και ένα πρόβλημα με πολλαπλά αλφαριθμητικά (multiclass classification). Για την εκπαίδευση του μοντέλου απαιτείται οι κατηγορίες να είναι γνωστές.
- Παλινδρόμησης (regression), είναι μια κατηγορία ανάλυσης που στόχο έχει την πρόβλεψη της τιμής ή αλλιώς ενός αριθμού. Η παλινδρόμηση, μπορεί να λύσει, εκτός από το κλασικό πρόβλημα εκτίμησης ενός ακέραιου αριθμού, και ένα πρόβλημα με συνεχής τιμές. Για την εκπαίδευση του μοντέλου απαιτείται οι κατηγορίες να είναι γνωστές.
- Ομαδοποίησης (clustering), είναι μια κατηγορία ανάλυσης που στόχο έχει την ομαδοποίηση κοινότυπων δεδομένων. Η ομαδοποίηση, μπορεί να λύσει, εκτός από το κλασικό πρόβλημα ομαδοποίησης δεδομένων σε ένα δυσδιάστατο χώρο, και ένα πρόβλημα σε πολυδιάστατο χώρο. Η ομαδοποίηση αυτή έχει ως στόχο να διαχωρίσει n παρατηρήσεις. Τέλος, μπορεί να χρησιμοποιηθεί ακόμα και με μη-αριθμητικές τιμές κάτω από συγκεκριμένες προϋποθέσεις και μετασχηματισμούς. Για την λειτουργία του μοντέλου δεν απαιτείται οι κατηγορίες να είναι γνωστές.
- Συσχέτισης (association), είναι μια κατηγορία ανάλυσης όπου στόχο έχει την συσχέτιση δεδομένων τα οποία προέρχονται από «καλάθια συναλλαγών». Η συσχέτιση μπορεί να εξάγει πληροφορίες οι οποίες εμφανίζονται με την πάροδο του χρόνου. Οι κανόνες συσχέτισης αποτελούν μία από τις σημαντικότερες και νεότερες τεχνικές εξόρυξης γνώσης από μεγάλες βάσεις δεδομένων. Τέλος μπορεί να χρησιμοποιηθεί ώστε να απεικονιστεί η συσχέτιση ανάμεσα στους προγόνους (ancestors) και στους απογόνους (descendants). Για την λειτουργία του μοντέλου δεν απαιτείται οι κατηγορίες να είναι γνωστές.
- Πρόβλεψη χρονο-σειράς (forecasting), είναι μια κατηγορία ανάλυσης όπου στόχο έχει την πρόβλεψη των επόμενων τιμών δεδομένου μιας ιστορικότητας. Η πρόβλεψη χρονο-σειράς μπορεί να λύσει ακόμη και προβλήματα συνδυαστικής ιστορικότητας (multi-variant). Μια χρονολογική σειρά είναι μια σειρά σημείων δεδομένων με ευρετηρίαση (είτε εισηγμένη είτε διαγραμμισμένη) με χρονο-σειρά. Συχνότερα, μια χρονολογική σειρά είναι μια ακολουθία που λαμβάνεται σε διαδοχικά ισαπέχουσες χρονικές στιγμές. Τέλος μπορεί να χρησιμοποιηθεί και για την ανάλυση τάσης, εποχικότητας και άλλα.

Συνοπτικά οι κατηγορίες ανάλυσης διακρίνονται στον παρακάτω πίνακα:

Εποπτευόμενη	**Μη-Εποπτευόμενη**	**Άλλες**
Classification	Clustering	Forecasting
Regression	Association	

Πίνακας 1. Κατηγορίες Μηχανικής Μάθησης

Σε αυτήν την εργασία θα γίνει μια εμβάθυνση στις κατηγορίες ανάλυσης της ταξινόμησης αλλά και της παλινδρόμησης. Υπάρχουν αρκετές ομοιότητες ανάμεσα σε αυτές τις δύο κατηγορίες μηχανικής μάθησης όπως θα διαπιστωθεί παρακάτω.

3.3 Αλγόριθμοι Μηχανικής Μάθησης

Όπως είναι ήδη εμφανές, υπάρχει πληθώρα αλγορίθμων ανά πεδίο και κατηγορία ανάλυσης δεδομένων (Jason Brownlee, 2016). Πολλοί από αυτούς τους αλγορίθμους εμφανίζουν κοινά χαρακτηριστικά αφού έχουν σχεδιαστεί ακριβώς με τον ίδιο τρόπο, όμως για την επίλυση διαφορετικού προβλήματος σε διαφορετικές κατηγορίες ανάλυσης (classification / regression). Σε αυτές τις περιπτώσεις η κεντρική ιδέα και η σχεδιαστική λειτουργία είναι η ίδια, επομένως είναι λογικό να παρουσιάζουν και παρόμοιες παραμετρικές απαιτήσεις.

Έτσι θα γίνει μια παρουσίαση των αλγορίθμων και των υπερ-παραμέτρων τους ανά «οικογένεια» και «αρχή λειτουργίας» στην οποία ανήκουν καθώς αλγόριθμοι με κοινή «οικογένεια» και «αρχή λειτουργίας» έχουν σε μεγάλο βαθμό τα ίδια χαρακτηριστικά, αν όχι ίδια ενώ άλλοι δεν μπορούν να παρομοιαστούν με κάποιον από τους υπόλοιπους κάτι που τους καθιστά μοναδικούς στο είδος τους. Κάθε αλγόριθμος έχει αναπτυχθεί ή βασιστεί σε κάποια έρευνα και συνήθως είναι καταλληλότερος για συγκεκριμένα σύνολα δεδομένων.

Στον παρακάτω πίνακα φαίνονται μερικοί από τους πιο γνωστούς αλγορίθμους μηχανικής μάθησης σε μια οργανωμένη μορφή υπό την «οικογένεια» και «αρχή»:

ID	Οικογένεια	Αρχή	Κατηγοριοποίηση
0	ensemble	RandomForest	RandomForestClassifier
1	tree	DecisionTree	DecisionTreeClassifier
2	neighbors	KNeighbors	KNeighborsClassifier
3	svm	SupportVectors	SVC
4	linear	StohasticGradient	SGDClassifier
5	linear	Ridge	RidgeClassifier
6	linear	N/A	LogisticRegression
7	linear	N/A	Perceptron
8	linear	N/A	PassiveAggressiveClassifier
9	naive_bayes	N/A	MultinomialNB
10	naive_bayes	N/A	GaussianNB
11	naive_bayes	N/A	BernoulliNB
12	discriminant_analysis	N/A	LinearDiscriminantAnalysis

ID	Οικογένεια	Αρχή	Παλινδρόμηση
0	ensemble	RandomForest	RandomForestRegressor
1	tree	DecisionTree	DecisionTreeRegressor
2	neighbors	KNeighbors	KNeighborsRegressor
3	svm	SupportVectors	SVR
4	linear	StohasticGradient	SGDRegressor
5	linear	Ridge	Ridge
6	linear	N/A	LinearRegression
7	linear	N/A	BayesianRidge
8	linear	N/A	ARDRegression

Πίνακας 2. Αλγόριθμοι Μηχανικής Μάθησης

Σε αυτήν την διπλωματική εργασία θα γίνει μια εμβάθυνση στον τρόπο επιλογής υπερ-παραμέτρων για τους αλγόριθμους που βρίσκονται στην ίδια «οικογένεια» και «αρχή» (γκρι χρώμα), ενώ θα γίνουν αναφορές και στους αλγορίθμους και στις υπερ-παραμέτρους που δεν φαίνεται να παρουσιάζουν ομοιότητες με κάποιον από τους υπόλοιπους.

3.4 Υπερπαράμετροι Αλγορίθμων Μηχανικής Μάθησης

Όπως είναι ήδη εμφανές, η μηχανική μάθηση έχει οργανωθεί σε πεδία και κατηγορίες με επίκεντρο τους αλγορίθμους και τις υπερ-παραμέτρους τους. Πρέπει να σημειωθεί πως οι υπερ-παράμετροι αντιπροσωπεύουν τα «ζωτικής σημασίας όργανα» στην λειτουργία ενός αλγορίθμου αφού μπορούν και αλλάξουν συμπεριφορές και διαδικασίες. Οι αλλαγές αυτές είναι σημαντικές όταν πρόκειται να γίνει μια ανάλυση σε ένα συγκεκριμένο σύνολο δεδομένων (Philipp Probst, et al., 2019).

Οι υπερ-παράμετροι ελέγχουν διαδικασίες βάση των τιμών τους όπως την συμπεριφορά της λειτουργίας εκμάθησης και της πρόβλεψης. Η λέξη «υπέρ» υποδηλώνει ότι είναι παράμετροι «υψηλότερου επιπέδου». Κάποιες κοινές υπερ-παράμετροι που συνήθως συναντιούνται στους περισσότερους αλγορίθμους είναι το random seed για τον ορισμό τυχαιότητητας της γεννήτριας αριθμών και το n jobs δηλαδή το πόσες παράλληλες διαδικασίες μπορεί να δημιουργήσει ένας αλγόριθμος.

Οι υπερ-παράμετροι τείνουν να είναι πλέον δύσκολες στην απομνημόνευση και στην κατανόηση αφού έχει παρατηρηθεί πως κάθε αλγόριθμος μπορεί να έχει έως και μερικές δεκάδες υπερ-παραμέτρους. Αντίθετα πολλές υπερ-παράμετροι φαίνεται παρουσιάζουν ομοιότητες ανάμεσα σε αλγορίθμους στην ίδια «οικογένεια» και «αρχή» όπως φαίνεται στον παρακάτω πίνακα:

ID	**Οικογένεια**	**Αρχή**	**Τύπος Υπερ-παραμέτρων**
0	ensemble	RandomForest	Υπερ-παράμετροι σχετικές με τα δένδρα
1	tree	DecisionTree	Υπερ-παράμετροι σχετικές με τα δένδρα
2	neighbors	KNeighbors	Υπερ-παράμετροι σχετικές με κοντινότερους γείτονες
3	svm	SupportVectors	Υπερ-παράμετροι σχετικές με διανυσματικές γεννήτριες
4	linear	StohasticGradient	Υπερ-παράμετροι σχετικές με στοχαστικές διαδικασίες
5	linear	Ridge	Υπερ-παράμετροι σχετικές με κανόνες τακτοποίησης

Πίνακας 3. Υπερ-παράμετροι Μηχανικής Μάθησης

Οι ειδικοί μηχανικής μάθησης που σχεδιάζουν ένα μοντέλο, επιλέγουν και ορίζουν τιμές υπερ-παραμέτρων που θα χρησιμοποιήσει ο αλγόριθμος εκμάθησής πριν καν ξεκινήσει η εκπαίδευση του μοντέλου. Υπό αυτό το πρίσμα, οι υπερ-παράμετροι λέγεται ότι είναι εξωτερικές του μοντέλου επειδή το μοντέλο δεν μπορεί να αλλάξει τις τιμές του κατά τη διάρκεια της εκμάθησης η αλλιώς της εκπαίδευσης.

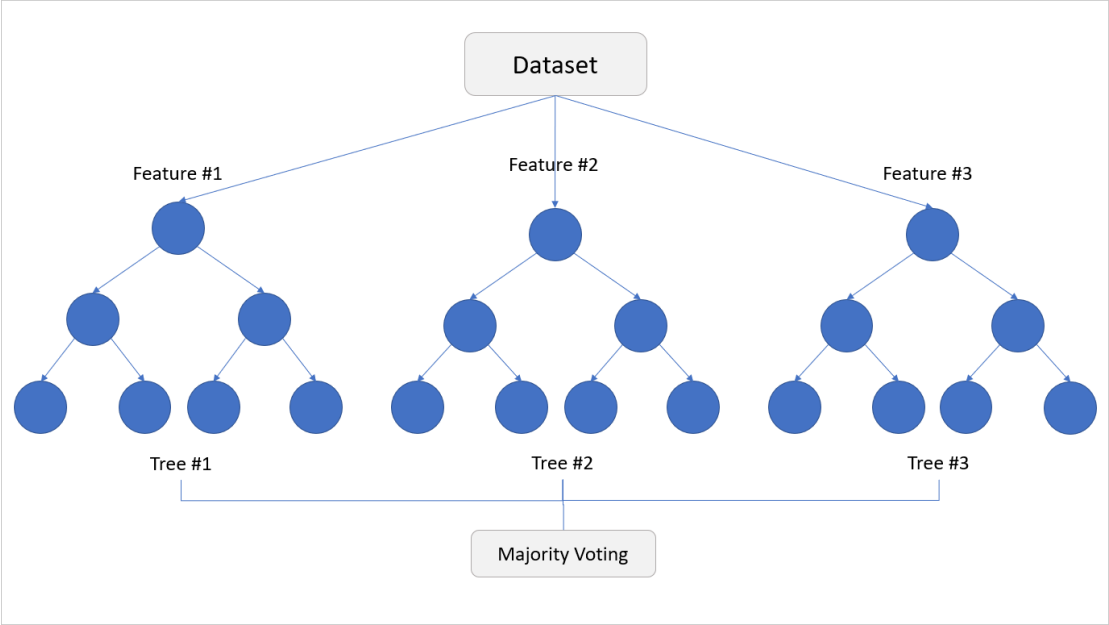
Οι υπερ-παράμετροι χρησιμοποιούνται από τον αλγόριθμο μάθησης όταν μαθαίνει, και αποτελούν μέρος του προκύπτοντος μοντέλου. Στο τέλος της μαθησιακής διαδικασίας, έχουμε τις παραμέτρους του εκπαιδευμένου μοντέλου που ουσιαστικά είναι αυτό που αποκαλούμε μοντέλο. Οι υπερ-παράμετροι που χρησιμοποιήθηκαν κατά τη διάρκεια της εκπαίδευσης δεν αποτελούν μέρος αυτού του μοντέλου.

Λόγο της πληθώρας των υπερ-παραμέτρων, κρίνεται σκόπιμο να γίνει μια επισκόπηση των υπερ-παραμέτρων για κάθε αλγόριθμο ώστε να γίνει κατανοητή η λειτουργία τους, να ερευνηθεί το πως μπορεί να γίνει η επιλογή τους αλλά και να ειδικότερα ποιες τιμές μπορούν να πάρουν αυτές οι παράμετροι.

RandomForest, διακρίνεται σε classifier (RandomForestClassifier) και regressor (RandomForestRegressor).

Είναι ένας μέτα-εκτιμητής ο οποίος εφαρμόζει δένδρα απόφασης και χρησιμοποιεί δένδρα με τις συχνότερες επιλογές για διαδικασίες κατηγοριοποίησης, ενώ χρησιμοποιεί δένδρα με μέσους όρους για διαδικασίες παλινδρόμησης. Ένα δάσος αποτελείται από δέντρα και αφενός όσο περισσότερα δέντρα έχει ένα δάσος, τόσο πιο εύρωστο είναι. Το Random Forest δημιουργεί δέντρα αποφάσεων σε τυχαία επιλεγμένα δείγματα δεδομένων, λαμβάνει πρόβλεψη από κάθε δέντρο και επιλέγει την καλύτερη λύση μέσω ψηφοφορίας. Τα τυχαία δάση έχουν τη συνήθεια των δέντρων αποφάσεων, δηλαδή την δυνατότητα να προσαρμόζονται υπερβολικά στο σετ εκπαίδευσής τους. Ωστόσο, τα χαρακτηριστικά δεδομένων μπορούν να επηρεάσουν την απόδοσή τους. Τα τυχαία δάση χρησιμοποιούνται συχνά ως μοντέλα "blackbox" στις επιχειρήσεις, καθώς δημιουργούν λογικές προβλέψεις σε ένα ευρύ φάσμα δεδομένων, ενώ απαιτούν μικρή παραμετροποίηση.

Ο πρώτος αλγόριθμος για δάση τυχαίας απόφασης δημιουργήθηκε το 1995 από τον Tin Kam Ho χρησιμοποιώντας τη μέθοδο του τυχαίου υπο-χώρου, η οποία, στη διατύπωση του Ho, είναι ένας τρόπος για την εφαρμογή της προσέγγισης της «στοχαστικής διάκρισης» στην ταξινόμηση που προτάθηκε από τον Eugene Kleinberg. Μια επέκταση του αλγορίθμου αναπτύχθηκε από τον Leo Breiman και την Adele Cutler, οι οποίοι κατοχύρωσαν το "Random Forests" ως εμπορικό σήμα το 2006 (από το 2019, ιδιοκτησία της Minitab, Inc.). Η επέκταση συνδυάζει την ιδέα του Breiman για το «bagging» και την τυχαία επιλογή χαρακτηριστικών, που εισήχθη πρώτα από τον Ho και αργότερα ανεξάρτητα από τους Amit και Geman προκειμένου να κατασκευαστεί μια συλλογή δέντρων αποφάσεων με ελεγχόμενη διακύμανση.



Εικόνα 3. Αλγόριθμος RandomForest

Στην παραπάνω εικόνα απεικονίζεται η λειτουργία του Random Forest και πως κάθε χαρακτηριστικό δημιουργεί ένα δένδρο απόφασης. Κάθε δέντρο αντλεί ένα τυχαίο δείγμα από το αρχικό σύνολο δεδομένων κατά τη δημιουργία των διαχωρισμών του, προσθέτοντας ένα επιπλέον στοιχείο τυχαιότητας που αποτρέπει την υπερβολική προσαρμογή.

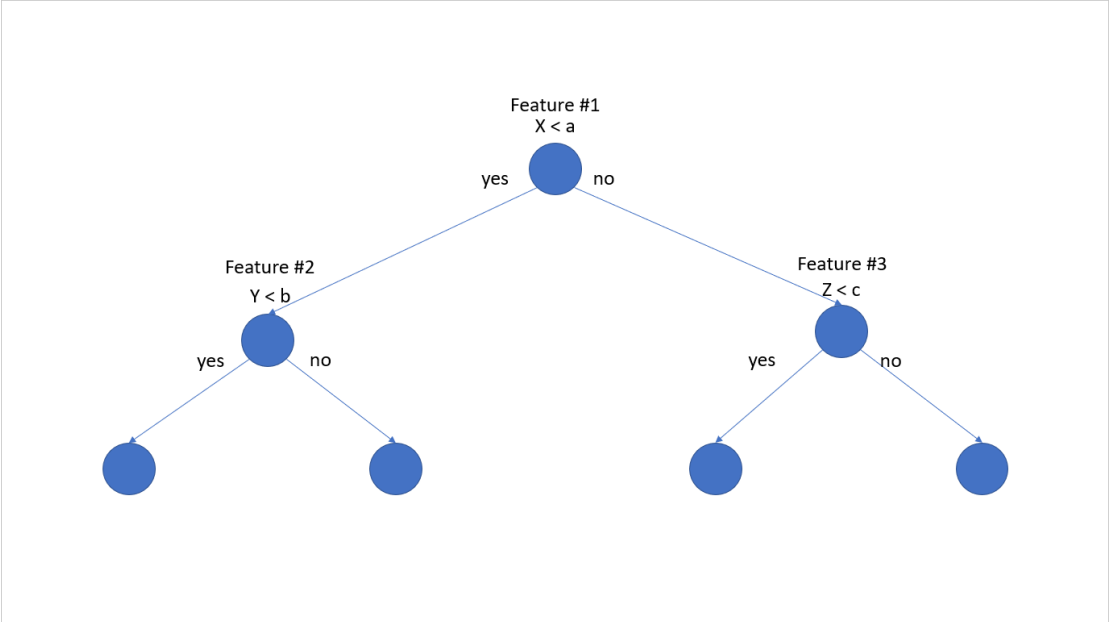
Οι υπερ-παράμετροι που χρησιμοποιούνται με την χρήση αυτής της μεθόδου:

- n_estimators, τα διαθέσιμα παραγόμενα δένδρα στο δάσος για επιλογή
  for = both, type = numeric, values = integer, default = 100
- criterion, η μέθοδος εκτίμησης της ποιότητας του διαχωρισμού στους κόμβους
  for = classifier, type = categorical, values = {gini, entropy}, default = gini
  for = regressor, type = categorical, values = {mse, mae}, default = mse
- max_depth, το μέγιστο βάθος των δένδρων με ανοικτά φύλλα
  for = both, type = numeric, values = integer, default = none
- min_samples_split, ο ελάχιστος αριθμός δειγμάτων-φύλλα σε κόμβο του δένδρου
  for = both, type = numeric, values = float, default = 2
- min_weight_fraction_leaf, το ελάχιστο άθροισμα βάρους των δειγμάτων-φύλλα σε κόμβο
  for = both, type = numeric, values = float, default = 0.0
- max_features, ο αριθμός των χαρακτηριστικών για τον διαχωρισμό δεδομένων
  for = both, type = categorical, values = {auto, sqrt, log2}, default = auto
- max_leaf_nodes, το μέγεθος του δένδρου ανάλογα τα φύλλα του
  for = both, type = numeric, values = integer, default = none
- min_impurity_decrease, οι κόμβοι θα χωριστούν αν το χώρισμα έχει impurity μεγαλύτερο ή ίσο από αυτή την τιμή
  for = both, type = numeric, values = float, default = 0.0
- min_impurity_split, οι κόμβοι θα χωριστούν αν το impurity είναι μεγαλύτερο ή ίσο από αυτή την τιμή.
  for = both, type = numeric, values = float, default = 0.0
- bootstrap, η μέθοδος κατασκευής του δένδρων, με η χωρίς, όλο το σύνολο δεδομένων για κάθε δένδρο που παράγεται
  for = both, type = boolean, values = {true, false}, default = true
- oob_score, χρήση out-of-bag δειγμάτων για την εκτίμησης γενικευμένου αποτελέσματος
  for = both, type = boolean, values = {true, false}, default = true
- n_jobs, ο αριθμός των εργασιών προς παράλληλη εκτέλεση
  for = both, type = numeric, values = integer, default = none(1)
- random_state, ο αριθμός τυχαιότητας της δειγματοληψίας, το οποίο έχει επίπτωση στο bootstrap όταν το σύνολο δεδομένων για κάθε δένδρο είναι μεγαλύτερου του μέγιστου αριθμού φύλλων
  for = both, type = numeric, values = integer, default = none
- verbose, το επίπεδο ανάλυσης της περιγραφής των εργασιών
  for = both, type = numeric, values = integer, default = 0
- warm_start, χρήση υπάρχοντών δένδρων και χρήση περισσότερων εκτιμητών
  for = both, type = boolean, values = {true, false}, default = false
- ccp_alpha, αριθμός που δηλώνει την πολυπλοκότητα, τιμές με μεγαλύτερο κόστος θα έχουν ως αποτέλεσμα το κούρεμα των δένδρων
  for = both, type = numeric, values = float, default = 0.0
- max_samples, αριθμός των μέγιστων δειγμάτων για κάθε εκτιμητή αν το bootstrap είναι true.
  for = both, type = numeric, values = float, default = none
- class_weight, η μέθοδος ισορροπίας μεταξύ των κλάσεων των δειγμάτων.
  for = classifier, type = categorical, values = {dict[class_label: weight],balanced, balanced_subsample}, default = none

DecisionTree, διακρίνεται σε classifier (DecisionTreeClassifier) και regressor (DecisionTreeRegressor).

Είναι ένας μέτα-εκτιμητής ο οποίος εφαρμόζει μοντέλα αποφάσεων. Ένα δέντρο απόφασης είναι μια δενδροειδής δομή όπου ένας εσωτερικός κόμβος αντιπροσωπεύει ένα χαρακτηριστικό, ο κλάδος αντιπροσωπεύει έναν κανόνα απόφασης και κάθε κόμβος φύλλου αντιπροσωπεύει το αποτέλεσμα (κλάση). Ο αρχικός κόμβος σε ένα δέντρο αποφάσεων ονομάζεται και ρίζα του δένδρου. Το δένδρο απόφασης μαθαίνει να χωρίζει με βάση τις τιμές του χαρακτηριστικού και διαχωρίζει το δέντρο με αναδρομικό τρόπο. Ακόμη τα δένδρα αυτά παρέχουν διαδικασίες κλαδέματος για να αφαιρεθούν άσχετα κλαδιά που θα μπορούσαν να μειώσουν την ακρίβεια. Το κλάδεμα περιλαμβάνει τον εντοπισμό ακραίων σημείων, σημείων δεδομένων που δεν ακολουθούν κανόνες και που θα μπορούσαν να απορρίψουν τους υπολογισμούς δίνοντας υπερβολική βαρύτητα σε σπάνιες εμφανίσεις στα δεδομένα. Ειδικότερα, ένα δέντρο αποφάσεων είναι ένα εργαλείο υποστήριξης που χρησιμοποιεί ένα μοντέλο αποφάσεων με δενδροειδή μορφή με τις πιθανές συνέπειές. Είναι ένας τρόπος εμφάνισης ενός αλγόριθμου που περιέχει μόνο εντολές ελέγχου υπό όρους. Τα δέντρα αποφάσεων είναι εύκολα στην κατανόηση και συνήθως χρησιμοποιούνται στην επιχειρησιακή έρευνα, ειδικά στην ανάλυση αποφάσεων, αλλά είναι επίσης ένα δημοφιλές εργαλείο στη μηχανική μάθηση.

Το πρώτο σχετικά άρθρο που αναπτύσσει μια προσέγγιση «δέντρου αποφάσεων» χρονολογείται από το 1959, ένας Βρετανός ερευνητής, ο William Belson, σε μια εργασία με τίτλο Matching and Prediction on the Principle of Biological Classification, (JRSS, Series C, Applied Statistics, Vol. 8, No. 2, June, 1959, σελ. 65-75)



Εικόνα 4. Αλγόριθμος DecisionTree

Στην παραπάνω εικόνα απεικονίζεται η λειτουργία του Decision Tree όπου η μεταβλητή στόχος μπορεί να λάβει ένα διακριτό σύνολο τιμών. Σε αυτές τις δομές δέντρων, τα φύλλα αντιπροσωπεύουν ετικέτες κλάσεων και τα κλαδιά αντιπροσωπεύουν τα χαρακτηριστικά που οδηγούν σε αυτές τις ετικέτες κλάσεων. Τα δέντρα απόφασης επίσης δεν χτίζονται με διάφορες υποθέσεις, όπως η κανονική κατανομή.

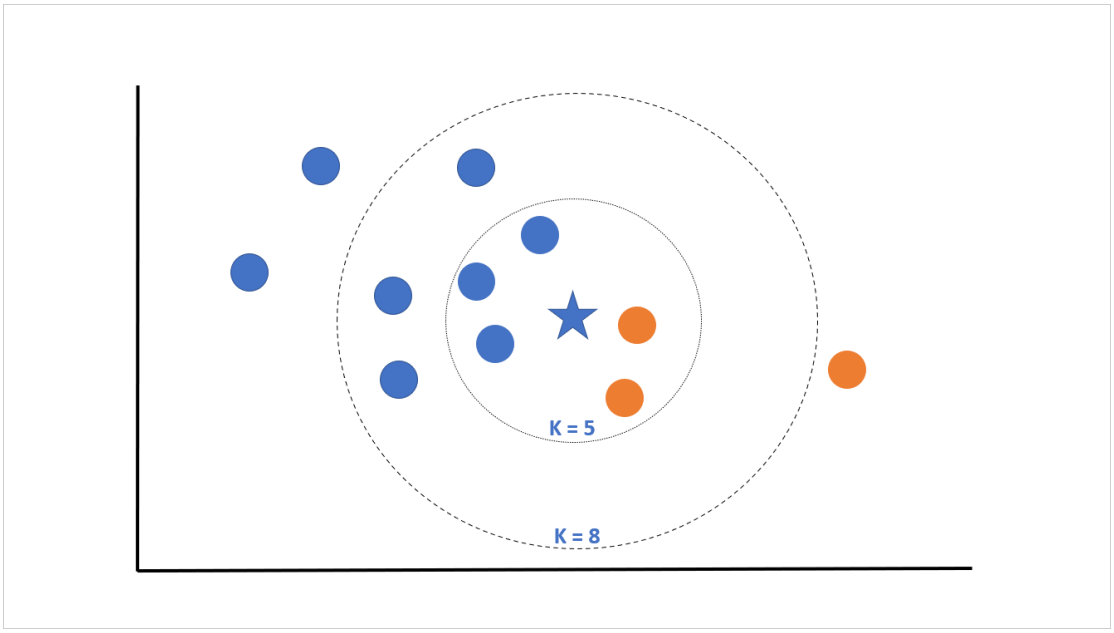
Οι υπερ-παράμετροι που χρησιμοποιούνται με την χρήση αυτής της μεθόδου:

- criterion, η μέθοδος εκτίμησης της ποιότητας του διαχωρισμού στους κόμβους
  for = classifier, type = categorical, values =
  {gini, entropy}, default = gini
  for = regressor, type = categorical, values =
  {mse, friedman_mse, mae, poisson}, default = gini
- splitter, η μέθοδος διαχωρισμού σε κάθε κόμβο
  for = both, type = categorical, values = {best, random}, default = best
- max_depth, το μέγιστο βάθος των δένδρων με ανοικτά φύλλα
  for = both, type = numeric, values = integer, default = none
- min_samples_split, ο ελάχιστος αριθμός δειγμάτων για τον διαχωρισμό εσωτερικού κόμβου
  for = both, type = numeric, values = float, default = 2
- min_samples_leaf, ο ελάχιστος αριθμός δειγμάτων που απαιτούνται σε ένα κόμβο
  for = both, type = numeric, values = float, default = 1
- min_weight_fraction_leaf, το ελάχιστο βάρος του αθροίσματος των δειγμάτων στα φύλλα
  for = both, type = numeric, values = float, default = 0.0
- max_features, ο αριθμός των χαρακτηριστικών για τον διαχωρισμό δεδομένων
  for = both, type = categorical, values = {auto, sqrt, log2}, default = auto
- random_state, ο αριθμός τυχαιότητας της δειγματοληψίας, το οποίο έχει επίπτωση στο splitter όταν το σύνολο δεδομένων για κάθε δένδρο είναι μεγαλύτερου του μέγιστου αριθμού φύλλων
  for = both, type = numeric, values = integer, default = none
- max_leaf_nodes, το μέγεθος του δένδρου ανάλογα τα φύλλα του το οποίο θα καθορίσει και τα επίπεδα του δένδρου
  for = both, type = numeric, values = integer, default = none
- min_impurity_decrease, οι κόμβοι θα χωριστούν αν το χώρισμα έχει impurity μεγαλύτερο ή ίσο από αυτή την τιμή
  for = both, type = numeric, values = float, default = 0.0
- min_impurity_split, οι κόμβοι θα χωριστούν αν το impurity είναι μεγαλύτερο ή ίσο από αυτή την τιμή.
  for = both, type = numeric, values = float, default = 0.0
- ccp_alpha, αριθμός που δηλώνει την πολυπλοκότητα, τιμές με μεγαλύτερο κόστος θα έχουν ως αποτέλεσμα το κούρεμα των δένδρων
  for = both, type = numeric, values = float, default = 0.0
- class_weight, η μέθοδος ισορροπίας μεταξύ των κλάσεων των δειγμάτων
  for = classifier, type = categorical, values =
  {dict[class_label: weight],balanced, balanced_subsample}, default = none

KNeighbors, διακρίνεται σε classifier (KNeighborsClassifier) και regressor (KNeighborsRegressor).

Είναι ένας μέτα-εκτιμητής ο οποίος εφαρμόζει κοντινότερους γείτονες και χρησιμοποιείται τόσο για ταξινόμηση όσο και για παλινδρόμηση. Ο KNN είναι ένας μη παραμετρικός η αλλιώς «νωθρός αλγόριθμος» μάθησης. Μη παραμετρικός σημαίνει ότι δεν υπάρχει υπόθεση για την υποκείμενη κατανομή δεδομένων. Οι γείτονες λαμβάνονται από ένα σύνολο αντικειμένων για τα οποία είναι γνωστή η κλάση (για ταξινόμηση) ή η τιμή αντικειμένου (για παλινδρόμηση). Αυτό μπορεί να θεωρηθεί ως το σετ εκπαίδευσης για τον αλγόριθμο, αν και δεν απαιτείται ρητό βήμα εκπαίδευσης. Με άλλα λόγια, η δομή του μοντέλου καθορίζεται από το σύνολο δεδομένων. Το K-Nearest Neighbors (K-NN) είναι ένας από τους απλούστερους αλγόριθμους μηχανικής εκμάθησης. Όταν εμφανίζεται μια νέα κατάσταση, εξετάζει όλες τις προηγούμενες εμπειρίες και αναζητά τις k πιο κοντινές εμπειρίες. Αυτές οι εμπειρίες (ή σημεία δεδομένων) είναι αυτό που ονομάζουμε k πλησιέστερους γείτονες.

Η πρώτη εφαρμογή της ιδέας αναπτύχθηκε από την Evelyn Fix και τον Joseph Hodges το 1951 ενώ αργότερα επεκτάθηκε από τον Thomas Cover. Ο αλγόριθμος αυτός μπορεί να χρησιμοποιηθεί και για μη αριθμητικές τιμές. Μια μετρική απόστασης που χρησιμοποιείται συνήθως για συνεχείς μεταβλητές είναι η Ευκλείδεια απόσταση ενώ για τις διακριτές μεταβλητές, όπως για την ταξινόμηση κειμένου, μπορεί να χρησιμοποιηθεί μια άλλη μέτρηση όπως είναι η απόσταση Hamming).



Εικόνα 5. Αλγόριθμος KNeighbors

Στην παραπάνω εικόνα απεικονίζεται η λειτουργία του KNeighbors όπου ένα νέο σημείο δεδομένων δίνεται στο σύνολο δεδομένων, και στην συνέχεια ερευνάται σε ποια ομάδα εμπίπτει σύμφωνα με την μετρική της απόστασης που έχει καθοριστεί. Συνοπτικά η λειτουργία του αλγορίθμου καθορίζει την απόσταση από το σημείο δεδομένων έως τα άλλα σημεία που είναι πλησιέστερα σε αυτό.

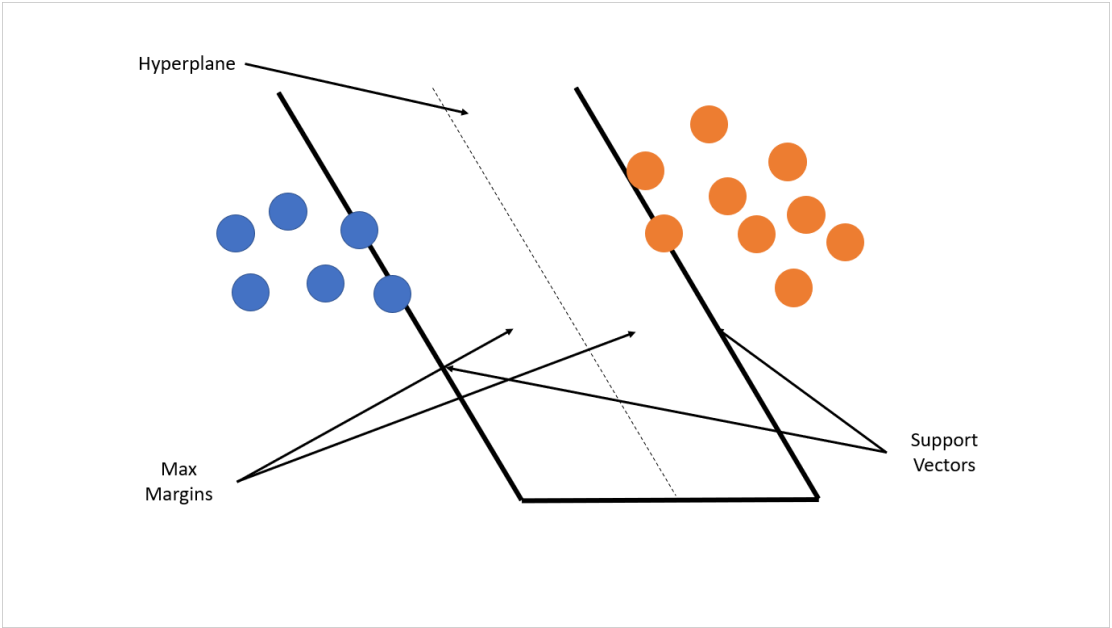
Οι υπερ-παράμετροι που χρησιμοποιούνται με την χρήση αυτής της μεθόδου:

- n_neighbors, ο αριθμός γειτόνων που θα χρησιμοποιηθούν
  for = both, type = numeric, values = integer, default = 5
- weights, τα βάρη συμμετοχής στην μέθοδο πρόβλεψης
  for = both, type = categorical, values = {uniform, distance}, default = uniform
- algorithm, ο αλγόριθμος υπολογισμού του κοντινότερου γείτονα
  for = both, type = categorical, values =
  {ball_tree, kd_tree, brute, auto}, default = auto
- leaf_size, το μέγεθος των φύλλων για τους αλγορίθμους δένδρων
  for = both, type = numeric, values = integer, default = 30
- p, ο εκθέτης της μεθόδου της απόστασης Minkowski
  for = both, type = categorical, values =
  {1(manhattan),2(euclidean)} default = 2
- metric, η μέθοδος της απόστασης Minkowski
  for = both, type = categorical, values =
  {euclidean,manhattan,chebyshev,minkowski,
  wminkowski,seuclidean,mahalanobis,haversine,hamming,canberra,braycurtis, jaccard,
  matching,dice,kulsinski,rogerstanimoto,russellrao,sokalmichener,sokalsneath,pyfunc}
  default = minkowski
- metric_params, επιπρόσθετες παράμετροι στην μέθοδο απόστασης
  for = both, type = categorical, values = {1,2} default = 2
- n_jobs, ο αριθμός των εργασιών προς παράλληλη εκτέλεση
  for = both, type = numeric, values = integer, default = none (1)

SupportVectors, διακρίνεται σε classifier (SVC) και regressor (SVR).

Είναι ένας μέτα-εκτιμητής ο οποίος κατασκευάζει ένα υπερ-επίπεδο σε έναν πολυδιάστατο χώρο για να διαχωρίσει διαφορετικές κλάσεις. Το SVM δημιουργεί το βέλτιστο υπερ-επίπεδο με επαναληπτικό τρόπο, το οποίο χρησιμοποιείται για την ελαχιστοποίηση ενός σφάλματος. Η βασική ιδέα του SVM είναι να βρει ένα μέγιστο οριακό υπερ-επίπεδο (MMH) που διαιρεί καλύτερα το σύνολο δεδομένων σε κλάσεις. Λαμβάνοντας υπόψη ένα σύνολο παραδειγμάτων εκπαίδευσης, το καθένα επισημαίνεται ότι ανήκει σε μία από τις δύο κατηγορίες, ένας αλγόριθμος εκπαίδευσης SVM δημιουργεί ένα μοντέλο που εκχωρεί νέα παραδείγματα στη μία ή στην άλλη κατηγορία. Το SVM αντιστοιχίζει παραδείγματα εκπαίδευσης σε σημεία στο χώρο, ώστε να μεγιστοποιήσει το πλάτος του χάσματος μεταξύ των δύο κατηγοριών. Στη συνέχεια, τα νέα παραδείγματα χαρτογραφούνται στον ίδιο χώρο και προβλέπεται ότι ανήκουν σε μια κατηγορία με βάση την πλευρά του κενού.

Αναπτύχθηκε στα AT&T Bell Laboratories από τον Vladimir Vapnik με συναδέλφους (Boser, Guyon, Cortes και Vapnik). Τα SVM είναι μια από τις πιο ισχυρές μεθόδους πρόβλεψης, που βασίζονται σε στατιστικά πλαίσια μάθησης ή στη θεωρία VC που προτάθηκαν από τους Vapnik (1982, 1995) και Chervonenkis (1974).



Εικόνα 6. Αλγόριθμος SupportVectors

Στην παραπάνω εικόνα απεικονίζεται η λειτουργία του SVM όπου ένα υπερ-επίπεδο ή ένα σύνολο υπερ-επίπεδων σε ένα χώρο υψηλής διάστασης, το οποίο μπορεί να χρησιμοποιηθεί για ταξινόμηση, παλινδρόμηση ή άλλες εργασίες όπως η ανίχνευση ακραίων σημείων. Διαισθητικά, ένας καλός διαχωρισμός επιτυγχάνεται από το υπερ-επίπεδο που έχει τη μεγαλύτερη απόσταση από το πλησιέστερο σημείο εκπαίδευσης δεδομένων οποιασδήποτε κατηγορίας (το λεγόμενο περιθώριο), αφού γενικά όσο μεγαλύτερο είναι το περιθώριο, τόσο μικρότερο είναι το σφάλμα γενίκευσης του ταξινομητή.

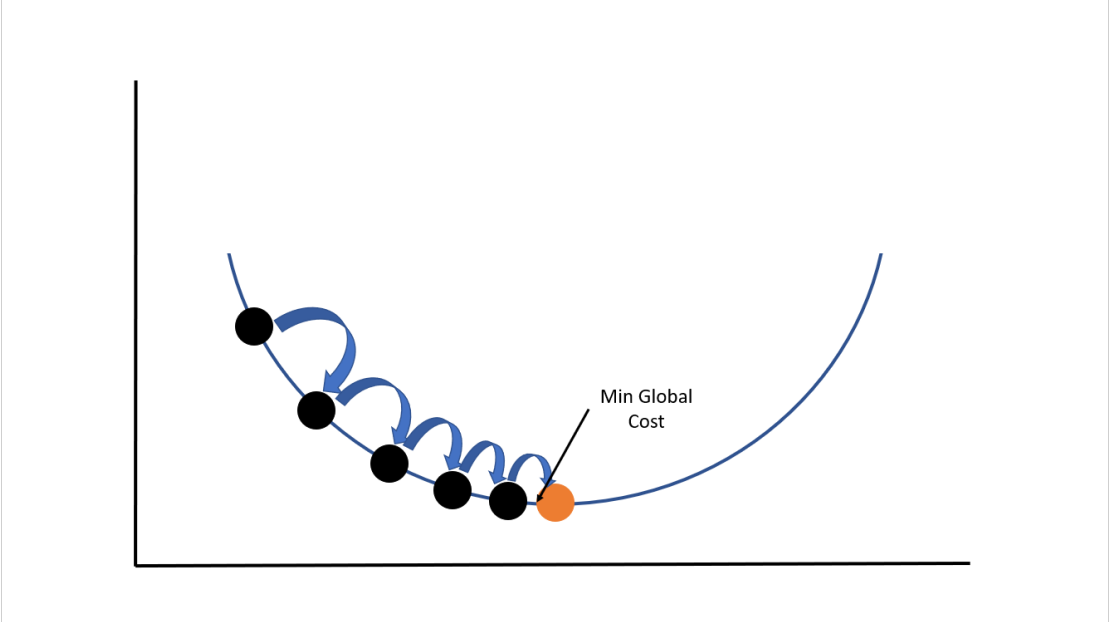
Οι υπερ-παράμετροι που χρησιμοποιούνται με την χρήση αυτής της μεθόδου:

- C, το βάρος τακτοποίησης
  for = both, type = numeric, values = float, default = 1
- kernel, ο πυρήνας που θα χρησιμοποιηθεί στον αλγόριθμο
  for = both, type = categorical, values = {linear,poly,rbf,sigmoid,precomputed}
- degree, ο βαθμός διαστάσεων για τον πυρήνα
  for = both, type = numeric, values = integer, default = 3
- gamma, παράμετρος συντελεστή πυρήνα
  for = both, type = categorical, values = {scale, auto}, default = scale
- coef0, ανεξάρτητη παράμετρος συντελεστή του πυρήνα
  for = both, type = numeric, values = float, default = 0.0
- shrinking, ευρετική συρρίκνωση
  for = both, type = boolean, values = {true, false}, default = true
- probability, η μέθοδος απόφασης της πρόβλεψης
  for = both, type = boolean, values = {true, false}, default = true
- tol, κριτήριο ανοχής διακοπής
  for = both, type = numeric, values = float, default = 0.001
- cache_size, το μέγεθος μνήμης
  for = both, type = numeric, values = integer, default = 200
- class-weight, η μέθοδος ισορροπίας μεταξύ των κλάσεων των δειγμάτων
  for = classifier, type = numeric, values = {dict, balanced}, default = none
- verbose, το επίπεδο ανάλυσης της περιγραφής των εργασιών
  for = both, type = boolean, values = {true, false}, default = false
- max_iter, το όριο επαναλήψεων για την διακοπή εκπαίδευσης
  for = both, type = numeric, values = integer, default = -1
- decision_function_shape, η μέθοδος απόφασης
  for = both, type = categorical, values = {ovo,ovr}, default = ovr
- break_ties, το σπάσιμο δεσμών των κλάσεων
  for = both, type = boolean, values = {true, false}, default = false
- random_state, ο αριθμός τυχαιότητας του bootstrapping και της δειγματοληψίας
  for = both, type = numeric, values = integer, default = none

StohasticGradientDecent, διακρίνεται σε classifier (SGDClassifier) και regressor (SGDRegressor).

Είναι ένας μέτα-εκτιμητής ο οποίος εφαρμόζει στοχαστική διαδικασία για τη βελτιστοποίηση μιας αντικειμενικής συνάρτησης με κατάλληλες ιδιότητες ομαλότητας καθώς αντικαθιστά την πραγματική κλίση με μια εκτίμηση αυτής. Η στοχαστική κλίση καθόδου (συχνά συντομογραφία SGD) είναι μια επαναληπτική μέθοδος για τη βελτιστοποίηση μιας αντικειμενικής συνάρτησης με κατάλληλες ιδιότητες ομαλότητας (π.χ. διαφοροποιήσιμη ή υποδιαφορίσιμη). Μπορεί να θεωρηθεί ως μια στοχαστική προσέγγιση της βελτιστοποίησης gradient descent, καθώς αντικαθιστά την πραγματική κλίση (υπολογισμένη από ολόκληρο το σύνολο δεδομένων) με μια εκτίμηση αυτής (υπολογισμένη από ένα τυχαία επιλεγμένο υποσύνολο δεδομένων). Ειδικά σε προβλήματα βελτιστοποίησης υψηλών διαστάσεων, αυτό μειώνει τον υπολογιστικό φόρτο, επιτυγχάνοντας ταχύτερες επαναλήψεις στο εμπόριο για χαμηλότερο ποσοστό σύγκλισης.

Ενώ η βασική ιδέα πίσω από τη στοχαστική προσέγγιση μπορεί να ανιχνευθεί στον αλγόριθμο Robbins-Monro της δεκαετίας του 1950, η στοχαστική κλίση καθόδου έχει γίνει μια σημαντική μέθοδος βελτιστοποίησης στη μηχανική μάθηση.



Εικόνα 7. Αλγόριθμος StohasticGradientDecent

Στην παραπάνω εικόνα απεικονίζεται η λειτουργία του SVM όπου ένας επαναληπτικός αλγόριθμος, που ξεκινά από ένα τυχαίο σημείο μιας συνάρτησης και διανύει την κλίση της σε βήματα μέχρι να φτάσει στο χαμηλότερο σημείο αυτής της συνάρτησης και να εντοπίσει αυτό το σημείο που ελαχιστοποιεί κάποια συνάρτηση μετρικής αξιολόγησης.

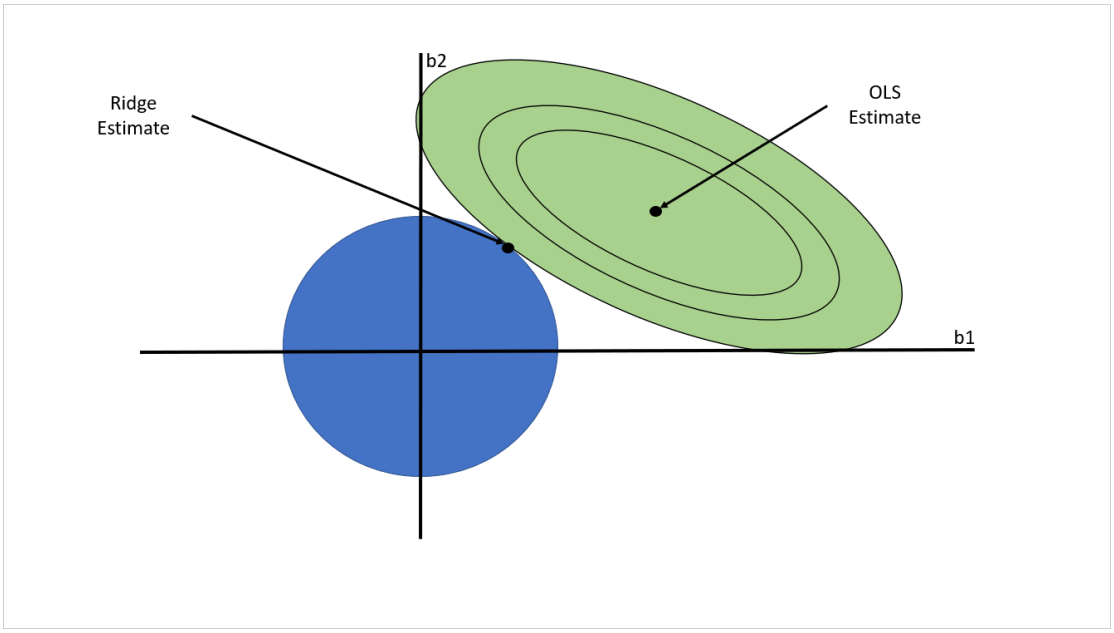
Οι υπερ-παράμετροι που χρησιμοποιούνται με την χρήση αυτής της μεθόδου:

- loss, η μετρική απώλειας
  for = classifier, type = categorical, values = {hinge,log,modified_huber,squared_hinge,perceptron}, default = hinge
  for = regressor, type = categorical, values = {squared_error,huber,epsilon_insensitive,squared_epsilon_insensitive}, default = squared_error
- penalty, η μέθοδος προστίμου εκτίμησης κόστους
  for = both, type = categorical, values = {l2,l1,elasticnet}
- alpha, η σταθερά που πολλαπλασιάζει τον όρο τακτοποίησης C
  for = both, type = numeric, values = float, default = 0.0001
- l1_ratio, ο συντελεστής κόστους
  for = both, type = categorical, values = [0.0:1.0], default = 0.15
- fit_intercept, η μέθοδος εφαρμογής των δειγμάτων κατά την εκπαίδευση
  for = both, type = boolean, values = [true,false], default = true
- max_iter, το όριο επαναλήψεων για την διακοπή εκπαίδευσης
  for = both, type = numeric, values = integer, default = -1
- tol, κριτήριο ανοχής διακοπής
  for = both, type = numeric, values = float, default = 0.001
- shuffle, επιλογή ανακατέματος στα δείγματα
  for = both, type = boolean, values = {true,false], default = true
- verbose, το επίπεδο ανάλυσης της περιγραφής των εργασιών
  for = both, type = numeric, values = integer, default = 0
- epsilon, η ευαισθησία στην μετρική κόστους
  for = both, type = numeric, values = float, default = 0.1
- n_jobs, ο αριθμός των εργασιών προς παράλληλη εκτέλεση
  for = classifier, type = numeric, values = integer, default = none(1)
- random_state, ο αριθμός τυχαιότητας της δειγματοληψίας
  for = both, type = numeric, values = integer, default = none
- learning_rate, ο ρυθμός μάθησης
  for = both, type = categorical, values = {constant,optimal,invscaling,adaptive}, default = optimal
- eta0, ο αρχικός ρυθμός μάθησης
  for = both, type = categorical, values = float, default = 0.0
- power_t, ο εκθέτης του ρυθμού μάθησης αντίστροφης κλίμακας
  for = both, type = categorical, values = float, default = 0.5
- early_stopping, η μέθοδος τερματισμού εκμάθησης
  for = both, type = categorical, values = {true,false}, default = false
- validation_fraction, το ποσοστό δεδομένων εκπαίδευσης για το την διακοπή εκπαίδευσης
  for = both, type = categorical, values = float, default = 0.1
- n_iter_no_change, ο αριθμός επαναλήψεων χωρίς βελτίωση για την διακοπή εκπαίδευσης
  for = both, type = categorical, values = integer, default = 5
- class_weight, τα βάρη συμετοχής των κλάσεων στην μέθοδο πρόβλεψης
  for = classifier, type = categorical, values = {dict[class:weight], balanced}, default = none
- warm_start, χρήση υπάρχοντών εφαρμογών απο προηγούμενες επαναλήψεις
  for = both, type = boolean, values = [true,false], default = false
- average, χρήση υπάρχοντών εφαρμογών απο προηγούμενες επαναλήψεις
  for = both, type = boolean/numeric, values = {true,false}/[1:+00], default = false

Ridge, διακρίνεται σε classifier (RidgeClassifier) και regressor (Ridge).

Είναι ένας μέτα-εκτιμητής ο οποίος εκτιμά τους συντελεστές πολλαπλής παλινδρόμησης σε σενάρια όπου οι γραμμικά ανεξάρτητες μεταβλητές έχουν υψηλή συσχέτιση. Το Ridge μετατρέπει τα δεδομένα σε [-1, 1] και λύνει το πρόβλημα με τη μέθοδο παλινδρόμησης. Η υψηλότερη τιμή στην πρόβλεψη γίνεται αποδεκτή ως κλάση στόχος και για δεδομένα πολλαπλών κλάσεων εφαρμόζεται παλινδρόμηση πολλαπλής εξόδου. Αυτή είναι μια μέθοδος συντονισμού μοντέλων που χρησιμοποιείται για την ανάλυση τυχόν δεδομένων που πάσχουν από πολυ-συγγραμμικότητα. Αυτή η μέθοδος εκτελεί τακτοποίηση L2. Η τακτοποίηση L2 λειτουργεί σαν μια δύναμη που αφαιρεί ένα μικρό ποσοστό βαρών σε κάθε επανάληψη. Όταν παρουσιάζεται το ζήτημα της πολυ-συγγραμμικότητας, τα ελάχιστα τετράγωνα είναι αμερόληπτα και οι διακυμάνσεις είναι μεγάλες, με αποτέλεσμα οι προβλεπόμενες τιμές να είναι πολύ μακριά από τις πραγματικές τιμές. Ιδιαίτερα χρήσιμο είναι να μετριαστεί το πρόβλημα της πολυ-συγγραμμικότητας στη γραμμική παλινδρόμηση, το οποίο εμφανίζεται συνήθως σε μοντέλα με μεγάλο αριθμό παραμέτρων.

Η τακτοποίηση Tikhonov, που πήρε το όνομά της από τον Andrey Tikhonov. Είναι μια μέθοδος γνωστή και ως παλινδρόμηση κορυφογραμμών. Γενικά, η μέθοδος παρέχει βελτιωμένη αποτελεσματικότητα σε προβλήματα εκτίμησης παραμέτρων σε αντάλλαγμα για ένα ανεκτό ποσό μεροληψίας (βλ. συμβιβασμό μεροληψίας-διακύμανσης).



Εικόνα 8. Αλγόριθμος Ridge

Στην παραπάνω εικόνα απεικονίζεται η λειτουργία του Ridge όπου με την χρήση της μια σταθερής παραμέτρου λάμδα (όρος ποινής), ελέγχεται η ποινή. Η ποινή είναι λάμδα επί το άθροισμα των τετραγώνων των συντελεστών. Όσο υψηλότερες είναι οι τιμές του λάμδα, τόσο μεγαλύτερη είναι η ποινή και επομένως το μέγεθος των συντελεστών μειώνεται. Οι ισοϋψείς ελλείψεις απεικονίζουν το άθροισμα των τετραγώνων των υπολοίπων RSS (Residual Sum of Squares) που καλούμαστε να ελαχιστοποιήσουμε ως προς τις παραμέτρους β.

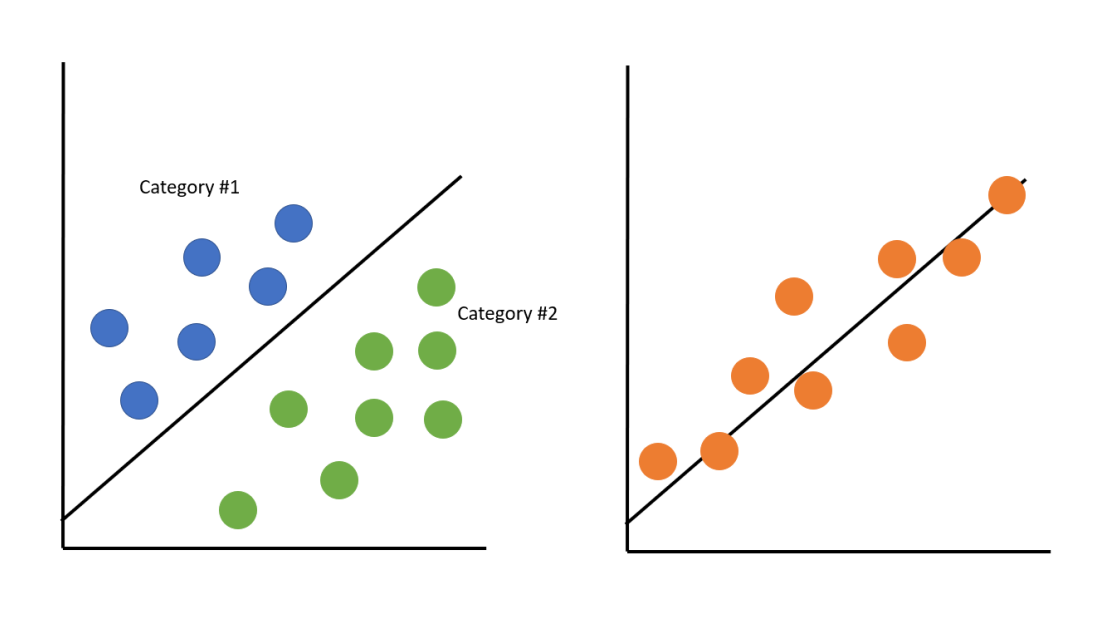
Οι υπερ-παράμετροι που χρησιμοποιούνται με την χρήση αυτής της μεθόδου:

- alpha, η σταθερά που πολλαπλασιάζει τον όρο τακτοποίησης C
  for = both, type = numeric, values = float, default = 1
- fit_intercept, η μέθοδος εφαρμογής των δειγμάτων κατά την εκπαίδευση
  for = both, type = boolean, values = {true,false}, default = true
- normalize, η μέθοδος κανονικοποίησης των δεδομένων κατά την εκπαίδευση
  for = both, type = boolean, values = {true,false}, default = true
- copy_X, η μέθοδος ανάγνωσης του συνόλου X με αντιγραφή ή επανεγγραφή
  for = both, type = boolean, values = {true,false}, default = true
- n_jobs, ο αριθμός των εργασιών προς παράλληλη εκτέλεση
  for = classifier, type = numeric, values = integer, default = none(1)
- positive, η μέθοδος συντελεστών, με θετικούς συντελεστές ή όχι
  for = both, type = boolean, values = {true,false}, default = false
- max_iter, το όριο επαναλήψεων για την διακοπή εκπαίδευσης
  for = both, type = numeric, values = integer, default = none
- tol, κριτήριο ανοχής διακοπής της εκπαίδευσης
  for = both, type = numeric, values = float, default = 0.001
- class_weight, τα βάρη συμμετοχής των κλάσεων στην μέθοδο πρόβλεψης
  for = classifier, type = categorical, values =
  {dict[class_label:weight], balanced}, default = none
- solver, τα βάρη συμμετοχής των κλάσεων στην μέθοδο πρόβλεψης
  for = classifier, type = categorical, values =
  {auto,svd,cholesky,lsqr,sparse_cg,sag,saga,lbfgs}, default = auto
- random_state, ο αριθμός τυχαιότητας της δειγματοληψίας
  for = both, type = numeric, values = integer, default = none

LinearModel, διακρίνεται σε classifier (LogisticRegression) και regressor (LinearRegression).

Είναι ένας μέτα-εκτιμητής ο οποίος εφαρμόζει γραμμικούς διαχωρισμούς για προκειμένου να διαχωρίσει το σύνολο δειγμάτων εκπαίδευσης γραμμικά. Η γραμμικότητα αναφέρεται στη γραμμικότητα των συντελεστών πρόβλεψης. Τα γραμμικά μοντέλα ερευνούν μια γραμμική σχέση μεταξύ των μεταβλητών εισόδου (x) και της μεταβλητής εξόδου (y). Πιο συγκεκριμένα, ότι το y μπορεί να υπολογιστεί από έναν γραμμικό συνδυασμό των μεταβλητών εισόδου (x). Το μοντέλο αυτό πρακτικά μπορεί να αναπαρασταθεί γραφικά σαν μια ευθεία η οποία προσπαθεί να διαχωρίσει τα δεδομένα (classification) ή να προσαρμοστεί σε αυτά (regression).

Οι σχέσεις μοντελοποιούνται χρησιμοποιώντας γραμμικές συναρτήσεις πρόβλεψης των οποίων οι άγνωστες παράμετροι του μοντέλου εκτιμώνται από τα δεδομένα. Τέτοια μοντέλα ονομάζονται γραμμικά μοντέλα. Συνηθέστερα, ο υπό όρους μέσος όρος της απόκρισης με δεδομένες τις τιμές των επεξηγηματικών μεταβλητών (ή προβλέψεων) θεωρείται ότι είναι μια συγγενική συνάρτηση αυτών των τιμών. Σπανιότερα, χρησιμοποιείται η υπό όρους διάμεσος ή κάποιο άλλο ποσό.



Εικόνα 9. Αλγόριθμος LinearModel

Στην παραπάνω εικόνα απεικονίζεται η λειτουργία του LinearModel όπου έχει εφαρμοστεί η «γραμμή της καλύτερου διαχωρισμού» (classification) και η «γραμμή της καλύτερης εφαρμογής» (regression). Η γραμμή εκφράζεται φυσικά με τον γενικό τύπο y = m(x) + b, τον τύπο για μια ευθεία γραμμή.

Οι υπερ-παράμετροι που χρησιμοποιούνται με την χρήση της μεθόδου LogisticRegression:

- penalty, η μέθοδος προστίμου εκτίμησης κόστους
  for = both, type = categorical, values = {none, l2, l1, elasticnet}, default = l2
- dual, η μέθοδος σύνθεσης
  for = both, type = boolean, values = {true,false}, default = false
- tol, κριτήριο ανοχής διακοπής της εκπαίδευσης
  for = both, type = numeric, values = float, default = 0.0001
- C, το βάρος τακτοποίησης
  for = both, type = numeric, values = float, default = 1
- fit_intercept, η μέθοδος εφαρμογής των δειγμάτων κατά την εκπαίδευση
  for = both, type = boolean, values = {true,false}, default = true
- intercept_scaling, η μέθοδος εφαρμογής των δειγμάτων κατά την εκπαίδευση
  for = both, type = numeric, values = float, default = 1
- class_weight, τα βάρη συμετοχής των κλάσεων στην μέθοδο πρόβλεψης
  for = classifier, type = categorical, values = {dict[class_label:weight], balanced}, default = none
- random_state, ο αριθμός τυχαιότητας της δειγματοληψίας
  for = both, type = numeric, values = integer, default = none
- solver, τα βάρη συμετοχής των κλάσεων στην μέθοδο πρόβλεψης
  for = classifier, type = categorical, values = {newton-cg,lbfgs,liblinear,sag,saga}, default = lbfgs
- max_iter, το όριο επαναλήψεων για την διακοπή εκπαίδευσης
  for = both, type = numeric, values = integer, default = none
- multiclass, η μέθοδος διενέργιας του αλλγορίθμου ανα τύπο προβλήματος
  for = both, type = categorical, values = {auto,ovr,multinomial}, default = auto
- verbose, το επίπεδο ανάλυσης της περιγραφής των εργασιών
  for = both, type = numeric, values = integer, default = 0
- warm_start, χρήση υπάρχοντών εφαρμογών απο προηγούμενες επαναλήψεις
  for = both, type = boolean, values = {true,false}, default = false
- n_jobs, ο αριθμός των εργασιών προς παράλληλη εκτέλεση
  for = both, type = numeric, values = integer, default = none(1)
- l1_ratio, ο συντελεστής κόστους
  for = both, type = categorical, values = [0.0:1.0], default = none

Οι υπερ-παράμετροι που χρησιμοποιούνται με την χρήση της μεθόδου LinearRegression:

- fit_intercept, η μέθοδος εφαρμογής των δειγμάτων κατά την εκπαίδευση
  for = both, type = boolean, values = {true,false}, default = true
- normalize, η μέθοδος κανονικοποίησης των δεδομένων κατά την εκπαίδευση
  for = both, type = boolean, values = {true,false}, default = true
- copy_X, η μέθοδος ανάγνωσης του συνόλου X με αντιγραφή ή επανεγγραφή
  for = both, type = boolean, values = {true,false}, default = true
- n_jobs, ο αριθμός των εργασιών προς παράλληλη εκτέλεση
  for = both, type = numeric, values = integer, default = none(1)
- positive, η μέθοδος συντελεστών, με θετικούς συντελεστές ή όχι
  for = both, type = boolean, values = {true,false}, default = false

4. Αυτόματη Μηχανική Μάθηση

Στην σημερινή εποχή η μηχανική μάθηση είναι γεγονός καθώς εμπίπτει ολοένα και περισσότερο σε πολύπλοκες εφαρμογές στην καθημερινότητα μας. Ένα νέο όμως πρόβλημα που δημιουργείται είναι οι τεχνικές και ο χρόνος που απαιτείται για να δουλέψει σωστά ένα μοντέλο μηχανικής μάθησης. Για αυτό τον λόγο γίνονται προσπάθειες αυτόματης μηχανικής μάθησης οι οποίες ενσωματώνουν νέες κορυφαίες τεχνικές από επιστήμονες του χώρου. Ο νέος λοιπόν στόχος στην μηχανική μάθηση είναι να αυτοματοποιηθούν δύσκολες αλλά και χρονοβόρες διαδικασίες όπως αυτές της προ-επεξεργασίας δεδομένων αλλά και αυτή της μοντελοποίησης δηλαδή της επιλογής κατάλληλων αλγορίθμων και υπερ-παραμέτρων.

Κατά το πέρας του χρόνου έχουν προταθεί αρκετές διαδικασίες για την επιλογή αλγορίθμου και την βέλτιστη επιλογή υπερ-παραμέτρων. Ωστόσο, τον τελευταίο καιρό έχουν αναπτυχθεί και διαδικασίες που αναλαμβάνουν και την περιοχή της προ-επεξεργασίας των δεδομένων καθώς είναι ένα αναπόσπαστο κομμάτι σε κάθε διαδικασία μηχανικής μάθησης και σίγουρα παίζει και αυτό καθοριστικό ρόλο στην απόδοση του μοντέλου.

Εικόνα 10. Φάσεις αυτόματης μηχανικής μάθησης

Με λίγα λόγια, σκοπός της αυτοματοποιημένης μηχανικής μάθησης είναι να παρέχει μεθόδους και διαδικασίες σε ανθρώπους που δεν είναι ειδικοί στη μηχανική μάθηση. Η μηχανική μάθηση αποτελεί σημαντικό κομμάτι πολλών εφαρμογών τα τελευταία χρόνια και ένας διαρκώς αυξανόμενος αριθμός επιστημονικών κλάδων βασίζεται στην χρήση αυτής (Jonathan Waring, et al., 2020).

Ωστόσο, αυτή η επιτυχία βασίζεται σε σημαντικό βαθμό στους ειδικούς για την εκτέλεση των ακόλουθων κύριων εργασιών με τις αυτόματες λύσεις μηχανικής μάθησης να προσπαθούν να προσομοιώσουν αυτές τις εργασίες:

- Προ-επεξεργασία δεδομένων
    - Επιλογή σημαντικών δειγμάτων
    - Επιλογή σημαντικών χαρακτηριστικών
    - Αντιμετώπιση αλφαριθμητικών τιμών
    - Αντιμετώπιση ελλιπών τιμών
- Μοντελοποίηση μηχανικής μάθησης
    - Επιλογή κατάλληλης οικογένειας μοντέλων
    - Βελτιστοποίηση υπερ-παραμέτρων μοντέλου
- Ανάλυση αποτελεσμάτων που προέκυψαν
    - Αποτελέσματα προβλέψεων
    - Μετρικές αξιολόγησης

4.1 Τεχνικές Επιλογής Μηχανικής Μάθησης

Όπως έχει ήδη προαναφερθεί ξεκινώντας από τα πεδία μηχανικής μάθησης, συνεχίζοντας στις κατηγορίες μηχανικής μάθησης με στόχο την επιλογή του κατάλληλου αλγορίθμου μηχανικής μάθησης, ο μεγάλος αριθμός υπερ-παραμέτρων καθιστά δύσκολη την επιλογή και την παραμετροποίηση της εκάστοτε διαδικασίας. Τα μοντέλα μηχανικής εκμάθησης έχουν υπερ-παραμέτρους που πρέπει να οριστούν για να προσαρμοστεί το μοντέλο στο σύνολο δεδομένων. Συχνά τα γενικά αποτελέσματα των υπερ-παραμέτρων σε ένα μοντέλο είναι γνωστά, αλλά ο καλύτερος τρόπος ρύθμισης μιας υπερ-παραμέτρου και συνδυασμών υπερ-παραμέτρων που αλληλοεπιδρούν για ένα δεδομένο σύνολο δεδομένων είναι δύσκολος. Δεδομένου των παραπάνω, υπάρχουν εμπειρικοί κανόνες για τη διαμόρφωση των υπερ-παραμέτρων. Μια καλύτερη προσέγγιση είναι η αντικειμενική αναζήτηση διαφορετικών τιμών για τις υπερ-παραμέτρους του μοντέλου και η επιλογή ενός υποσυνόλου που οδηγεί σε ένα μοντέλο που επιτυγχάνει την καλύτερη απόδοση σε ένα δεδομένο σύνολο δεδομένων. Αυτό ονομάζεται βελτιστοποίηση υπερ-παραμέτρων ή συντονισμός υπερ-παραμέτρων. Το αποτέλεσμα αυτής είναι ένα ενιαίο σύνολο υπερ-παραμέτρων με καλή απόδοση που μπορεί να χρησιμοποιηθεί για να διαμορφωθεί το μοντέλο.

Υπάρχουν διάφορες διαδικασίες ώστε να γίνει εύρεση του κατάλληλου αλγορίθμου ή και των υπερ-παραμέτρων για ένα συγκεκριμένο σύνολο δεδομένων (Petro Liashchynskyi, et al., 2019).

- Δεδομένου ενός συνόλου μπορούν να δοκιμαστούν όλοι οι πιθανοί συνδυασμοί αλγορίθμων ή και υπερ-παραμέτρων, η τεχνική αυτή ονομάζεται GridSearchCV, η τεχνική αυτή εγγυάται πως θα βρει τον καλύτερο συνδυασμό αλλά θα πρέπει να τους δοκιμάσει όλους, κάτι το οποίο θα αποτελέσει μια χρονοβόρα διαδικασία.
- Δεδομένου ενός συνόλου μπορούν να δοκιμαστούν τυχαία κάποιοι πιθανοί συνδυασμοί αλγορίθμων ή και υπερ-παραμέτρων, η τεχνική αυτή ονομάζεται RandomSearchCV, η τεχνική αυτή εγγυάται πως θα βρει έναν ικανοποιητικό συνδυασμό ανάμεσα σε τυχαίους ελέγχους κάτι πιο γρήγορο από την προηγούμενη τεχνική αλλά ίσως κάποιοι συνδυασμοί δεν δοκιμαστούν ποτέ χωρίς να αποφέρει το βέλτιστο αποτέλεσμα.
- Δεδομένου ενός συνόλου μπορούν να δοκιμαστούν συγκεκριμένοι συνδυασμοί αλγορίθμων ή και υπερ-παραμέτρων με την επιλογή να εξαρτάται από την προηγούμενη τιμή της μετρικής η οποία εξετάζεται. Η τεχνική αυτή ονομάζεται Bayesian Search και δύναται να αποφέρει το βέλτιστο αποτέλεσμα με μεγάλη πιθανότητα χωρίς να δοκιμάσει όλους τους συνδυασμούς. Αυτές οι τεχνικές ανήκουν σε μια γενικότερη μεθοδολογία που ονομάζεται Sequential Model-based Bayesian Optimization (SMBO). Υπό το πρίσμα αυτής της μεθοδολογίας πολλές τεχνικές έχουν προταθεί όπως το Sequential Model Based Algorithm Configuration (SMAC) το οποίο συνδυάζει δενδροειδείς αλγορίθμους (random forest) και το Tree-Structured Parzen Estimetors (TPE) το οποίο λαμβάνει υπόψιν στατιστικά πυκνότητας.
- Δεδομένου ενός συνόλου μπορούν να δοκιμαστούν συγκεκριμένοι συνδυασμοί αλγορίθμων ή και υπερ-παραμέτρων σύμφωνα με κάποια χαρακτηριστικά του συνόλου όπως ο αριθμός χαρακτηριστικών, ο αριθμός των συνεχών τιμών ή των τιμών των κλάσεων, όπου συγκεκριμένοι αλγόριθμοι είναι πιο πιθανό να εξάγουν ένα καλύτερο αποτέλεσμα, η τεχνική αυτή ονομάζεται MetaSearch και στηρίζεται σε ένα προ-εκπαιδευμένο σύστημα δοκιμών.
- Δεδομένου ενός συνόλου μπορούν να δοκιμαστούν διάφορες τεχνικές για την εύρεση υπερ-παραμέτρων με διάφορους όχι τόσο δημοφιλής τρόπους όπως το Aggressive Racing (ROAR) ή το Deep Network for Global Optimization (DNGO) και όπου το καθένα έχει την δική του φιλοσοφία για την οργάνωση και εύρεση υπερ-παραμέτρων.

4.2 Υπερπαράμετροι Τεχνικών Επιλογής Μηχανικής Μάθησης

Όπως έχει ήδη προαναφερθεί υπάρχουν αρκετές τεχνικές επιλογής αλγορίθμων και υπερ-παραμέτρων μηχανικής μάθησης οι οποίες παρουσιάζουν πλεονεκτήματα και μειονεκτήματα (Li Yang, et al., 2020).

Οι παραπάνω τεχνικές έχουν συγκεκριμένες απαιτήσεις σε συγκεκριμένη δομή προκειμένου να είναι λειτουργικές:

- Τον αλγόριθμο μηχανικής μάθησης
- Τον χώρο υπερ-παραμέτρων του εκάστοτε αλγορίθμου
- Την μέθοδο αναζήτησης στον χώρο των υπερ-παραμέτρων
- Τον τρόπο διαχωρισμού των δειγμάτων στο σύνολο δεδομένων
- Την μετρική προς βελτιστοποίηση του εκάστοτε αλγορίθμου

Είναι λοιπόν σαφές πως ο κάθε αλγόριθμος έχει συγκεκριμένες υπερ-παραμέτρους οι οποίες φαίνεται να παρουσιάζουν ομοιότητες αν βρίσκονται στην ίδια οικογένεια. Παρόλα αυτά πρέπει να δοθεί απαραίτητη προσοχή στις διαθέσιμες υπερ-παραμέτρους, στο εύρος των διαθέσιμων τιμών αλλά και στην μεταξύ τους εξάρτηση.

Για παράδειγμα, ο αλγόριθμος μηχανικής μάθησης Random Forest δέχεται 18 υπερ-παραμέτρους εκ-των οποίων:

- οι δύο δεν χρειάζονται αναζήτηση, η n_jobs η οποία θέτει τα threads και η random_state η οποία καθορίζει την τυχαιότητα της numpy.
- μια έχει εξάρτηση, η bootstrap, η οποία όταν είναι True, η oob_score μπορεί να πάρει μόνο την τιμή False ενώ όταν η bootstrap είναι False μπορεί να πάρει τιμές [False, True]

Επομένως, θα πρέπει να ορισθεί ένα συγκεκριμένο σύνολο υπερ-παραμέτρων για τις οποίες υπάρχει νόημα αναζήτησης σε λογικό εύρος και με την κατάλληλη εξάρτηση αλλιώς οι πιθανοί συνδυασμοί θα οδηγήσουν σε ανεξέλεγκτα αποτελέσματα χρόνου αλλά και απόδοσης.

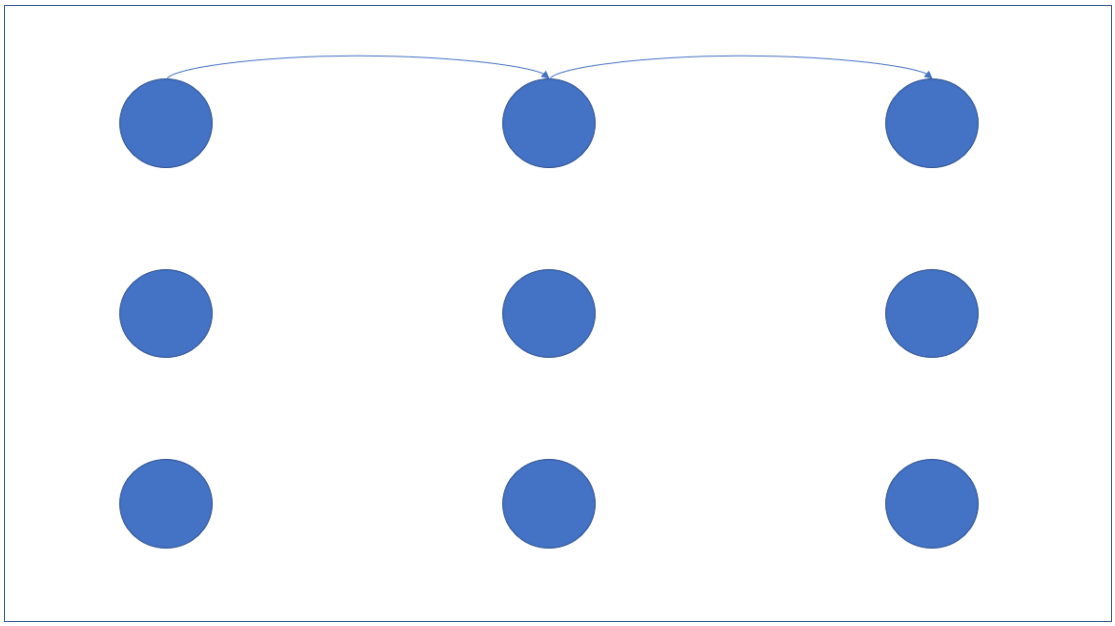
Τέλος είναι σημαντικό οι εξαρτήσεις να μπορούν να αποτυπωθούν σαν μια μέθοδος η οποία εμπεριέχει if else κανόνες και όχι η αποτύπωση τους να γίνεται ως αναπαραγωγή του χώρου αναζήτησης με τις απαραίτητες αλλαγές καθώς η κατανόηση του χώρου θα γίνει δύσκολη.

Για παράδειγμα, η πολλαπλή έκφραση για το random forest:
space → [{ bootstrap: True, oob_score: False}, { bootstrap: False, oob_score: True, False}]

Ενώ θα μπορούσε να γίνει με ποιο ομοιόμορφο τρόπο όπως παρακάτω:
space → [{ bootstrap: True / False, oob_score: True / False}]
config →{when bootstrap: True then oob_score: False }

Έτσι η επιλογή της διαδικασίας αναζήτησης δεν έχει να κάνει μόνο με τον τρόπο αναζήτησης των υπερ-παραμέτρων αλλά και την ικανότητα οργάνωσης των υπερ-παραμέτρων με συγκεκριμένο τρόπο κατάλληλο και καλύπτοντας πολλαπλά προ-απαιτούμενα.

GridSearchCV, είναι μία μέθοδος αναζήτησης υπερ-παραμέτρων σε ένα δεδομένο χώρο με επαναληπτικό τρόπο. Ο διαχωρισμός των δεδομένων γίνεται με cross-validation. Η μέθοδος θα αναζητήσει τις καλύτερες υπερ-παραμέτρους ώστε να εξάγει το βέλτιστο αποτέλεσμα για την επιλεγμένη μετρική δοκιμάζοντας όλους τους πιθανούς συνδυασμούς.

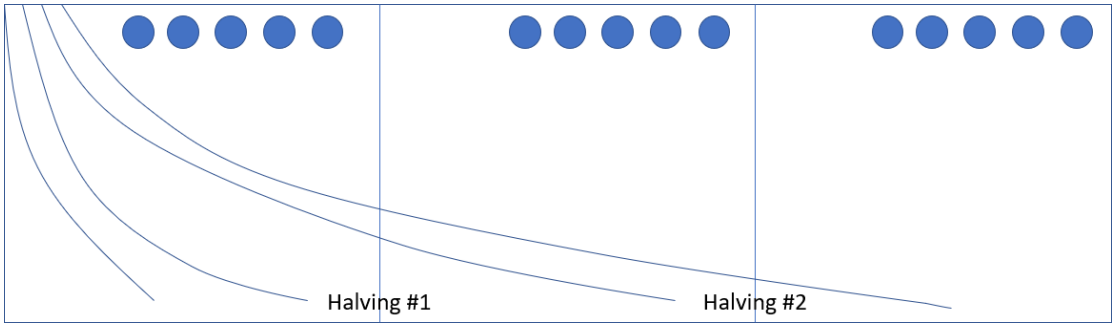


Εικόνα 11. Μέθοδος Αναζήτησης Grid Search CV

Οι υπερ-παράμετροι που χρησιμοποιούνται με την χρήση αυτής της μεθόδου:

- estimator, ο αλγόριθμος μηχανικής μάθησης
- param_ grid, ο χώρος αναζήτησης των υπερ-παραμέτρων
- scoring, η μετρική αξιολόγησης των προβλέψεων
  type = categorical, values = {}, default = none
- n_jobs, ο αριθμός των εργασιών προς παράλληλη εκτέλεση
  type = numeric, values = integer, default = none(1)
- refit, η τεχνική επιλογής του καλύτερου χώρου υπερ-παραμέτρων
  type = boolean, values = {true,false}, default = true
- cv, ο αριθμός των κουβάδων του cross-validation
  type = numeric, values = integer, default = 5
- verbose, το επίπεδο ανάλυσης της περιγραφής των εργασιών
  type = boolean, values = {true,false}, default = false
- pre_dispatch, η μέθοδος οργάνωσης των εργασιών
  type = categorical/numeric, values = {true,false}, default = false
- error_score, η τεχνική αντιμετώπισης ενός σφάλματος
  type = categorical/numeric, values = {raise,numeric}, default = none
- return_train_score, η επιστροφή αποτελεσμάτων του σκορ της εκπαίδευσης
  type = boolean, values = {true,false}, default = false

HalvingGridSearchCV, είναι μία μέθοδος αναζήτησης υπερ-παραμέτρων σε ένα δεδομένο χώρο με επαναληπτικό τρόπο που όμως είναι ενισχυμένη με διαδικασίες sampling ώστε να βελτιωθεί η απόδοση (successive halving). Ο διαχωρισμός των δεδομένων γίνεται με cross-validation. Η μέθοδος θα αναζητήσει τις καλύτερες υπερ-παραμέτρους ώστε να εξάγει το βέλτιστο αποτέλεσμα για την επιλεγμένη μετρική δοκιμάζοντας όλους τους πιθανούς συνδυασμούς. Η μέθοδος halving θα χωρίσει την διαδικασία σε στάδια ενώ σε κάθε στάδιο θα γίνεται ψηφοφορία προκειμένου να επιλεχθούν οι αλγόριθμοι που θα προχωρήσουν.

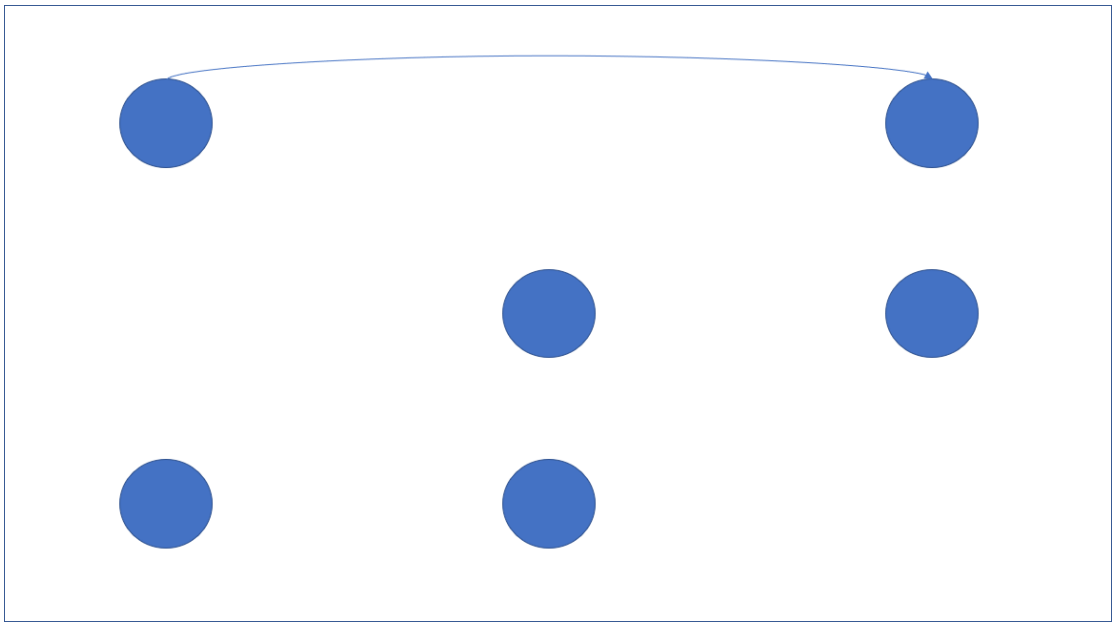


Εικόνα 12. Μέθοδος Αναζήτησης Halving Grid Search CV

Οι υπερ-παράμετροι που χρησιμοποιούνται με την χρήση αυτής της μεθόδου:

- estimator, ο αλγόριθμος μηχανικής μάθησης
- param_grid, ο χώρος αναζήτησης των υπερ-παραμέτρων
- factor, ο αριθμός των υποψήφιων παραμέτρων σε κάθε επόμενη επανάληψη
  type = numeric, values = float, default = 3
- resource, ο τύπος πόρων προς αύξηση σε κάθε επανάληψη
  type = categorical, values = {n_samples,n_iterations,n_estimators}, default = n_samples
- max_resources, ο μέγιστος αριθμός των πόρων του κάθε υποψηφίου σε κάθε επανάληψη
  type = numeric, values = integer, default = auto
- min_resources, ο ελάχιστος αριθμός των πόρων του κάθε υποψηφίου σε κάθε επανάληψη
  type = numeric, values = integer, default = smallest
- aggressive_elimination, η μέθοδος κουρέματος των υποψηφίων όταν δεν υπάρχουν πόροι
  type = boolean, values = {true,false}, default = false
- cv, ο αριθμός των κουβάδων του cross-validation
  type = numeric, values = integer, default = 5
- scoring, η μετρική αξιολόγησης των προβλέψεων
  type = categorical, values = {}, default = none
- refit, η τεχνική επιλογής του καλύτερου χώρου υπερ-παραμέτρων
  type = boolean, values = {true,false}, default = true
- error_score, η τεχνική αντιμετώπισης ενός σφάλματος
  type = categorical/numeric, values = {raise\numeric}, default = none
- return_train_score, η επιστροφή αποτελεσμάτων του σκορ της εκπαίδευσης
  type = boolean, values = {true,false}, default = false
- random_state, ο αριθμός τυχαιότητας της δειγματοληψίας
  type = numeric, values = integer, default = none
- n_jobs, ο αριθμός των εργασιών προς παράλληλη εκτέλεση
  type = numeric, values = integer, default = none(1)
- verbose, το επίπεδο ανάλυσης της περιγραφής των εργασιών
  type = boolean, values = {true,false}, default = false

RandomizedSearchCV, είναι μία μέθοδος αναζήτησης υπερ-παραμέτρων σε ένα δεδομένο χώρο με τυχαίο επαναληπτικό τρόπο. Ο διαχωρισμός των δεδομένων γίνεται με cross-validation. Η μέθοδος θα αναζητήσει τις καλύτερες υπερ-παραμέτρους ώστε να εξάγει το βέλτιστο αποτέλεσμα για την επιλεγμένη μετρική δοκιμάζοντας τυχαίες τιμές κάθε φορά παραβλέποντας κάποιες.

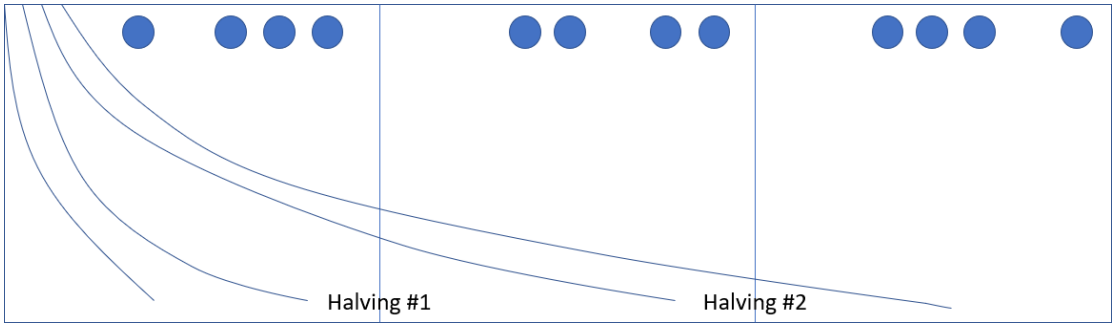


Εικόνα 13. Μέθοδος Αναζήτησης Random Search CV

Οι υπερ-παράμετροι που χρησιμοποιούνται με την χρήση αυτής της μεθόδου:

- estimator, ο αλγόριθμος μηχανικής μάθησης
- param_distributions, ο χώρος αναζήτησης των υπερ-παραμέτρων
- n_iter, ο αριθμός δειγματοληψίας των παραμέτρων
  type = numeric, values = integer, default = 10
- scoring, η μετρική αξιολόγησης των προβλέψεων
  type = categorical, values = {}, default = none
- n_jobs, ο αριθμός των εργασιών προς παράλληλη εκτέλεση
  type = numeric, values = integer, default = none(1)
- refit, η τεχνική επιλογής του καλύτερου χώρου υπερ-παραμέτρων
  type = boolean, values = {true,false}, default = true
- cv, ο αριθμός των κουβάδων του cross-validation
  type = numeric, values = integer, default = 5
- verbose, το επίπεδο ανάλυσης της περιγραφής των εργασιών
  type = boolean, values = {true,false}, default = false
- pre_dispatch, η μέθοδος οργάνωσης των εργασιών
  type = categorical/numeric, values = {true,false}, default = false
- random_state, ο αριθμός τυχαιότητας της δειγματοληψίας
  type = numeric, values = integer, default = none
- error_score, η τεχνική αντιμετώπισης ενός σφάλματος
  type = categorical/numeric, values = {raise\numeric}, default = none
- return_train_score, η επιστροφή αποτελεσμάτων του σκορ της εκπαίδευσης
  type = boolean, values = {true,false}, default = false

HalvingRandomSearchCV, είναι μία μέθοδος αναζήτησης υπερ-παραμέτρων σε ένα δεδομένο χώρο με τυχαίο επαναληπτικό τρόπο που όμως είναι ενισχυμένη με διαδικασίες sampling ώστε να βελτιωθεί η απόδοση (successive halving). Ο διαχωρισμός των δεδομένων γίνεται με cross-validation. Η μέθοδος θα αναζητήσει τις καλύτερες υπερ-παραμέτρους ώστε να εξάγει το βέλτιστο αποτέλεσμα για την επιλεγμένη μετρική δοκιμάζοντας τυχαίες τιμές παραβλέποντας κάποιες. Η μέθοδος halving θα χωρίσει την διαδικασία σε στάδια ενώ σε κάθε στάδιο θα γίνεται ψηφοφορία προκειμένου να επιλεχθούν οι αλγόριθμοι που θα προχωρήσουν.

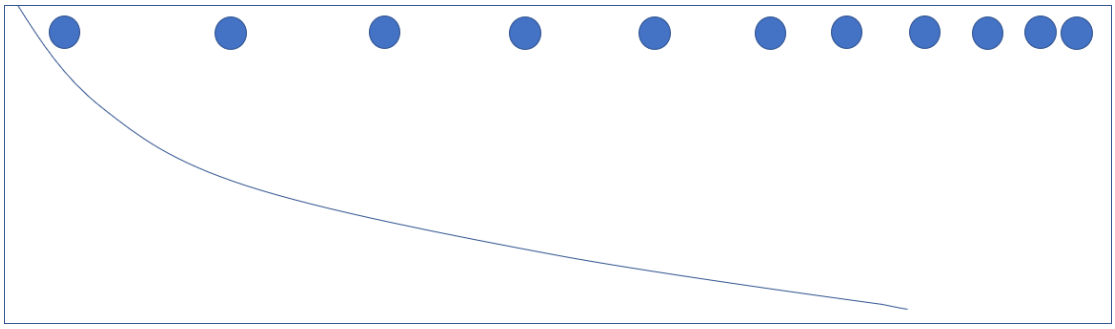


Εικόνα 14. Μέθοδος Αναζήτησης Halving Random Search CV

Οι υπερ-παράμετροι που χρησιμοποιούνται με την χρήση αυτής της μεθόδου:

- estimator, ο αλγόριθμος μηχανικής μάθησης
- param_distributions, ο χώρος αναζήτησης των υπερ-παραμέτρων
- n_candidates, ο αριθμός των υποψήφιων παραμέτρων στην πρώτη επανάληψη
  type = numeric, values = integer, default = exhaust
- factor, ο αριθμός των υποψήφιων παραμέτρων σε κάθε επόμενη επανάληψη
  type = numeric, values = float, default = 3
- max_resources, ο μέγιστος αριθμός των πόρων του κάθε υποψηφίου σε κάθε επανάληψη
  type = numeric, values = integer, default = auto
- min_resources, ο ελάχιστος αριθμός των πόρων του κάθε υποψηφίου σε κάθε επανάληψη
  type = numeric, values = integer, default = smallest
- aggressive_elimination, η μέθοδος κουρέματος των υποψηφίων όταν δεν υπάρχουν πόροι
  type = boolean, values = {true,false}, default = false
- cv, ο αριθμός των κουβάδων του cross-validation
  type = numeric, values = integer, default = 5
- scoring, η μετρική αξιολόγησης των προβλέψεων
  type = categorical, values = {}, default = none
- refit, η τεχνική επιλογής του καλύτερου χώρου υπερ-παραμέτρων
  type = boolean, values = {true,false}, default = true
- error_score, η τεχνική αντιμετώπισης ενός σφάλματος
  type = categorical/numeric, values = {raise\numeric}, default = none
- return_train_score, η επιστροφή αποτελεσμάτων του σκορ της εκπαίδευσης
  type = boolean, values = {true,false}, default = false
- random_state, ο αριθμός τυχαιότητας της δειγματοληψίας
  type = numeric, values = integer, default = none
- n_jobs, ο αριθμός των εργασιών προς παράλληλη εκτέλεση
  type = numeric, values = integer, default = none(1)
- verbose, το επίπεδο ανάλυσης της περιγραφής των εργασιών
  type = boolean, values = {true,false}, default = false

BayesSearchCV, είναι μία μέθοδος αναζήτησης υπερ-παραμέτρων σε ένα δεδομένο χώρο με επαναληπτικό τρόπο που όμως είναι ενισχυμένη με στοχαστικές διαδικασίες (Bayes). Ο διαχωρισμός των δεδομένων γίνεται με cross-validation. Η μέθοδος θα αναζητήσει τις καλύτερες υπερ-παραμέτρους για την επιλεγμένη μετρική δοκιμάζοντας τιμές σε κάθε επανάληψη υπολογίζοντας την πιθανότητα βελτίωσης με βάση την προηγούμενη εμπειρία.



Εικόνα 15. Μέθοδος Αναζήτησης Bayes Search CV

Οι υπερ-παράμετροι που χρησιμοποιούνται με την χρήση αυτής της μεθόδου:

- estimator, ο αλγόριθμος μηχανικής μάθησης
- search_spaces, ο χώρος αναζήτησης των υπερ-παραμέτρων
- n_iter, ο αριθμός των υποψήφιων παραμέτρων στην πρώτη επανάληψη
  type = numeric, values = integer, default = 50
- optimizer_kwargs, παράμετροι για τον εκτιμητή
  type = dict, values = dict[str], default = none
- scoring, η μετρική αξιολόγησης των προβλέψεων
  type = categorical, values = {}, default = none
- fit_params, παράμετροι για την διαδικασία εκπαίδευσης
  type = dict, values = dict[{'base_estimator': 'RF'}], default = none
- n_jobs, ο αριθμός των εργασιών προς παράλληλη εκτέλεση
  type = numeric, values = integer, default = none(1)
- n_points, ο αριθμός των παραμέτρων προς παράλληλη εκτέλεση
  type = numeric, values = integer, default = none(1)
- pre_dispatch, ο αριθμός των παραμέτρων προς παράλληλη εκτέλεση
  type = none/int/string, values = {integer\'expression'}, default = none
- cv, ο αριθμός των κουβάδων του cross-validation
  type = numeric, values = integer, default = 5
- iid, σηματοδοτεί την ισορροπημένη κατανομή των δεδομένων στους κουβάδες
  type = boolean, values = {true,false}, default = true
- refit, η τεχνική επιλογής του καλύτερου χώρου υπερ-παραμέτρων
  type = boolean, values = {true,false}, default = true
- verbose, το επίπεδο ανάλυσης της περιγραφής των εργασιών
  type = boolean, values = {true,false}, default = false
- random_state, ο αριθμός τυχαιότητας της δειγματοληψίας
  type = numeric, values = integer, default = none
- error_score, η τεχνική αντιμετώπισης ενός σφάλματος
  type = categorical/numeric, values = {raise\numeric}, default = none
- return_train_score, η επιστροφή αποτελεσμάτων του σκορ της εκπαίδευσης
  type = boolean, values = {true,false}, default = false

4.3 Ολοκληρωμένες Λύσεις Επιλογής Μηχανικής Μάθησης

Όπως είναι ήδη εμφανές, με αρκετές τεχνικές επιλογής αλγορίθμου και υπερ-παραμέτρων ήταν ζήτημα χρόνου προτού προταθούν πιο ολοκληρωμένες λύσεις οι οποίες όχι μόνο έρχονται να βελτιώσουν ή αυτοματοποιήσουν τις ήδη προ-υπάρχουσες τεχνικές αλλά και να προσθέσουν το κομμάτι της προ-επεξεργασίας των δεδομένων προκειμένου να μπορούν να χρησιμοποιηθούν από τον οποιοδήποτε χωρίς να απαιτείται μεγάλη ανάλυση (Matthias Feurer, et al., 2020).

Υπάρχει ένας περιορισμένος αριθμός ολοκληρωμένων λύσεων αυτή την στιγμή εκ των οποίων κάποιες χρησιμοποιούνται αρκετό καιρό στον χώρο της αυτόματης μηχανικής μάθησης και άλλες οι οποίες ενσωματώνουν τεχνικές που βρίσκονται σε πειραματικό στάδιο.

- auto-sklearn, η οποία μπορεί να χρησιμοποιηθεί είτε για classification είτε για regression διαδικασίες. Η επιλογή αλγορίθμου είναι βασισμένη στο Meta Search (δηλαδή την επιλογή αλγορίθμου με βάση προηγούμενη εμπειρία) ενώ η βελτιστοποίηση υπερ-παραμέτρων είναι βασισμένη στο Bayesian Optimization (δηλαδή με την επιλογή των επόμενων συνδυασμών βάση της πιθανότητας βελτίωσης σε κάθε επανάληψη). Η συγκεκριμένη λύση εμπεριέχει και την δυνατότητα προ-επεξεργασίας των δεδομένων. Επιπλέον η λύση στοχεύει περισσότερο στην εκτέλεση της αναζήτησης εντός ενός χρονικού ορίου. Τέλος η λύση προσφέρεται μέσο του pypi καθώς είναι υλοποιημένη σε γλώσσα python (Matthias Feurer, et al., 2019).

- hp-sklearn, η οποία μπορεί να χρησιμοποιηθεί είτε για classification είτε για regression διαδικασίες. Η επιλογή αλγορίθμου είναι βασισμένη σε επαναληπτική διαδικασία ενώ η βελτιστοποίηση υπερ-παραμέτρων είναι βασισμένη στο Bayesian Optimization (δηλαδή με την επιλογή των επόμενων συνδυασμών βάση της πιθανότητας βελτίωσης σε κάθε επανάληψη). Η συγκεκριμένη λύση εμπεριέχει και την δυνατότητα προ-επεξεργασίας των δεδομένων. Επιπλέον η λύση στοχεύει περισσότερο στην εκτέλεση της αναζήτησης εντός ενός χρονικού ορίου για κάθε αλγόριθμο. Θα πρέπει να σημειωθεί πως η λύση αυτή είναι ένα περίβλημα της hyperopt η οποία μπορεί να ελαχιστοποιήσει μια συνάρτηση κόστους με παράλληλη επεξεργασία χρησιμοποιώντας spark ή mongo jobs αλλά δυστυχώς αναμένεται η υποστήριξη στο περίβλημα. Τέλος η λύση προσφέρεται μέσο του pypi καθώς είναι υλοποιημένη σε γλώσσα python (Brent Komer, et al., 2019).

- auto-weka, η οποία μπορεί να χρησιμοποιηθεί είτε για classification είτε για regression (regression - σε πειραματικό στάδιο) διαδικασίες. Η επιλογή αλγορίθμου είναι βασισμένη σε επαναληπτική διαδικασία ενώ η βελτιστοποίηση υπερ-παραμέτρων είναι βασισμένη στο Bayesian Optimization (δηλαδή με την επιλογή των επόμενων συνδυασμών βάση της πιθανότητας βελτίωσης σε κάθε επανάληψη) αλλά και διάφορες Random (δηλαδή τυχαίες) ή Meta (δηλαδή την επιλογή υπερ-παραμέτρων με βάση προηγούμενη εμπειρία) τεχνικές ανάλογα τον αλγόριθμο επιλογής. Η συγκεκριμένη λύση εμπεριέχει και την δυνατότητα προ-επεξεργασίας των δεδομένων. Επιπλέον η λύση στοχεύει περισσότερο στην εκτέλεση της αναζήτησης εντός ενός χρονικού ορίου. Η λύση αυτή προσφέρεται μέσα από γραφικό περιβάλλον το οποίο είναι γραμμένα σε java εδώ δεν υπάρχει η αντίστοιχη python υλοποίηση α η οποία να ακολουθεί τις νέες ενημερώσεις της autoWeka. (Lars Kotthoff, και συν., 2019).

4.4 Υπερπαράμετροι Ολοκληρωμένων Λύσεων Επιλογής Μηχανικής Μάθησης

Χρησιμοποιώντας κάποιες από τις ολοκληρωμένες λύσεις μηχανικής μάθησης αρκετά προβλήματα όπως η προ-επεξεργασία, η επιλογή κατάλληλου αλγορίθμου αλλά και η επιλογή υπερ-παραμέτρων μπορεί να λύνονται, αλλά αυτό δεν σημαίνει σε καμία περίπτωση ότι όλα θα δουλέψουν με τον βέλτιστο τρόπο ή πλήρως αυτοματοποιημένα ενώ πολλές αποφάσεις μπορούν να οριστούν και σε πιο οργανωμένες μορφές (Randal S. Olson, et al., 2016).

Οι παραπάνω τεχνικές έχουν συγκεκριμένες απαιτήσεις σε συγκεκριμένη δομή προκειμένου να είναι λειτουργικές:

- Το πεδίο μηχανικής μάθησης (classification ή regression)
- Τον αλγόριθμο μηχανικής μάθησης (συνήθως προαιρετικό)
- Τον χώρο υπερ-παραμέτρων του εκάστοτε αλγορίθμου (συνήθως προαιρετικό)
- Την μετρική προς βελτιστοποίηση που πρόκειται να βρεθεί ανάλογα με το πεδίο

Είναι λοιπόν σαφές πως η κάθε λύση έχει συγκεκριμένες διαδικασίες, για παράδειγμα στο στάδιο τις προ-επεξεργασίας (feature selection, feature extraction, normalization) ενώ υπάρχουν αυτοματοποιημένοι μηχανισμοί, οι αποφάσεις τους μπορεί να αλλάξουν με βάση υπερ-παραμέτρους σε πιο υψηλό επίπεδο (στο επίπεδο της λύσης / περιβλήματος). Ακόμη στο στάδιο της μοντελοποίησης, δηλαδή επιλογής αλγορίθμου και υπερ-παραμέτρων, μπορεί να ορισθούν συγκεκριμένες απαιτήσεις, ή περιορισμοί για την διαδικασία αναζήτησης. Αυτό σημαίνει πως συγκεκριμένες τεχνικές μπορούν να ενεργοποιηθούν, ή να αλλάξουν με την κατάλληλη υπερ-παράμετρο.

Ακόμη πολλές αυτοματοποιημένες λύσεις προσφέρουν και είσοδο ροών από στάδια προ-επεξεργασίας και μοντελοποίησης όπως φαίνεται παρακάτω σε ένα pipeline σε γλώσσα python:

```
>>> from sklearn.svm import SVC
>>> from sklearn.preprocessing import StandardScaler
>>> from sklearn.datasets import make_classification
>>> from sklearn.model_selection import train_test_split
>>> from sklearn.pipeline import Pipeline
>>> X, y = make_classification(random_state=0)
>>> X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
>>> pipe = Pipeline([('scaler', StandardScaler()), ('svc', SVC())])
>>> # The pipeline can be used as any other estimator
>>> # and avoids leaking the test set into the train set
>>> pipe.fit(X_train, y_train)
Pipeline(steps=
[
    ('scaler', StandardScaler()),
    ('svc', SVC())
])
>>> pipe.score(X_test, y_test)
```

Πίνακας 4. Παράδειγμα ροής μηχανικής μάθησης

Τέλος, κρίνεται σκόπιμο να γίνει μια επισκόπηση των υπερ-παραμέτρων μερικών από τις δημοφιλέστερες ολοκληρωμένες λύσης μηχανικής μάθησης, οι οποίες δεν φαίνεται να παρουσιάζουν σημαντικές ομοιότητες δεδομένης της διαφορετικής αρχιτεκτονικής τους. Οι υπερ-παράμετροι κάθε λύσης ορίζονται από τον κατασκευαστή τους.

auto-sklearn, είναι μια ολοκληρωμένη λύση η οποία προσφέρει προ-επεξεργασία και μοντελοποίηση με κεντρική ιδέα το Meta Search και το Bayesian Search χρησιμοποιώντας SMAC. Ο όρος Meta προήλθε από τα meta δεδομένα που μπορούν να εξαχθούν από ένα σύνολο δεδομένων, όπως ο αριθμός γραμμών και στηλών, ο αριθμός των τιμών της κλάσης, οι ακραίες τιμές, η συσχέτιση, η λοξότητα, η κύρτωση, η συν-διακύμανση, η βαρύτητα και πολλές ακόμη μετρικές.

Οι διαθέσιμοι αλγόριθμοι για classification είναι οι ακόλουθοι: adaboost, decision_tree, extra_trees, gradient_boosting, k_nearest_neighbors, libsvm_svc, mlp, random_forest, gaussian_nb, multinomial_nb

Οι διαθέσιμοι αλγόριθμοι για regression είναι οι ακόλουθοι: adaboost, ard_regression, decision_tree, extra_trees, gaussian_process, gradient_boosting, k_nearest_neighbors, libsvm_svr, mlp, random_forest

Οι διαθέσιμες τεχνικές προ-επεξεργασίας είναι οι ακόλουθες: missing [mean, median, most_frequent], transformation [one_hot, normalize, standardize, abstract_rescaling, robust scaler, min_max_scaler]

Οι παράμετροι που δέχεται η συγκεκριμένη ολοκληρωμένη λύση είναι οι παρακάτω:

- time_left_for_this_task, ο μέγιστος χρόνος αναζήτησης, κάτι το οποίο θα επιφέρει σημαντικά διαφορετικά αποτελέσματα αφού μπορεί να μην έχει ολοκληρωθεί η αναζήτηση ακόμη και σε έναν ολόκληρο χώρο.
  type = numeric, values = integer, default = 3600
- per_run_time_limit, ο μέγιστος χρόνος αναζήτησης κάθε εκτέλεσης, κάτι το οποίο θα επιφέρει διαφορετικά αποτελέσματα αφού μπορεί να μην έχει ολοκληρωθεί η αναζήτηση σε όλο ον διαθέσιμο χώρο.
  type = numeric, values = integer, default = 1/10 * time_left_for_this_task
- initial_configurations_via_metalearing, η αρχική τοποθέτηση των υπερ-παραμέτρων βάση της ομοιότητας του συνόλου με άλλα προηγούμενα.
  type = numeric, values = integer, default = 25
- ensemble_size, ο αριθμός των μοντέλων που θα προστεθούνε στο σύνολο από την συνολική επιλογή από τις βιβλιοθήκες των μοντέλων.
  type = numeric, values = integer, default = 50
- ensemble_nbest, ο αριθμός των μοντέλων που θα πάρουν συμμετοχή από το ευρύτερο σύνολο που θα χτίσει το ensemble_size.
  type = numeric, values = integer, default = 50
- max_models_on_disc, ο μέγιστος αριθμός των μοντέλων που θα αποθηκευτούν στον δίσκο αλλιώς τα υπόλοιπα θα διαγράφονται.
  type = numeric, values = integer, default = 50
- seed, ο αριθμός τυχαίοτητας για το SMAC, ο οποίος θα καθορίσει και τα παραγόμενα ονόματα των αρχείων.
  type = numeric, values = integer, default = 1
- memory_limit, η μνήμη σε mbs για τον αλγόριθμο μηχανικής μάθησης, αν το όριο αυτό υπερβεί τότε η βιβλιοθήκη θα σταματήσει την εφαρμογή νέων δεδομένων.
  type = numeric, values = integer, default = 3072
- include, το οποίο ορίζει τα επίπεδα του workflow όπως και το ποια components θα πάρουν μέρος στην διαδικασία μεταξύ των ακόλουθων:
  {data_preprocessor, balancing, feature_preprocessor, classifier, regressor}
  type = dictionary, values = dict[str, list[str]], default = None

hp-sklearn, είναι μια ολοκληρωμένη λύση η οποία προσφέρει προ-επεξεργασία και μοντελοποίηση με κεντρική ιδέα το Bayesian Search χρησιμοποιώντας TPE και παρέχοντας την δυνατότητα παράλληλης αναζήτησης μελλοντικά. Ο όρος TPE προήλθε από τα αρχικά Tree Parsen Estimetors, δηλαδή ορίζοντας της υπερ-παραμέτρους σε δενδροειδή μορφή.

Οι διαθέσιμοι αλγόριθμοι για classification είναι οι ακόλουθοι: svc, svc_linear, svc_rbf, svc_poly, svc_sigmoid, liblinear_svc, knn, ada_boost, gradient_boosting, random_forest, extra_trees, decision_tree, sgd, xgboost_classification, multinomial_nb, gaussian_nb, passive_aggressive, linear_discriminant_analysis, quadratic_discriminant_analysis, one_vs_rest, one_vs_one, output_code

Οι διαθέσιμοι αλγόριθμοι για regression είναι οι ακόλουθοι: svr, svr_linear, svr_rbf, svr_poly, svr_sigmoid, knn_regression, ada_boost_regression, gradient_boosting_regression, random_forest_regression, extra_trees_regression, sgd_regression, xgboost_regression

Οι διαθέσιμες τεχνικές προ-επεξεργασίας είναι οι ακόλουθες: dimensionality [pca], transform [one_hot, standard_scaler, min_max_scaler, normalizer, ts_lagselector, tfidf, rbm, colkmeans]

Οι παράμετροι που δέχεται η συγκεκριμένη ολοκληρωμένη λύση είναι οι παρακάτω:

- preprocessing, ο τρόπος με τον οποίο θα γίνει η προ-επεξεργασία των δεδομένων.
  type = object, values = machine learning preprocessors, default = None
- classifier, ο αλγόριθμος μηχανικής μάθησης.
  type = object, values = machine learning classification algorithms, default = None
- regressor, ο αλγόριθμος μηχανικής μάθησης.
  type = object, values = machine learning regression algorithms, default = None
- space, ο χώρος αναζήτησης των υπερ-παραμέτρων ανά αλγόριθμο.
  type = object, values = machine learning regression algorithms, default = None
- algo, ή μέθοδος αναζήτησης μηχανικής μάθησης.
  type = object, values = machine learning algorithms, default = tpe.suggest
- max_evals, ο μέγιστος αριθμός αναζητήσεων, κάτι που θα επιφέρει σημαντικά διαφορετικά αποτελέσματα αφού μπορεί να μην έχει ολοκληρωθεί η αναζήτηση σε όλο τον χώρο.
  type = numeric, values = integer, default = 150
- loss_fn, ή μέθοδος ελαχιστοποίησης κόστους μεταξύ προβλέψεων και πραγματικών τιμών.
  type = function, values = custom_function_float, default = accuracy_score / r2_score
- continues_loss_fn, η loss_fn επιστρέφει το predict_proba ή predict σαν δεύτερο attribute.
  type = boolean, values = {true, false}, default = false
- trial_timeout, ο μέγιστος χρόνος αναζήτησης επανάληψης, όπου θα επιφέρει διαφορετικά αποτελέσματα αφού μπορεί να μην έχει ολοκληρωθεί η αναζήτηση σε όλο τον χώρο.
  type = numeric, values = integer, default = 60
- fit_increment, ο τρόπος ενημέρωσης των επαναλήψεων, ο οποίος είναι σειριακός αλλά ενδέχεται να υποστηριχτεί με την παράλληλη αναζήτηση με spark και mongo trials.
  type = numeric, values = integer, default = 60
- seed, αριθμός τυχαίοτητας για το FMIN, καθορίζει τα παραγόμενα ονόματα αρχείων
- use_partial_fit, αν το μοντέλο υποστηρίζει partial_fit τότε θα χρησιμοποιηθεί για συνεχή μάθηση με το early stopping να σταματά την εκπαίδευση όταν η βελτίωση σταματά.
  type = boolean, values = {true, false}, default = true
- refit, εφαρμογή του επιλεγμένου μοντέλου σε ολόκληρο το σύνολο δεδομένων
  type = boolean, values = {true, false}, default = true
- n_jobs, ο αριθμός των εργασιών προς παράλληλη εκτέλεση
  for = both, type = numeric, values = integer, default = none(1)

auto-weka, είναι μια ολοκληρωμένη λύση η οποία προσφέρει προ-επεξεργασία και μοντελοποίηση με κεντρική ιδέα την επαναληπτική διαδικασία και Bayesian Search και Meta Learning ανάλογα με τον αλγόριθμο επιλογής για την βελτιστοποίηση των υπερ-παραμέτρων.

Οι διαθέσιμοι αλγόριθμοι είναι οι ακόλουθοι: BayesNet, DecisionStump, DecisionTable, GaussianProcesses, IBk, J48, JRip, KStar, LinearRegression, LMT, Logistic, M5P, M5Rules, MultilayerPerceptron, NaiveBayes, NaiveBayesMultinomial, OneR, PART, RandomForest, RandomTree, REPTree, SGD, SimpleLinearRegression, SimpleLogistic, SMO, SMOreg, VotedPerceptron, ZeroR

Οι διαθέσιμοι αλγόριθμοι (Ensemble Methods) είναι οι ακόλουθοι: Stacking, Vote

Οι διαθέσιμοι αλγόριθμοι (Meta-Methods) είναι οι ακόλουθοι: LWL, AdaBoostM1, AdditiveRegression, AttributeSelectedClassifier, Bagging, RandomCommittee, RandomSubSpace

Οι διαθέσιμες τεχνικές επιλογές χαρακτηριστικών αλγόριθμοι είναι οι ακόλουθες: BestFirst 2 GreedyStepwise

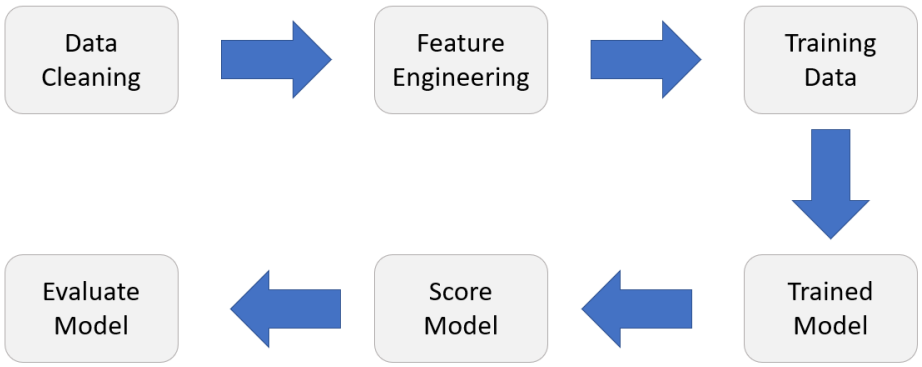
Οι διαθέσιμες τεχνικές προ-επεξεργασίας είναι οι ακόλουθες: outlier [no handling, remove], missing [no handling, replace, customreplace], dimensionality [pca, attribute selection (μια ευρεία γκάμα επιλογών)]

Οι παράμετροι που δέχεται η συγκεκριμένη ολοκληρωμένη λύση είναι οι παρακάτω:

- batchSize, ο αριθμός δειγμάτων ανά batch για την διαδικασία πρόβλεψης
  for = both, type = numeric, values = integer, default = 100
- debug, η μέθοδος εκτέλεσης, σε debug mode θα παραχθούν περισσότερες λεπτομερείς για την λειτουργία του μοντέλου
  for = both, type = boolean, values = {true, false}, default = false
- doNotCheckCapabilities, ο έλεγχος δυνατοτήτων πρώτού χτιστεί το μοντέλο
  for = both, type = boolean, values = {true, false}, default = false
- memLimit, η μέγιστη μνήμη σε megabytes
  for = both, type = numeric, values = integer, default = 1024
- metric, η μετρική αξιολόγησης ανάλογα το πρόβλημα
  for = both, type = nominal, values = {classification / regression metrics}, default = errorRate
  nBestConfigs, ο αριθμός καλύτερων συνδυασμών
  for = both, type = numeric, values = integer, default = 1
- numDecimalPlaces, ο αριθμός δεκαδικών ψηφίων προς προβολή
  for = both, type = numeric, values = integer, default = 2
- parallelRuns, οι παράλληλες εργασίες (σε πειραματικό στάδιο)
  for = both, type = numeric, values = integer, default = 1
- seed, αριθμός τυχαίοτητας
  for = both, type = numeric, values = integer, default = 123
- timeLimit, το χρονικό περιθώριο σε λεπτά
  for = both, type = numeric, values = integer, default = 15

5. Περιγραφή προβλήματος

Όπως είναι ήδη εμφανές, δύο είναι τα κύρια συστατικά ενός μοντέλου πρόβλεψης, το στάδιο της προ-επεξεργασίας και αυτό της μοντελοποίησης. Κάθε ένα από αυτά δρα σημαντικά στην απόδοση του μοντέλου πρόβλεψης καθώς αυτές οι διαδικασίες είναι τεχνικές ανάλυσης επιστημόνων στο χώρο που χρήζουν εντατικότερης ανάλυσης. Γίνεται όμως αυτές οι διαδικασίες να αυτοματοποιηθούν? (Marion Neumann, 2019).



Εικόνα 16. Βασικές διαδικασίες προβλήματος αξιολόγησης

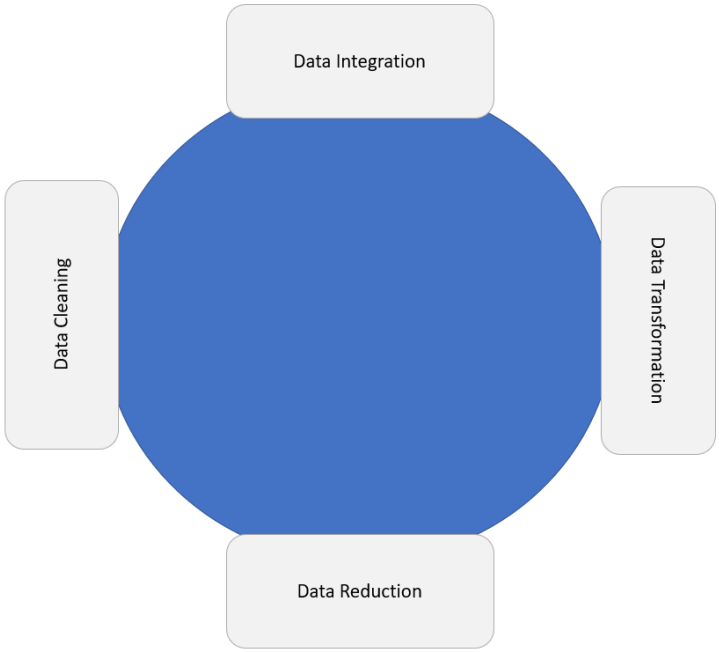
Τα δεδομένα του πραγματικού κόσμου περιέχουν γενικά θορύβους, ελλιπής τιμές και ίσως σε μη χρησιμοποιήσιμη μορφή που δεν μπορεί να χρησιμοποιηθεί απευθείας για μοντέλα μηχανικής εκμάθησης. Η προ-επεξεργασία δεδομένων είναι απαραίτητη εργασία για τον καθαρισμό των δεδομένων και την καθιέρωσή τους κατάλληλα για ένα μοντέλο μηχανικής εκμάθησης, το οποίο αυξάνει επίσης την ακρίβεια και την αποτελεσματικότητα του μοντέλου μηχανικής μάθησης.

Επιπρόσθετα, για οποιοδήποτε δεδομένο πρόβλημα μηχανικής μάθησης, μπορούν να εφαρμοστούν πολυάριθμοι αλγόριθμοι και μπορούν να δημιουργηθούν πολλαπλά μοντέλα. Ένα πρόβλημα ταξινόμησης ανίχνευσης ανεπιθύμητων μηνυμάτων, για παράδειγμα, μπορεί να επιλυθεί χρησιμοποιώντας μια ποικιλία μοντέλων, συμπεριλαμβανομένων των απλών τεχνικών, της λογιστικής παλινδρόμησης και τεχνικών βαθιάς εκμάθησης. Το να υπάρχουν πολλές επιλογές είναι καλό, αλλά η απόφαση για το ποιο μοντέλο θα εφαρμοστεί στην παραγωγή είναι μια κρίσιμη επιλογή. Αν και υπάρχει μια σειρά από μετρήσεις απόδοσης για την αξιολόγηση ενός μοντέλου, δεν είναι συνετό να εφαρμόζεται ο κάθε αλγόριθμος για κάθε πρόβλημα. Αυτό απαιτεί πολύ χρόνο και πολλή δουλειά.

Επομένως, η εργασία αναζήτησης του σωστού αλγορίθμου είναι σημαντική και για αυτό τον λόγο θα γίνει εμβάθυνση στα προβλήματα της προ-επεξεργασίας και της μοντελοποίησης.

5.1 Προεπεξεργασία

Η προ-επεξεργασία δεδομένων είναι ένα αναπόσπαστο βήμα στη Μηχανική Μάθηση, καθώς η ποιότητα των δεδομένων και οι χρήσιμες πληροφορίες που μπορούν να προκύψουν από αυτήν επηρεάζουν άμεσα την ικανότητα του μοντέλου να μαθαίνει. Επομένως, είναι εξαιρετικά σημαντικό γίνεται προ-επεξεργασία των δεδομένων πριν τροφοδοτηθούν στο μοντέλο (S. Banumathi, et al., 2016).



Εικόνα 17. Βασικά στάδια προ-επεξεργασίας προβλήματος αξιολόγησης

Το πρώτο στάδιο σίγουρα είναι η επιλογή των κατάλληλων γραμμών και στηλών, για παράδειγμα η αποφυγή γραμμών (δειγμάτων που εμφανίζουν ακραίες τιμές μπορούν είτε να απαλειφθούν είτε να κανονικοποιηθούν), ή, η αποφυγή στηλών (χαρακτηριστικά τα οποία προκαλούν θόρυβο ή επαναλαμβάνουν πληροφορία).

Κάποιες φορές η προ-επεξεργασία των δεδομένων δεν είναι απλά σημαντική αλλά απαραίτητη για την λειτουργία όπως στα ακόλουθα στάδια τα οποία είναι απαραίτητα σε συγκεκριμένα σύνολα δεδομένων που εμφανίζουν ελλιπείς ή αλφαριθμητικές τιμές καθώς οι περισσότεροι αλγόριθμοι δεν μπορούν να διαχειριστούν ελλιπείς ή αλφαριθμητικές τιμές.

- Μια ακόμη δυσκολία στην λειτουργία των μοντέλων είναι οι κενές τιμές, οι περισσότεροι αλγόριθμοι Μηχανικής Μάθησης δεν δέχονται κενές τιμές. Έτσι με κατάλληλες τεχνικές όπως αντικατάσταση με την συχνότερη τιμή για τις κατηγορικές τιμές, ή αντικατάσταση με την μέση τιμή για τις αριθμητικές τιμές μπορεί να γίνει απαλοιφή των κενών τιμών.
- Μια ακόμη δυσκολία στην λειτουργία των μοντέλων είναι οι αλφαριθμητικές τιμές, οι περισσότεροι αλγόριθμοι Μηχανικής Μάθησης δεν δέχονται αλφαριθμητικές τιμές για τα χαρακτηριστικά. Έτσι με κατάλληλες τεχνικές όπως ο μετασχηματισμός από αλφαριθμητικές σε πολλά μικρότερα αριθμητικά διανύσματα ή αριθμούς μπορεί να επιλύσει το πρόβλημα.

5.2 Μοντελοποίηση

Υπάρχουν αρκετοί αλγόριθμοι μηχανικής μάθησης, πολλές υπερ-παράμετροι αλλά και πολλές διαθέσιμες τιμές υπερ-παραμέτρων που μπορούν να εφαρμοστούν (Quanming Yao, et al., 2018). Κάποιες υπερ-παράμετροι μπορούν να πάρουν διακριτές τιμές και άλλες υπερ-παράμετροι πραγματικές τιμές όπου το πλήθος μπορεί να είναι ένα πρακτικά ένα άπειρο σύνολο. Αν σε αυτό το σημείο αναλογιστεί κανείς τους πιθανούς συνδυασμούς, όπως για παράδειγμα ένα σύνολο διαθέσιμων αλγορίθμων μηχανικής μάθησης με ένα σύνολο υπερ-παραμέτρων και το εύρος τιμών του διαθέσιμου χώρου εφαρμογής, τότε πραγματικά οι συνδυασμοί αυξάνονται με εκθετικό ρυθμό.

Εικόνα 18. Βασικά στάδια μοντελοποίησης προβλήματος αξιολόγησης

Ειδικότερα για το πρόβλημα επιλογής αλγορίθμου η μέθοδος meta-search έρχεται και δίνει λύση εύκολα και γρήγορα μειώνοντας έτσι τους συνδυασμούς των υπόλοιπων πιθανών αλγορίθμων. Ακόμη το πρόβλημα της βελτιστοποίησης των υπερ-παραμέτρων φαίνεται να λύνεται γρήγορα με την μέθοδο Bayesian Optimization αφού και εδώ η λύση δίνεται με την μείωση των δοκιμών βάση της πιθανότητας βελτίωσης. Παρόλα αυτά καθώς οι αλγόριθμοι και οι υπερ-παράμετροι πληθαίνουν και παράλληλα τα σύνολα δεδομένων γίνονται ολοένα και μεγαλύτερα το πρόβλημα της υπολογιστικής ισχύς και του χρόνου παραμένει στο προσκήνιο.

Όπως έχει προαναφερθεί η επιλογή των υπερ-παραμέτρων και του εύρους αυτών είναι θέμα όχι μόνο αποτελέσματος ακρίβειας αλλά και απόδοσης σε χρόνο. Αν για παράδειγμα υπολογιστούν οι συνδυασμοί ενός μόνο αλγορίθμου (Random Forest) ο οποίος χρησιμοποιείται σε διάφορα πεδία μηχανικής μάθησης (classification αλλά και regression) με δέκα υπερ-παραμέτρους οι οποίες έχουν νόημα στην βελτιστοποίηση του μοντέλου θα δούμε πως οι τέσσερις είναι διακριτές με κατά μέσο όρο δύο τιμές αλλά και οι άλλες έξι είναι πραγματικές. Από τις έξι αυτές πραγματικές υπερ-παραμέτρους οι τέσσερις παίρνουν ακέραιες τιμές ενώ οι υπόλοιπες δύο παίρνουν δεκαδικές τιμές. Μια ακόμη λύση σε αυτό το πρόβλημα είναι να χρησιμοποιηθούν μόνο υπερ-παράμετροι που πραγματικά έχουν νόημα για την καλύτερη λειτουργεία του μοντέλου αλλά και οι τιμές εφαρμογής που έχουν νόημα στην λειτουργία του μοντέλου.

Επιπλέον η αναζήτηση των υπερ-παραμέτρων δεν θα πρέπει να χειρίζεται σαν ένα «blackbox» αναζητώντας σε όλο το διαθέσιμο εύρος καθώς ανάλογα με το σύνολο δεδομένων αυτό το εύρος μπορεί να περιοριστεί. Ένα κλασικό παράδειγμα είναι αυτό το KNeighbords ο οποίος αν βρεθεί με ένα σύνολο δεδομένων με λίγα δείγματα και σε cross-validation διαδικασίες τότε τα δείγματα θα γίνουν ακόμη λιγότερα με αποτέλεσμα να μην υπάρχουν αρκετοί γείτονες για τον υπολογισμό. Κάτι τέτοιο μπορεί να διαχειριστεί και να δίνονται τιμές με τους μέγιστους γείτονες.

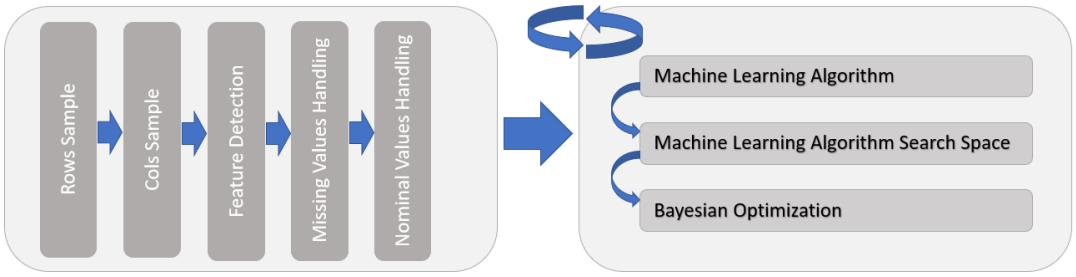
Τέλος, ακόμη και αν έχει γίνει η απαραίτητη πρόβλεψη για όλα τα παραπάνω, σε μεγάλα σύνολα δεδομένων θα υπάρξει και πάλι το πρόβλημα της απαίτησης για υπολογιστική ισχύ αλλά και η απαίτηση σε χρόνο επεξεργασίας.

6. Περιγραφή πρότασης

Σε άλλες έρευνες έχουν προταθεί διάφορες τεχνικές στο στάδιο της προ-επεξεργασίας αλλά και της μοντελοποίησης, όπως έπειτα από την προ-επεξεργασία γίνεται πρώτα η επιλογή του αλγορίθμου και ύστερα βελτιστοποίηση των υπερ-παραμέτρων (Włodzisław Duch, et al., 2018) (βλ. auto-sklearn), ή, η τυχαία επιλογή αλγορίθμου και έπειτα η βελτιστοποίηση υπερ-παραμέτρων (Jasper Snoek, et al., 2012) (βλ. hp-sklearn).

Εικόνα 19. Preprocessing with Meta Search and Bayesian Search

Αντίθετα, σε αυτήν την έρευνα θα γίνει μία προσπάθεια αξιολόγησης μια τεχνικής προ-επεξεργασίας αλλά και μοντελοποίησης όπου ο αλγόριθμος μηχανικής μάθησης και οι υπερ-παράμετροι επιλέγονται μαζί (πρόβλημα CASH) ώστε να εξαλειφθεί η περίπτωση ενός συνδυασμού αλγορίθμου / υπερ-παραμέτρων που μπορεί να επιφέρει καλύτερο αποτέλεσμα από μία επιλογή υπερ-παραμέτρων δεδομένου ενός μόνο αλγορίθμου η οποία έχει επέλθει από άλλη διαδικασία (βλ. meta-search).



Εικόνα 20. Preprocessing with Optimized Brute Bayesian Search

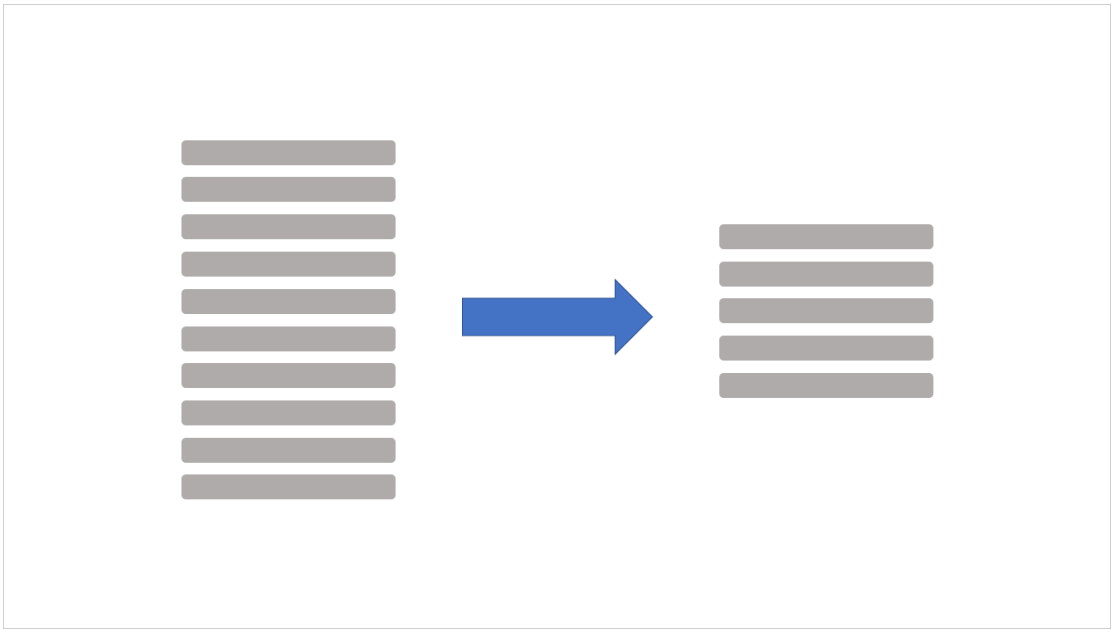
Προφανώς αυτή η τεχνική μοντελοποίησης θα ερευνήσει πλήθος αλγορίθμων και υπερ-παραμέτρων και το πρόβλημα επεξεργαστικής ισχύς και του χρόνου εντείνεται. Με αυτό το τρόπο θα δούμε ότι η συγκεκριμένη έρευνα προτείνει την αντιμετώπιση του προβλήματος με την τεχνική της δειγματοληψίας γραμμών και στηλών με συγκεκριμένη μεθοδολογία, παράλληλα με την χρήση μεθόδων state-of-the-art (βλ. Bayesian Optimization). Δηλαδή η τεχνική αυτή χρησιμοποιεί ένα υπο-σύνολο δεδομένων προκειμένου να «ζυγίσει» το κόστος δοκιμών όλων των αλγορίθμων με την εφαρμογή των μοντέλων σε δείγμα του αρχικού συνόλου. Για παράδειγμα μπορεί να επιλεχθεί ένα μικρότερο σύνολο δεδομένων (γραμμές & στήλες) με τυχαίο ή έξυπνο τρόπο ώστε να γίνει η αναζήτηση για τον καλύτερο αλγόριθμο αλλά και τις υπερ-παραμέτρους με την «υπόθεση» ότι οι ίδιοι συνδυασμοί θα δουλέψουν το ίδιο σε όλο το σύνολο δεδομένων. Στην περίπτωση ισοπαλίας μπορούν να ερευνηθούν διάφοροι τρόποι διαχείρισης.

Επομένως, θα πρέπει να ξεκινήσει μια πειραματική διαδικασία όπου θα αποδεικνύει αυτή την υπόθεση. Στα αρχικά στάδια αυτής της πειραματικής διαδικασίας, θα πρέπει να σχεδιαστεί μια λογική διαδικασία (workflow) η οποία αφενός είναι απαραίτητη για την εκτέλεση των αλγορίθμων μηχανικής μάθησης αφού προαπαιτούμενο είναι το στάδιο της προ-επεξεργασίας και αφετέρου η οργάνωση και η πραγματοποίηση των αποφάσεων με τον ίδιο τρόπο κάθε φορά αλλά και μεταξύ υποσυνόλου (sample) και ολόκληρου του συνόλου (complete).

6.1 Δειγματοληψία Γραμμών

Η δειγματοληψία δεδομένων και ειδικότερα γραμμών αναφέρεται σε στατιστικές ή μη στατιστικές μεθόδους για την επιλογή παρατηρήσεων σε ένα σύνολο με στόχο την εκτίμηση μιας παραμέτρου πληθυσμού. Τόσο η δειγματοληψία δεδομένων όσο και η επανα-δειγματοληψία δεδομένων είναι μέθοδοι που απαιτούνται σε ένα πρόβλημα προγνωστικής μοντελοποίησης. Φυσικά η δειγματοληψία θα πρέπει να γίνει μετά την αφαίρεση δειγμάτων που μπορεί να είναι λανθασμένα ή εμφανίζουν ακραίες ή πολλές κενές τιμές. Με αυτόν τον τρόπο μπορεί ένα μοντέλο μηχανικής μάθησης να τροφοδοτηθεί με λιγότερο αριθμό γραμμών (δειγμάτων) προκειμένου να παρθούν κάποιες αποφάσεις οι οποίες θα αντιπροσωπεύουν όλο το δείγμα σε λιγότερο χρόνο.

Συγκεκριμένα, η αντιμετώπιση μεγάλου όγκου δεδομένων (γραμμές) γίνεται με δειγματοληψία, δηλαδή αν ο όγκος δεδομένων είναι μεγαλύτερος από ένα threshold (α) θα γίνεται δειγματοληψία με ένα threshold (β) φυσικά με την κατάλληλη μεθοδολογία όπου θα υπάρχει ένα ελάχιστο όριο δειγμάτων προκειμένου να αποφευχθεί το underfitting αλλά και ένα μέγιστο όριο δειγμάτων προκειμένου να αποφευχθεί το overfitting. Σε αυτό το σημείο φυσικά πάλι θα μπορούσε να αναλογιστεί κανείς πως η αντιμετώπιση με δειγματοληψία χρησιμοποιώντας ένα κατώτατο και ένα ανώτατο όριο καθιστά μια διαφορετική έρευνα. Για αυτό τον λόγο, ο παράγοντας αυτός θα μείνει σταθερός, το threshold θα παραμείνει σταθερό χωρίς ελάχιστο και μέγιστο αριθμό δειγμάτων, ενώ το ποσοστό δειγματοληψίας θα πραγματοποιηθεί με 10% στα σύνολα δεδομένων. Η επιλογή των γραμμών μπορεί να γίνει με ποικίλους τρόπους όπως είτε με random, είτε με stratified τρόπο. Η συγκεκριμένη έρευνα προτείνει τον random τρόπο ο οποίος ακολουθεί μια κανονική κατανομή. Ιδιαίτερη προσοχή πρέπει να δοθεί κατά την διαδικασία Gaussian Sampling καθώς αν το ζητούμενο υπο-σύνολο δεν καλύπτει αρκετά δείγματα τότε θα πρέπει να ληφθεί μεγαλύτερο σύνολο δεδομένων.

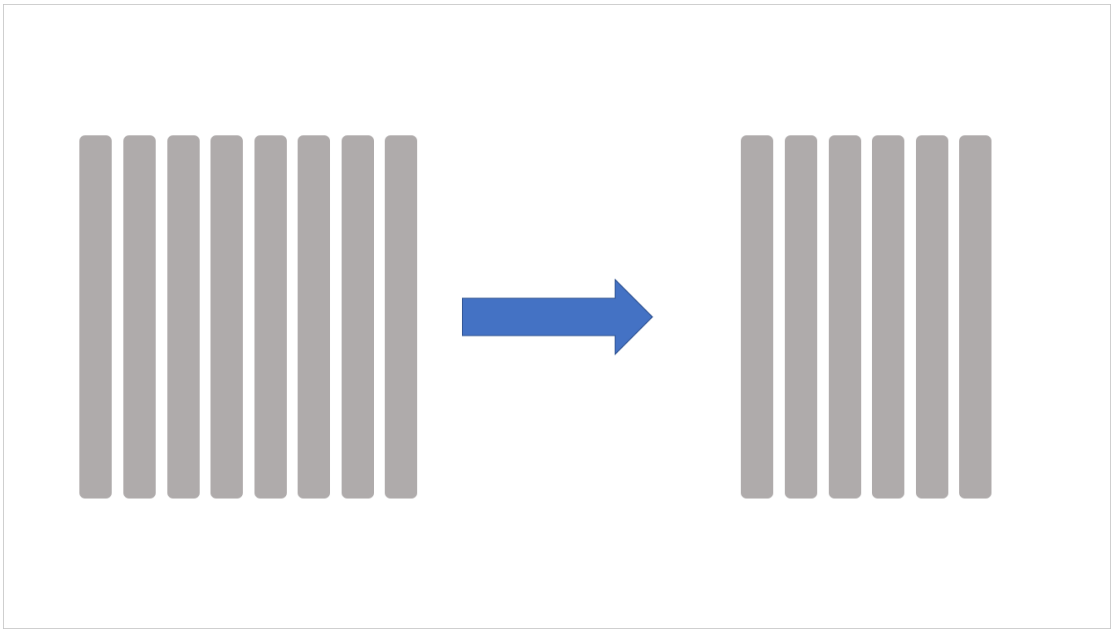


Εικόνα 21. Δειγματοληψία γραμμών

6.2 Δειγματοληψία Στηλών

Η δειγματοληψία δεδομένων και ειδικότερα στηλών είναι μια δύσκολη διαδικασία αφού θα πρέπει να επιλεχθούν στήλες (χαρακτηριστικά) που να είναι σημαντικά για το εκάστοτε πρόβλημα αλλά και τον στόχο (κλάση). Και εδώ είναι επιθυμητό να μειωθεί ο αριθμός των μεταβλητών εισόδου τόσο για τη μείωση του υπολογιστικού κόστους της μοντελοποίησης όσο και, σε ορισμένες περιπτώσεις, για τη βελτίωση της απόδοσης του μοντέλου αφού πολλά χαρακτηριστικά μπορεί να παράγουν «θόρυβο» στην λειτουργία του μοντέλου. Οι μέθοδοι επιλογής χαρακτηριστικών που βασίζονται σε στατιστικά, περιλαμβάνουν την αξιολόγηση της σχέσης μεταξύ κάθε μεταβλητής εισόδου και της μεταβλητής στόχου χρησιμοποιώντας στατιστικά και την επιλογή εκείνων των μεταβλητών εισόδου που έχουν την ισχυρότερη σχέση με τη μεταβλητή στόχο (κλάση). Αυτές οι μέθοδοι μπορεί να είναι γρήγορες και αποτελεσματικές, αν και η επιλογή των στατιστικών μέτρων εξαρτάται από τον τύπο δεδομένων τόσο των μεταβλητών εισόδου όσο και των μεταβλητών εξόδου.

Συγκεκριμένα, η αντιμετώπιση μεγάλου όγκου δεδομένων (στήλες) γίνεται με δειγματοληψία, αν ο αριθμός των χαρακτηριστικών των δεδομένων είναι μεγαλύτερος από ένα threshold (α) τότε θα εξετάζονται στήλες ανά δύο οι οποίες είναι correlated πάνω από ένα threshold (β) και θα διαγράφονται αυτές οι οποίες παρουσιάζουν το μικρότερο correlation με την κλάση. Σε αυτό το σημείο φυσικά θα μπορούσε να αναλογιστεί κανείς πως η αντιμετώπιση με δειγματοληψία στις στήλες χρησιμοποιώντας ένα κατώτατο και ένα ανώτατο όριο καθιστά μια διαφορετική έρευνα. Για αυτό τον λόγο ο παράγοντας αυτός θα μείνει σταθερός, το threshold θα παραμείνει σταθερό χωρίς και μέγιστο αριθμό στηλών, ενώ το ποσοστό συσχέτισης θα πραγματοποιηθεί με 80% στα σύνολα δεδομένων. Η συγκεκριμένη έρευνα προτείνει την επιλογή των στηλών με την συσχέτιση Pearson. Ιδιαίτερη προσοχή πρέπει να δοθεί κατά την διαδικασία Pearson Correlation καθώς όλες οι αλφαριθμητικές τιμές θα πρέπει να μετατραπούν σε αριθμητικές όπως θα γίνει εμφανές παρακάτω.

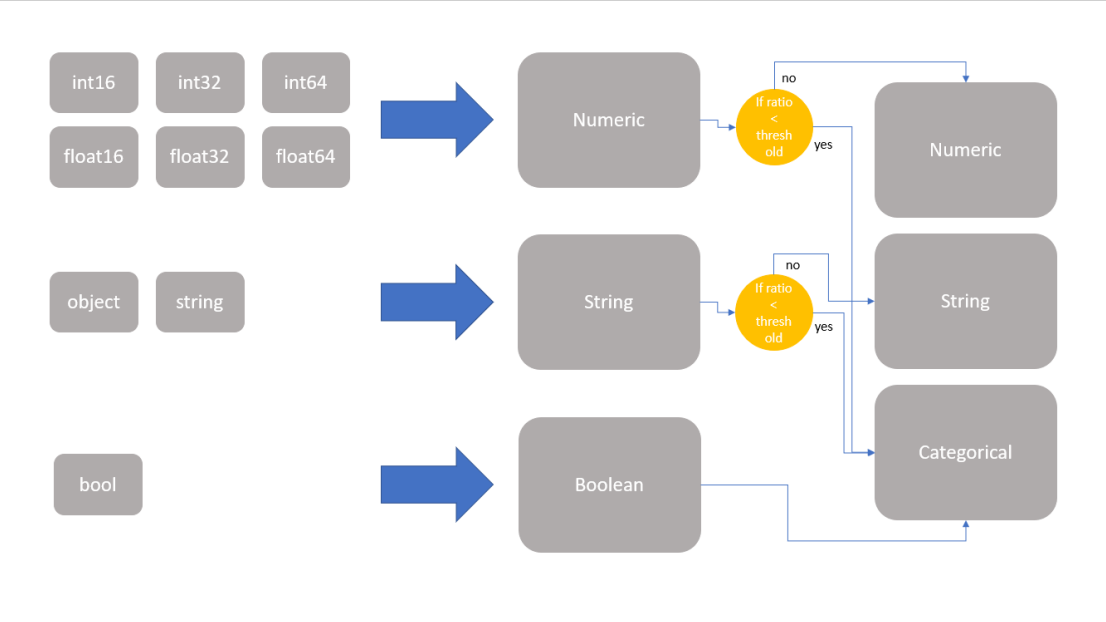


Εικόνα 22. Δειγματοληψία στηλών

6.3 Διαχείριση Χαρακτηριστικών

Ένα σύνολο δεδομένων έχει διάφορα χαρακτηριστικά τα οποία φέρουν διαφορετικούς τύπους, συγκεκριμένα στην python οι τύποι αυτοί μπορεί να είναι: object, string, int16, int32, int64, float16, float32, float64, bool και άλλα. Ο καθένας από αυτούς τους τύπους μπορεί να μετατραπεί αντίστοιχα σε έναν γενικότερο τύπο όπως string ή numeric. Επιπρόσθετα, αν ο λόγος του συνόλου των διαφορετικών τιμών έναντι όλων των τιμών είναι μικρός μπορεί να χαρακτηριστεί ακόμη πιο γενικά και ως categorical έναντι του multivariant σε περίπτωση που ο λόγος είναι μεγάλος. Στην περίπτωση που ένα string ή numeric χαρακτηριστεί categorical τότε εμπίπτει σε αυτήν την κατηγορία αλλιώς παραμένει στην ίδια. Η απόφαση αυτή θα βοηθήσει μετέπειτα βήματα για την διαχείριση των αλφαριθμητικών ή αλλιώς των string τιμών.

Συγκεκριμένα, η παραπάνω διαδικασία θα κάνει ξεκάθαρη την λήψη αποφάσεων στην διαχείριση κενών τιμών, αφού πλέον θα είναι γνωστός ο τύπος της πρόβλεψης, αλλά και στην αναγνώριση του προβλήματος μηχανικής μάθησης. Για παράδειγμα, αν το τύπος της κολώνας πρόβλεψης είναι categorical τότε το πρόβλημα είναι classification, από την άλλη μεριά αν είναι numeric τότε το πρόβλημα είναι regression, ενώ σε περίπτωση που είναι string θα είναι και πάλι classification.

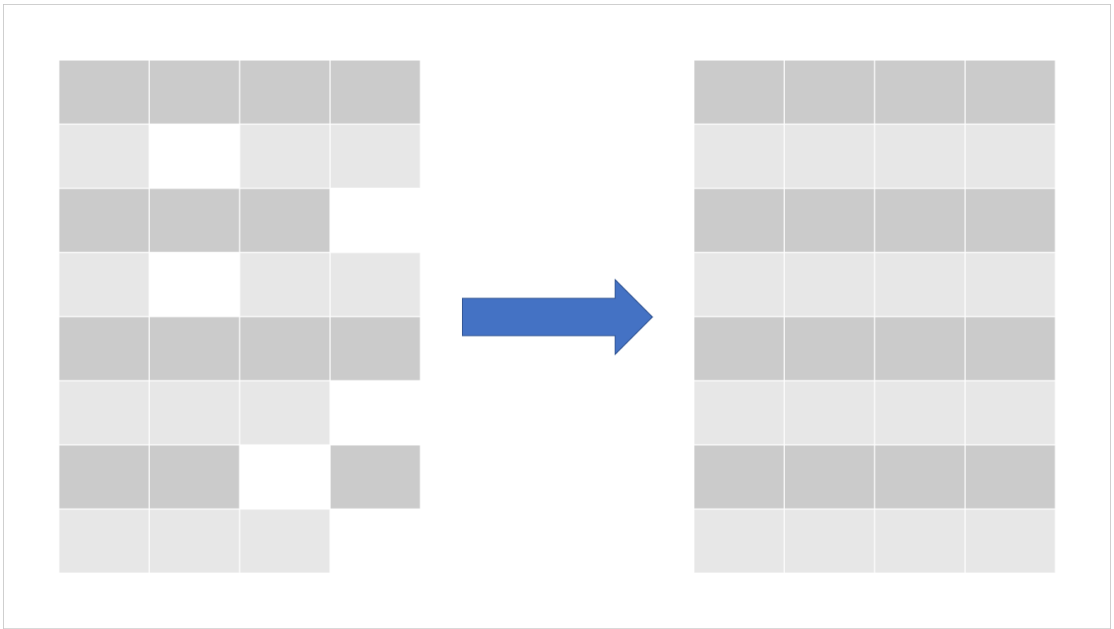


Εικόνα 23. Διαχείριση χαρακτηριστικών

6.4 Αντιμετώπιση Κενών Τιμών

Τα περισσότερα δεδομένα αποτελούνται και από κενές τιμές δηλαδή τιμές που λείπουν και η πιθανότητα να λείπουν τιμές αυξάνεται όσο αυξάνεται και το μέγεθος του συνόλου δεδομένων. Η έλλειψη τιμών σε ένα σύνολο δεδομένων μπορεί να προκαλέσει σφάλματα σε ορισμένους αλγόριθμους μηχανικής εκμάθησης. Οι ελλείπεις τιμές είναι συχνό φαινόμενο στα σύνολα δεδομένων. Δυστυχώς, οι περισσότερες τεχνικές πρόβλεψης δεν μπορούν να χειριστούν καμία τιμή που λείπει. Επομένως, αυτό το πρόβλημα πρέπει να αντιμετωπιστεί πριν από τη μοντελοποίηση. Υπάρχουν διάφορες τεχνικές για την αντιμετώπιση των κενών τιμών όπως η απαλοιφή τους, η αντικατάσταση τους με συγκεκριμένες τιμές ή και πιο προχωρημένες λύσεις όπως οι πρόβλεψη τους με βάση τις υπόλοιπες τιμές.

Συγκεκριμένα, η αντιμετώπιση κενών τιμών σε γραμμή και στήλη, αν οι κενές τιμές είναι περισσότερες από ένα συγκεκριμένο threshold θα διαγράφονται γραμμές και στήλες, ενώ αν είναι μικρότερο θα γίνονται impute με την χρήση μηχανικής μάθησης. Σε αυτό το σημείο φυσικά θα μπορούσε να αναλογιστεί κανείς πως η αντιμετώπιση με impute είναι μια διαφορετική μεμονωμένη έρευνα αφού θα τροποποιήσει το σύνολο δεδομένων και ενδεχομένως να επηρεάσει τα αποτελέσματα. Για αυτόν τον λόγο ο παράγοντας αυτός θα μείνει σταθερός, το threshold θα παραμείνει σταθερό στο 20% δηλαδή γραμμές και στήλες που εμφανίζουν ελλιπείς τιμές πάνω από το threshold θα διαγράφονται ενώ οι υπόλοιπες θα γίνονται impute. Συγκεκριμένα οι αλφαβητικές τιμές θα γίνονται impute με Logistic Regression ενώ οι αριθμητικές με Linear Regression, δηλαδή με γραμμικά μοντέλα.

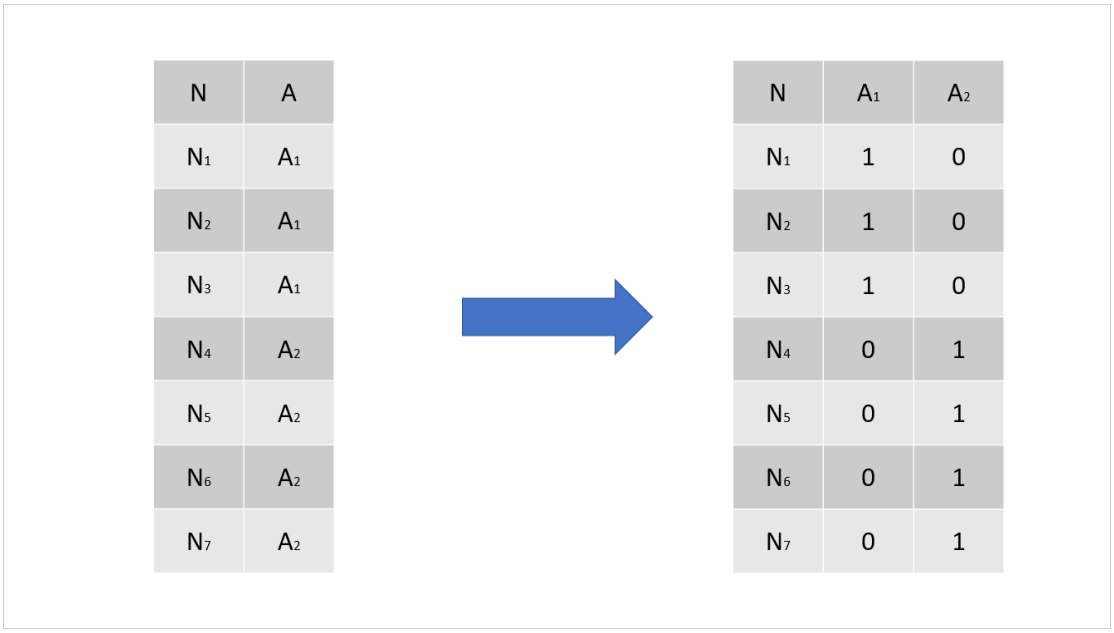


Εικόνα 24. Αντιμετώπιση κενών τιμών

6.5 Διαχείριση Αλφαριθμητικών Τιμών

Ένα σύνολο δεδομένων έχει διάφορες τιμές με διαφορετικό τύπο όπως string, και numeric. Δυστυχώς, οι περισσότεροι αλγόριθμοι μηχανικής μάθησης δεν μπορούν να δουλέψουν με string τιμές ως χαρακτηριστικά. Για αυτό τον λόγο πρέπει αυτές οι string τιμές να μετατραπούν σε numeric κάτι φυσικά που πρέπει να γίνει με συγκεκριμένο τρόπο ο οποίος δεν θα δημιουργεί συσχετίσεις βαθμού μεταξύ των αριθμών όταν δεν υπάρχει συσχέτιση. Ο μόνος τρόπος για να γίνει αυτό, είναι οι αλφαριθμητικές τιμές να μετατραπούν σε διανύσματα από μηδέν και ένα δίνοντας 50% βαρύτητα.

Συγκεκριμένα, η αντιμετώπιση αλφαριθμητικών τιμών γίνεται χρησιμοποιώντας τον MultiLabelBinirizer, ο οποίος δέχεται σαν είσοδο όλες τις αλφαριθμητικές τιμές και εξάγει διανύσματα από μηδέν και ένα. Ένα χαρακτηριστικό που θα πρέπει να δοθεί ιδιαίτερη προσοχή είναι πως αφού χρησιμοποιηθεί ο MultiLabelBinirizer για το training set θα πρέπει να χρησιμοποιηθεί και για το testing αφού και τα δύο θα πρέπει να έχουν την ίδια μορφή. Εφόσον στο testing set μπορεί να υπάρχει τιμή η οποία δεν έχει κωδικοποιηθεί στο training set και προφανώς δεν υπάρχει ούτε στο λεξικό τού MultiLabelBinirizer, δημιουργείται εύλογα ένα νέο πρόβλημα. Ευτυχώς, οι τελευταίες εκδόσεις του MultiLabelBinirizer (sklearn>=0.20.2) κωδικοποιούν τις unseen τιμές στο μηδενικό διάνυσμα.

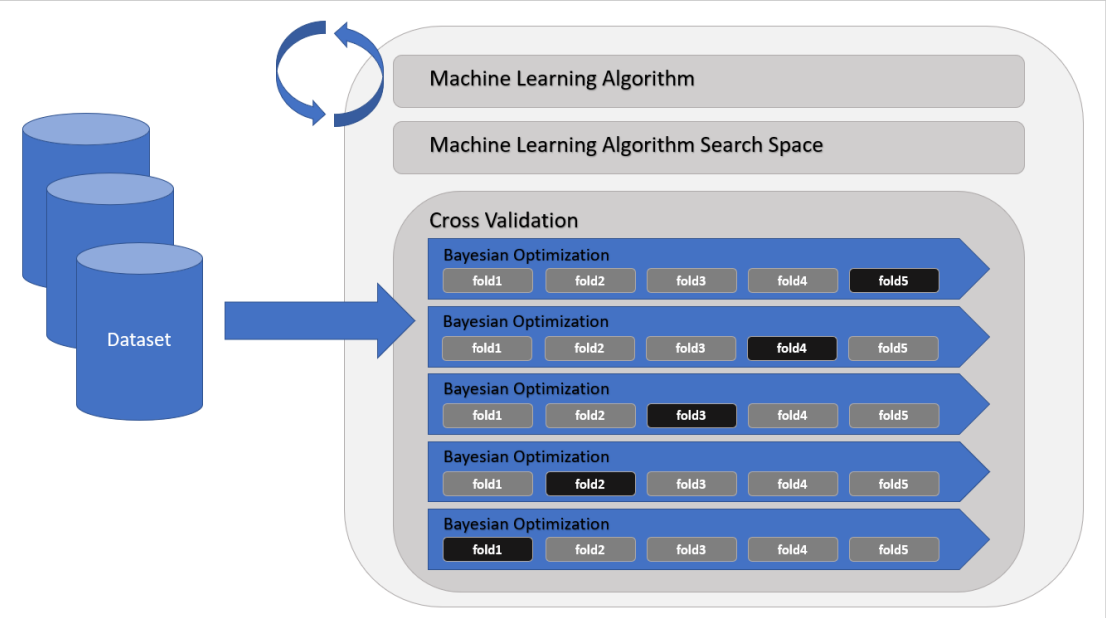


Εικόνα 25. Διαχείριση αλφαριθμητικών τιμών

6.6 Επιλογή Αλγορίθμων και Υπερπαραμέτρων

Η επιλογή υπερ-παραμέτρων λειτουργεί εκτελώντας πολλαπλές δοκιμές σε μία εργασία εκπαίδευσης. Κάθε δοκιμή είναι μια πλήρης εκτέλεση της εκπαιδευτικής εφαρμογής με τιμές για τις υπερ-παραμέτρους που έχουν επιλεχθεί εντός των ορισμένων ορίων. Η αναζήτηση υπερ-παραμέτρων βελτιστοποιεί μια μεμονωμένη μεταβλητή στόχο, που ονομάζεται επίσης μέτρηση υπερ-παραμέτρων, που έχει οριστεί. Η ακρίβεια του μοντέλου, όπως υπολογίζεται από ένα πάσο αξιολόγησης, είναι μια κοινή μέτρηση.

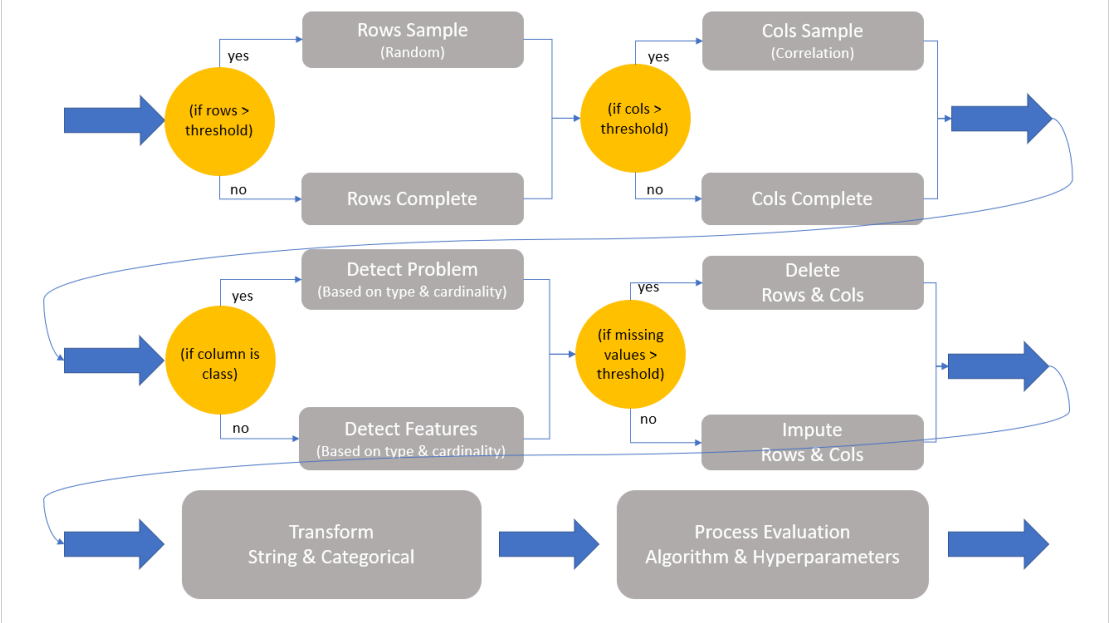
Η ταυτόχρονη επιλογή αλγορίθμων και υπερ-παραμέτρων γίνεται επαναληπτικά με την μέθοδο Bayesian Optimization Search στο υποσύνολο, η οποία υπολογίζει τις παραμέτρους της επόμενης επανάληψης με βάση την πιθανότητα βελτίωσης του αποτελέσματος σε κάθε επανάληψη. Ονομάζουμε την νέα τεχνική CASH-SBE (Combined Algorithm Selection and Hyperparameter Optimization with Sampled Bayesian Estimation). Επιπρόσθετα υπάρχει ένα βήμα διαχείρισης των τιμών των υπερ-παραμέτρων ανάλογα το σύνολο δεδομένων. Παράλληλα, η αξιολόγηση γίνεται με cross-validation και συγκεκριμένα 5 folds. Αυτό σημαίνει πως επαναληπτικά για εννιά αλγορίθμους (classification / regression) με όλες τις υπερ-παραμέτρους τους θα πραγματοποιηθεί cross-validation με 5 folds. Το cross-validation είναι μια διαδικασία όπου θα χωρίσει το σύνολο δεδομένων σε 5 κομμάτια από τα οποία θα χρησιμοποιήσει τα 4 για training και το 1 για validation. Αυτή η διαδικασία θα γίνει 5 φορές με διαφορετικά κάθε φορά τα 4 folds προς εκπαίδευση και το 1 fold προς validation. Στο τέλος, θα υπολογιστεί ο μέσος όρος της εκπαίδευσης και της επαλήθευσης από κάθε επανάληψη ώστε αυτές οι μετρικές να αντιπροσωπεύουν όλο το σύνολο δεδομένων. Η προ-επιλεγμένη μετρική για το classification είναι το accuracy και τη μετρική για το regression είναι το r2 score. Στις περιπτώσεις ισοπαλίας χρησιμοποιήθηκε ο αλγόριθμος με την μικρότερη διάρκεια εφαρμογής των δεδομένων. Τέλος, μέσο μιας πειραματικής διαδικασίας που πραγματοποιήθηκε κρίθηκε σκόπιμο πως από το μοντέλο που επιλέγεται με βάση το sample (αλγόριθμος και υπερ-παράμετροι) να χρησιμοποιηθεί μόνο ο επιλεγμένος αλγόριθμος, αφού οι υπερ-παράμετροι έχουν υψηλή σχέση με το σύνολο δεδομένων και αυτές αναζητούνται εκ-νέου στο complete σύνολο δεδομένων.



Εικόνα 26. Επιλογή Αλγορίθμων και Υπερ-παραμέτρων

7. Αξιολόγηση διαδικασίας

Χρησιμοποιώντας το workflow όπως αυτό περιγράφηκε αναλυτικότερα βήμα προς βήμα, παρακάτω παρουσιάζεται η γενικότερη μορφή της διαδικασίας σε υψηλό επίπεδο καθώς θα είναι αυτή που θα εκτελεστεί κατά την πειραματική διαδικασία.



Εικόνα 27. Ολοκληρωμένη διαδικασία μηχανικής μάθησης

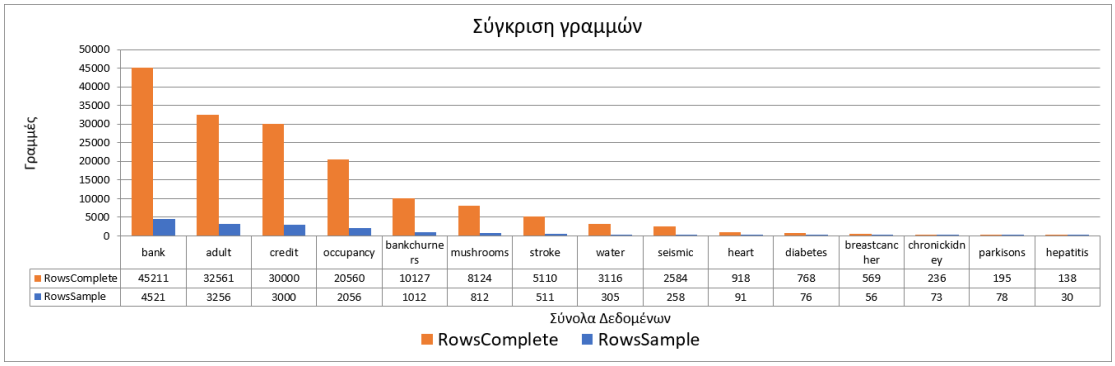
Συγκεκριμένα, πειραματική διαδικασία πραγματοποιήθηκε με την χρήση 35 συνόλων δεδομένων. Τα 30 από αυτά είναι σύνολα δεδομένων για classification ενώ τα υπόλοιπα 5 για regression. Από τα 30 σύνολα δεδομένων για classification τα μισά χρησιμοποιούνται για binary classification, ενώ τα υπόλοιπα χρησιμοποιούνται για multiclass classification.

Κατά την διαδικασία αξιολόγησης

- το random_seed είχε τιμή 5 και το n_jobs είχε την τιμή -1 ενώ όλες οι δοκιμές πραγματοποιήθηκαν στο ίδιο περιβάλλον το Google Colab ώστε να διασφαλιστεί και ο παράγοντας του χρόνου και αποτελεσμάτων.
- το threshold (percentage) για την δειγματοληψία στις γραμμές ήταν στο 10% ενώ το threshold των ελάχιστων και μέγιστων εγγραφών στο -1 ώστε να ληφθεί υπόψιν μόνο η απόφαση με βάση το ποσοστό. (Κατά την διαδικασία κάποια σύνολα δεδομένων τα οποία εμφάνιζαν λίγες εγγραφές η δειγματοληψία έγινε με πάνω από 10% στις γραμμές καθώς η χρήση λιγοστών εγγραφών δεν πραγματοποιεί σωστή εκπαίδευση, underfitting).
- το threshold (percentage) για την δειγματοληψία στις στήλες ήταν στο 80% ενώ το threshold των ελάχιστων και μέγιστων στηλών στο -1 ώστε να ληφθεί υπόψιν μόνο η απόφαση με βάση το ποσοστό.
- το threshold για τις ελλιπείς τιμές ήταν στο 20% με τις γραμμές και στήλες που εμφανίζουν ελλιπής τιμές πάνω από αυτό το ποσοστό να διαγράφονται ενώ κάτω από αυτό το ποσοστό να γίνονται impute χρησιμοποιώντας γραμμικά μοντέλα. Συγκεκριμένα χρησιμοποιήθηκε linear regression για πρόβλεψη numeric τιμών ενώ χρησιμοποιήθηκε logistic regression για πρόβλεψη string τιμών.

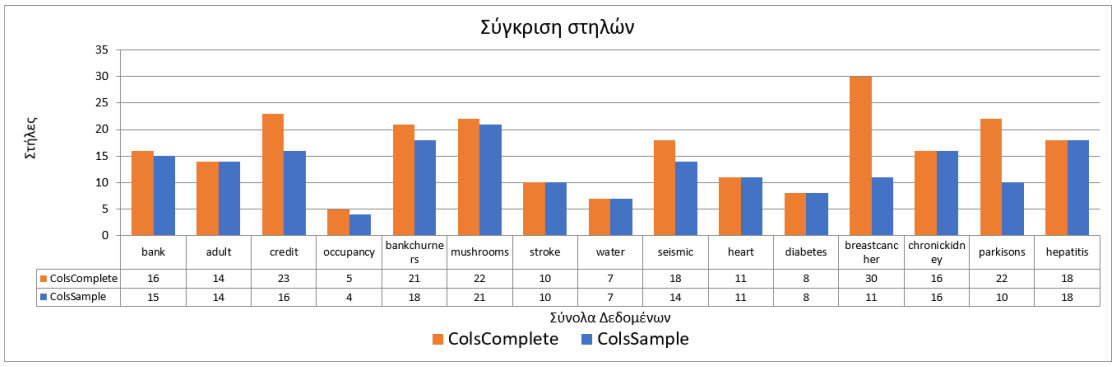
7.1 Αξιολόγηση Συνόλων

Παρακάτω, παρουσιάζονται τα χαρακτηριστικά όπως το μέγεθος της δειγματοληψίας σε γραμμές και στήλες ανά σύνολο δεδομένων για τα binary classification σύνολα δεδομένων. Με πορτοκαλί ο αριθμός σε όλο το σύνολο δεδομένων και με μπλε ο αριθμός στο σύνολο δειγματοληψίας. Τα δέκα από τα δεκαπέντε σύνολα που χρησιμοποιήθηκαν εμφάνιζαν λιγότερες από χίλιες εγγραφές στο sample λόγο της δυσεύρετης αναζήτησης μεγάλων συνόλων ενώ σε τρία από αυτά χρησιμοποιήθηκε μεγαλύτερο ποσοστό δειγματοληψίας λόγο του υπερβολικά μικρού αριθμού των δειγμάτων.



Εικόνα 28. Classification Binary σύγκριση γραμμών

Στο παραπάνω γράφημα παρουσιάζεται η δειγματοληψία σε γραμμές για τα binary classification σύνολα δεδομένων.

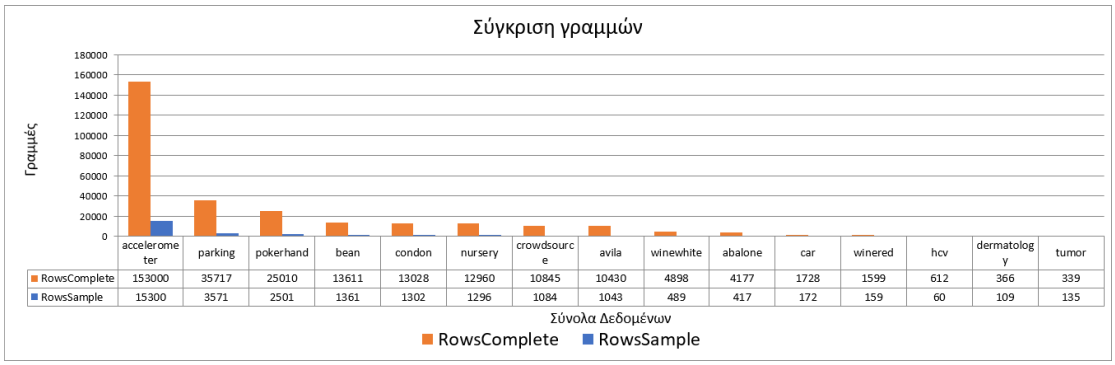


Εικόνα 29. Classification Binary σύγκριση στηλών

Στο παραπάνω γράφημα παρουσιάζεται η δειγματοληψία σε στήλες για τα binary classification σύνολα δεδομένων.

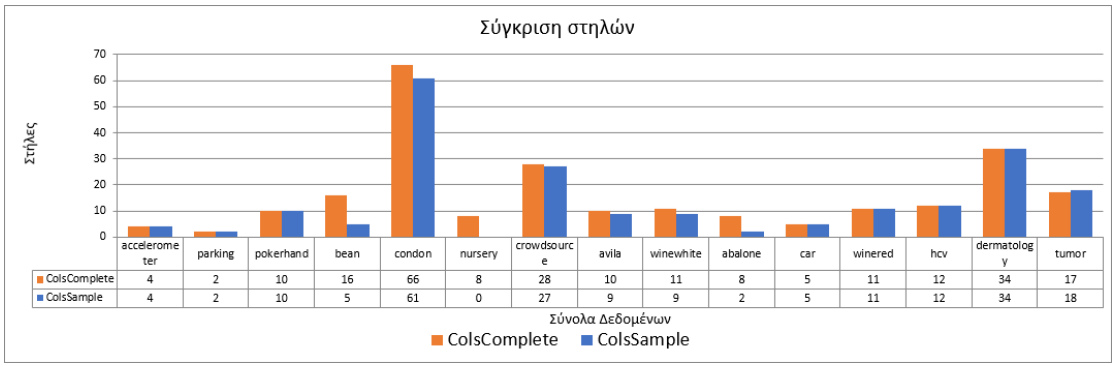
Εικόνα 30. Classification Binary εμφάνιση κενών και αλφαριθμητικών τιμών

Παρακάτω παρουσιάζονται τα χαρακτηριστικά όπως το μέγεθος της δειγματοληψίας σε γραμμές και στήλες ανά σύνολο δεδομένων για τα multiclass classification σύνολα δεδομένων. Με πορτοκαλί ο αριθμός σε όλο το σύνολο δεδομένων και με μπλε ο αριθμός στο σύνολο δειγματοληψίας. Τα επτά από τα δεκαπέντε σύνολα που χρησιμοποιήθηκαν εμφάνιζαν λιγότερες από χίλιες εγγραφές στο sample λόγο της δυσεύρετης αναζήτησης μεγάλων συνόλων ενώ σε δύο από αυτά χρησιμοποιήθηκε μεγαλύτερο ποσοστό δειγματοληψίας λόγο του υπερβολικά μικρού αριθμού των δειγμάτων.



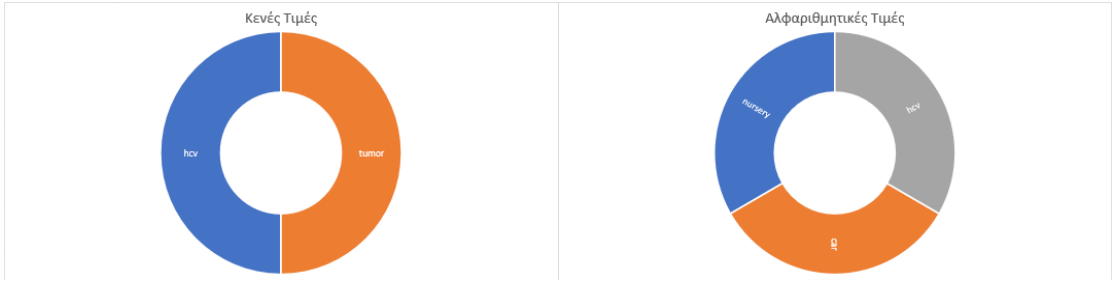
Εικόνα 31. Classification Multiclass σύγκριση γραμμών

Στο παραπάνω γράφημα παρουσιάζεται η δειγματοληψία σε γραμμές για τα multi classification σύνολα δεδομένων.



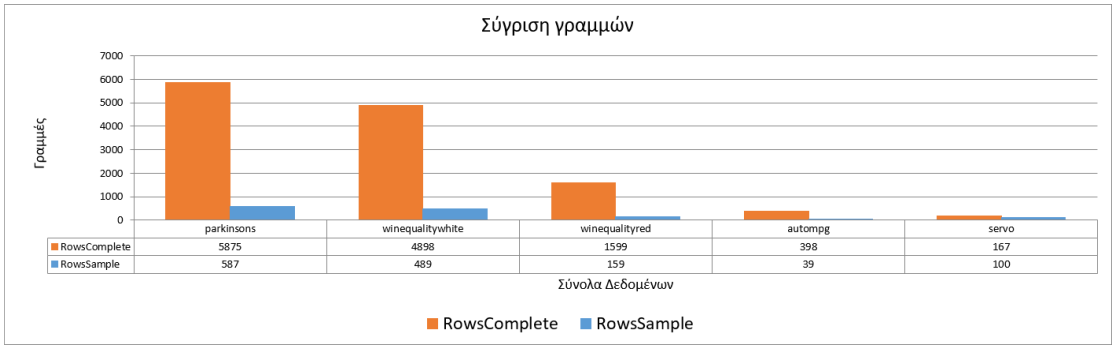
Εικόνα 32. Classification Multiclass σύγκριση στηλών

Στο παραπάνω γράφημα παρουσιάζεται η δειγματοληψία σε στήλες για τα multi classification σύνολα δεδομένων.



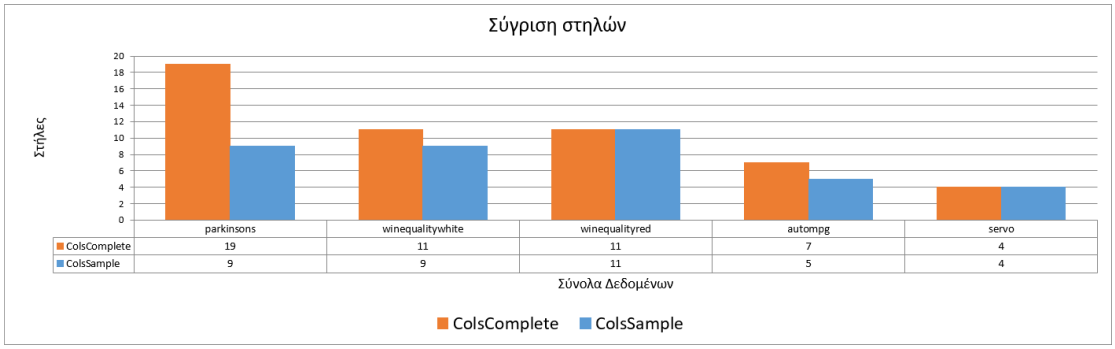
Εικόνα 33. Classification Multiclass εμφάνιση κενών και αλφαριθμητικών τιμών

Παρακάτω παρουσιάζονται τα χαρακτηριστικά όπως το μέγεθος της δειγματοληψίας σε γραμμές και στήλες ανά σύνολο δεδομένων για τα regression σύνολα δεδομένων. Με πορτοκαλί ο αριθμός σε όλο το σύνολο δεδομένων και με μπλε ο αριθμός στο σύνολο δειγματοληψίας. Όλα τα σύνολα που χρησιμοποιήθηκαν εμφάνιζαν λιγότερες από χίλιες εγγραφές στο sample λόγο της δυσεύρετης αναζήτησης μεγάλων συνόλων ενώ σε ένα από αυτά χρησιμοποιήθηκε μεγαλύτερο ποσοστό δειγματοληψίας λόγο του υπερβολικά μικρού αριθμού των δειγμάτων.



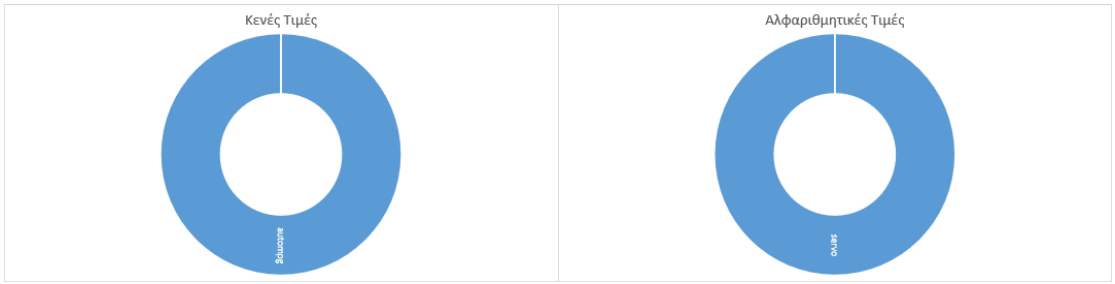
Εικόνα 34. Regression σύγκριση γραμμών

Στο παραπάνω γράφημα παρουσιάζεται η δειγματοληψία σε γραμμές για τα regression σύνολα δεδομένων.



Εικόνα 35. Regression σύγκριση στηλών

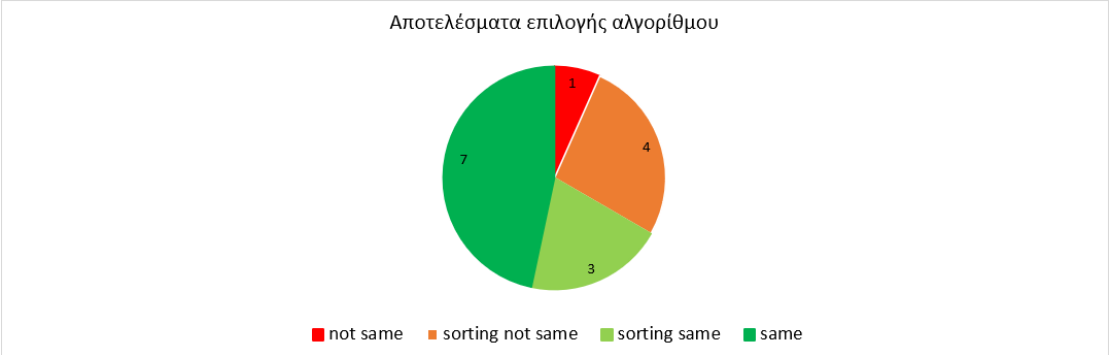
Στο παραπάνω γράφημα παρουσιάζεται η δειγματοληψία σε στήλες για τα regression σύνολα δεδομένων.



Εικόνα 36. Regression εμφάνιση κενών και αλφαριθμητικών τιμών

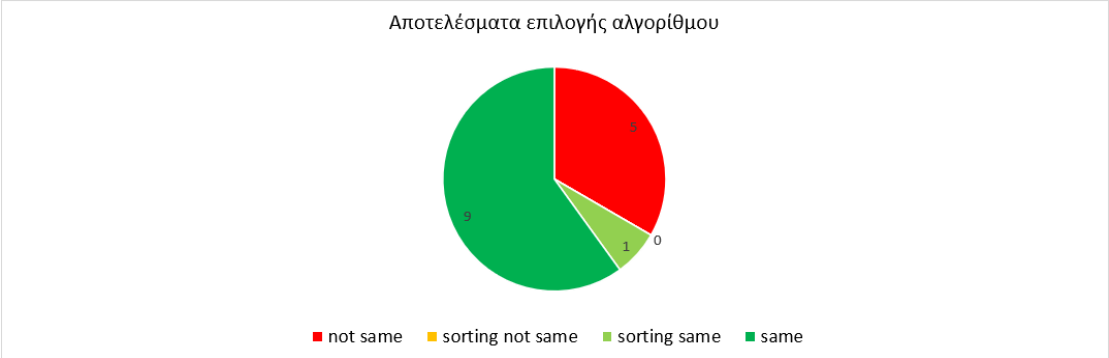
7.2 Αξιολόγηση Επιλογής Αλγορίθμου

Στα παρακάτω αποτελέσματα γίνεται εμφανές κατά πόσες φορές ο ίδιος αλγόριθμος επιλέχθηκε ανάμεσα στο sample (rows) και complete σύνολο, με σκούρο πράσινο η ίδια επιλογή, με ανοικτό πράσινο η ίδια επιλογή αλλά έχει επέλθει και ισοπαλία, με πορτοκαλί η διαφορετική επιλογή αλλά έχει επέλθει και ισοπαλία ενώ με κόκκινο η διαφορετική επιλογή.



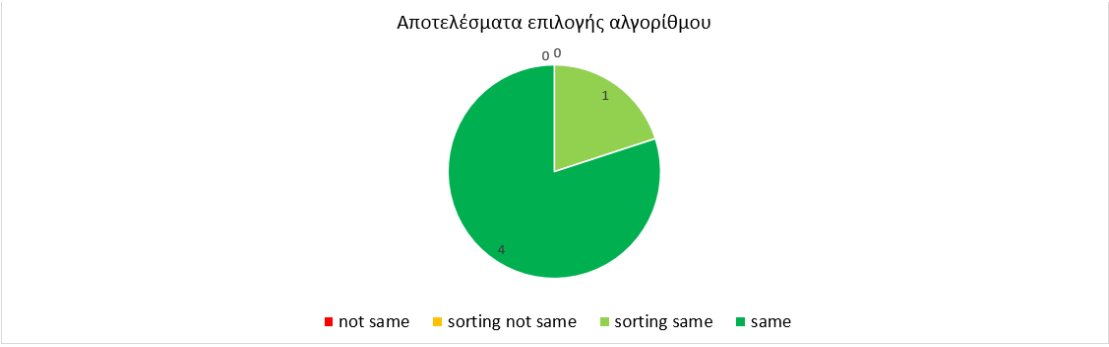
Εικόνα 37. Classification Binary επιλογή αλγορίθμου

Για τα binary classification, δέκα σύνολα δεδομένων επέλεξαν τον ίδιο αλγόριθμο (με τα τρία να παρουσίασαν ισοπαλία χωρίς απώλεια) ενώ πέντε επέλεξαν διαφορετικό (με τα τέσσερα να παρουσιάζουν ισοπαλία με απώλεια). Πιθανότητα 66,6%.



Εικόνα 38. Classification Multiclass επιλογή αλγορίθμου

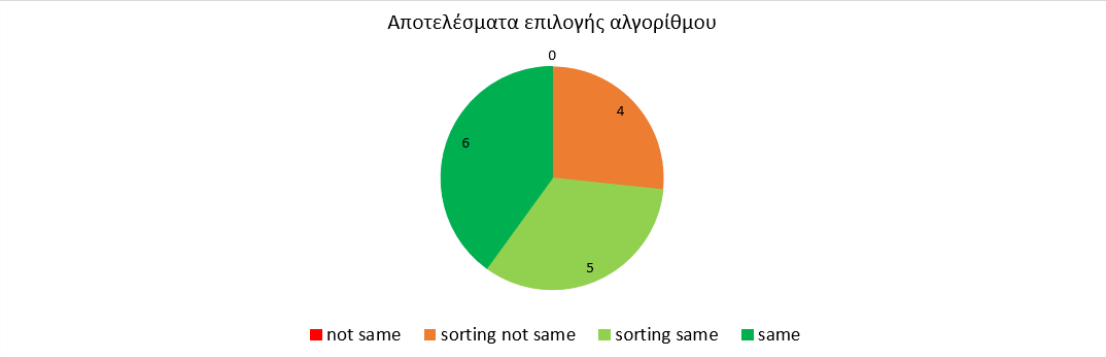
Για τα multi classification δέκα σύνολα δεδομένων επέλεξαν τον ίδιο αλγόριθμο (με το ένα να παρουσιάζει ισοπαλία χωρίς απώλεια) ενώ πέντε επέλεξαν διαφορετικό. Πιθανότητα 66,6%.



Εικόνα 39. Regression επιλογή αλγορίθμου

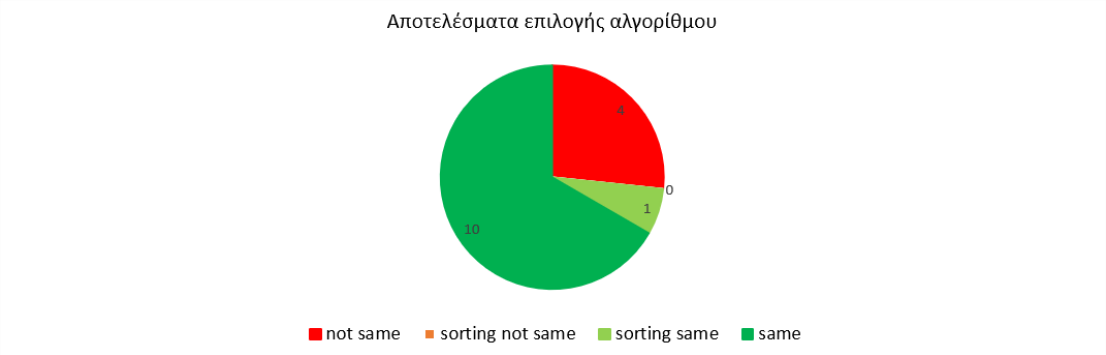
Για τα regression όλα τα σύνολα δεδομένων επέλεξαν τον ίδιο (με το ένα να παρουσιάζει ισοπαλία χωρίς απώλεια). Πιθανότητα 100%.

Στα παρακάτω αποτελέσματα γίνεται εμφανές κατά πόσες φορές ο ίδιος αλγόριθμος επιλέχθηκε ανάμεσα στο sample (rows & cols) και complete σύνολο, με σκούρο πράσινο η ίδια επιλογή, με ανοικτό πράσινο η ίδια επιλογή αλλά έχει επέλθει και ισοπαλία, με πορτοκαλί η διαφορετική επιλογή αλλά έχει επέλθει και ισοπαλία ενώ με κόκκινο η διαφορετική επιλογή.



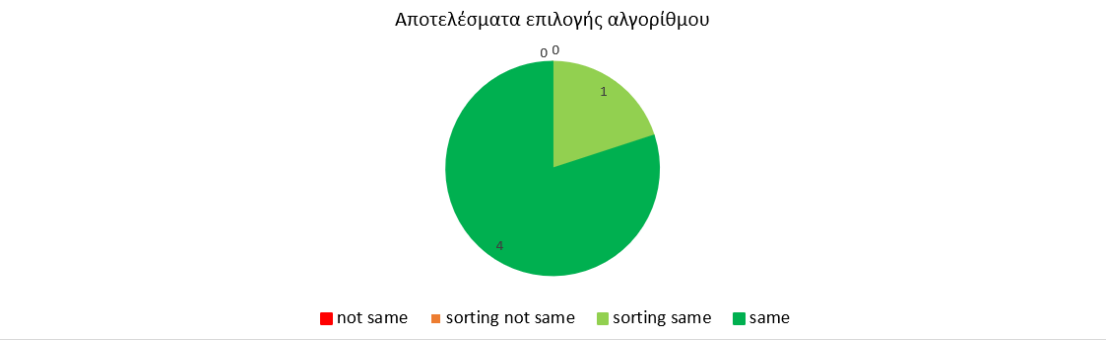
Εικόνα 40. Classification Binary επιλογή αλγορίθμου

Για τα binary classification, έντεκα σύνολα δεδομένων επέλεξαν τον ίδιο αλγόριθμο (με τα πέντε να παρουσίασαν ισοπαλία χωρίς απώλεια) ενώ τέσσερα παρουσιάζουν ισοπαλία με απώλεια). Το sampling και στις στήλες φαίνεται να διορθώνει την επιλογή δύο επιπλέον συνόλων δεδομένων. Πιθανότητα 73,3%.



Εικόνα 41. Classification Multiclass επιλογή αλγορίθμου

Για τα multi classification έντεκα σύνολα δεδομένων επέλεξαν τον ίδιο αλγόριθμο (με ένα να παρουσιάζει ισοπαλία χωρίς απώλεια) ενώ τέσσερα επέλεξαν διαφορετικό. Το sampling και στις στήλες φαίνεται να διορθώνει την επιλογή ενός επιπλέον συνόλου δεδομένων. Πιθανότητα 73,3%.

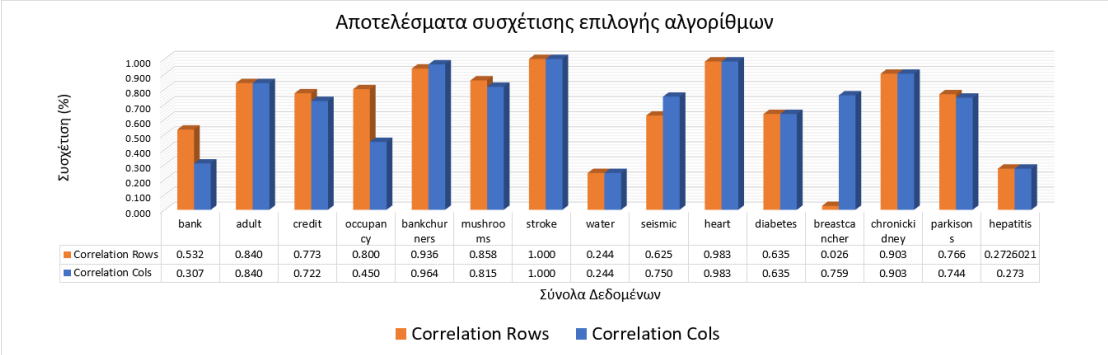


Εικόνα 42. Regression επιλογή αλγορίθμου

Για τα regression όλα τα σύνολα δεδομένων επέλεξαν τον ίδιο (με το ένα να παρουσιάζει ισοπαλία χωρίς απώλεια). Το sampling και στις στήλες δεν επηρέασε το αποτέλεσμα.

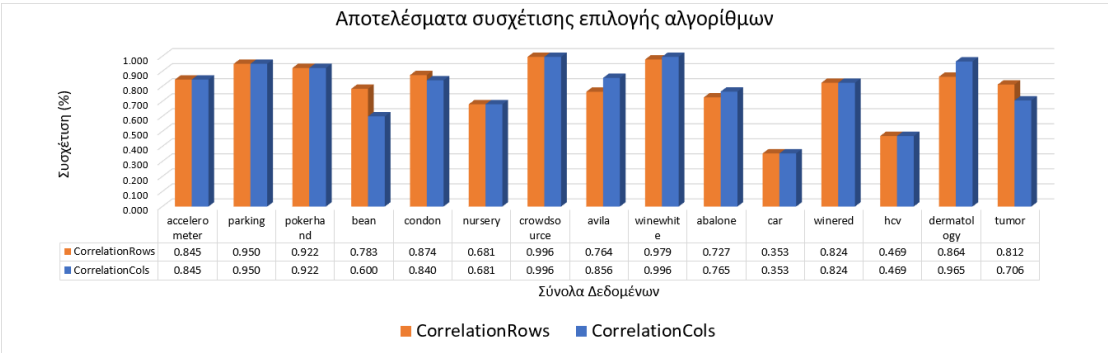
7.3 Αξιολόγηση Συσχέτισης Επιλογής Αλγορίθμων

Στα παρακάτω αποτελέσματα γίνεται εμφανές το ποσοστό συσχέτισης της σειράς επιλογής των αλγορίθμων (Spearman) ανάμεσα στο sample και complete σύνολο, με πορτοκαλί η συσχέτιση μεταξύ sample (στις γραμμές) και complete dataset ενώ με μπλε η συσχέτιση μεταξύ sample (στις γραμμές / στήλες) και complete dataset.



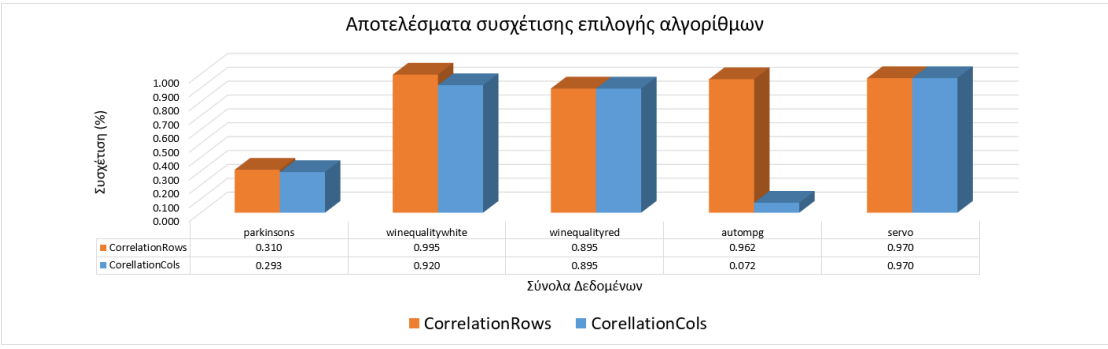
Εικόνα 43. Classification Binary συσχέτιση επιλογής αλγορίθμων

Για τα binary classification υπάρχει συσχέτιση της τάξης 70% που σημαίνει πως κατά μέσο όρο η πιθανότητα να είναι ίδια η σειρά της επιλογής των αλγορίθμων είναι αρκετά υψηλή. Το sampling και στις στήλες φαίνεται να μην παρουσιάζει μεγάλες αποκλίσεις.



Εικόνα 44. Classification Multiclass συσχέτιση επιλογής αλγορίθμων

Για τα multi classification υπάρχει συσχέτιση της τάξης 80% που σημαίνει πως κατά μέσο όρο η πιθανότητα να είναι ίδια η σειρά της επιλογής των αλγορίθμων είναι αρκετά υψηλή. Το sampling και στις στήλες φαίνεται να μην παρουσιάζει μεγάλες αποκλίσεις.

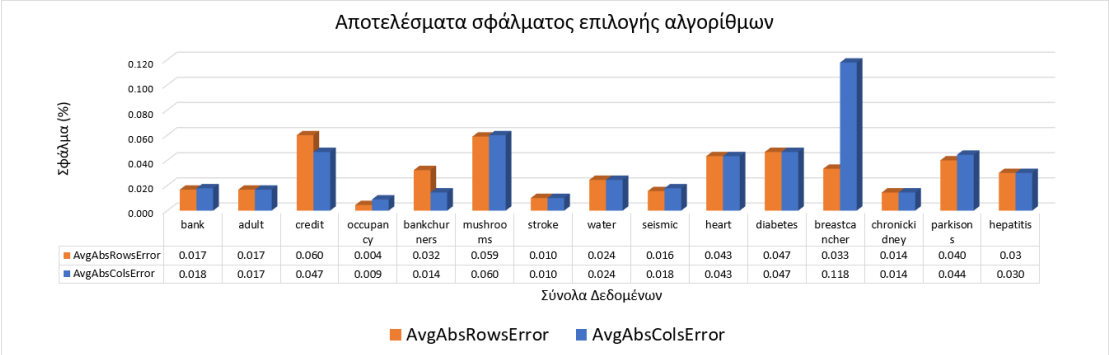


Εικόνα 45. Regression συσχέτιση επιλογής αλγορίθμων

Για τα regression υπάρχει συσχέτιση της τάξης 80% που σημαίνει πως κατά μέσο όρο η πιθανότητα να είναι ίδια η σειρά της επιλογής των αλγορίθμων είναι αρκετά υψηλή. Το sampling και στις στήλες φαίνεται να μην παρουσιάζει μεγάλες αποκλίσεις.

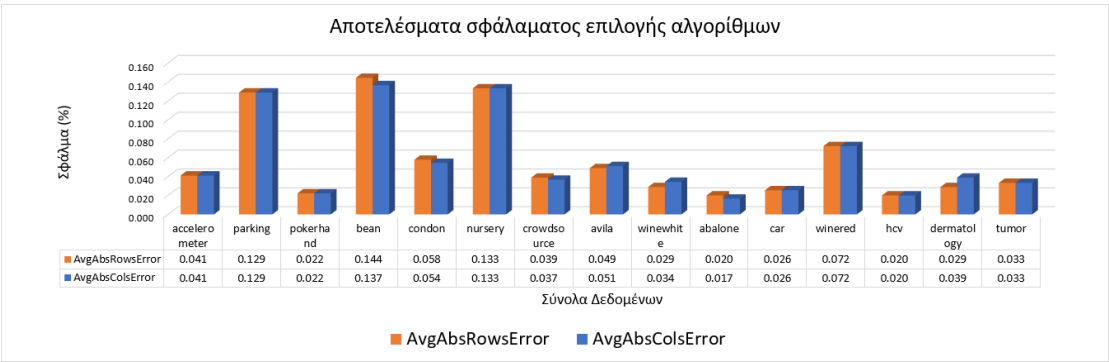
7.4 Αξιολόγηση Σφάλματος Επιλογής Αλγορίθμων

Στα παρακάτω αποτελέσματα γίνεται εμφανές το ποσοστό σφάλματος των αλγορίθμων λόγο επιλογής ανάμεσα στο sample και complete σύνολο, με πορτοκαλί το σφάλμα μεταξύ sample (στις γραμμές) και complete dataset ενώ με μπλε το σφάλμα μεταξύ sample (στις γραμμές / στήλες) και complete dataset.



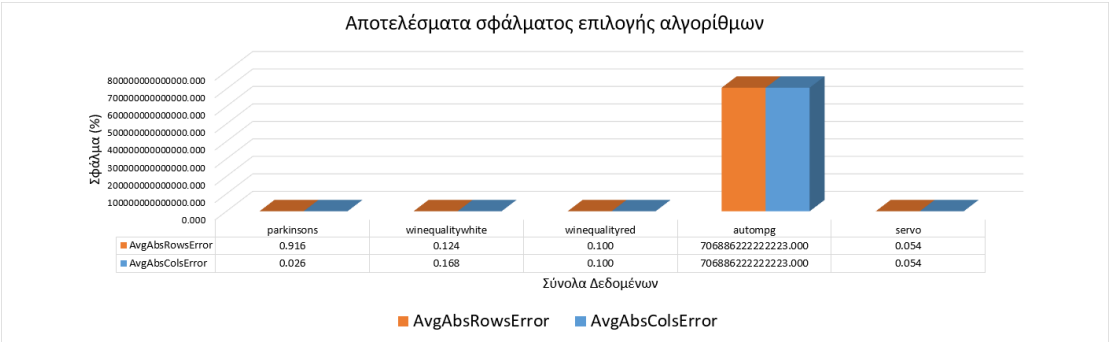
Εικόνα 46. Classification Binary σφάλμα επιλογής αλγορίθμων

Για τα binary classification παρατηρείται σφάλμα της τάξης 3% που σημαίνει πως κατά μέσο όρο το σφάλμα επικαιροποίησης μεταξύ sample και complete dataset είναι αρκετά μικρό.



Εικόνα 47. Classification Multiclass σφάλμα επιλογής αλγορίθμων

Για τα multi classification παρατηρείται σφάλμα της τάξης 5% που σημαίνει πως κατά μέσο όρο το σφάλμα επικαιροποίησης μεταξύ sample και complete dataset είναι αρκετά μικρό.

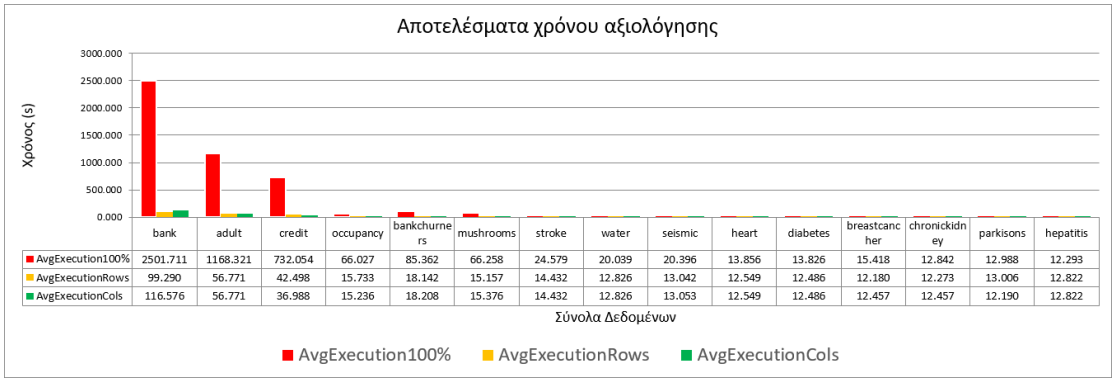


Εικόνα 48. Regression σφάλμα επιλογής αλγορίθμων

Για τα regression παρατηρείται σφάλμα μεγάλης τάξης (ενώ εξαιρουμένου ενός συνόλου στο οποίο δεν εφαρμόζεται σωστά ο γραμμικός διαχωρισμός τότε το σφάλμα γίνεται της τάξης 28%) που σημαίνει πως το σφάλμα είναι σχετικά μικρό. Το sampling και στις στήλες φαίνεται να παρουσιάζει μεγάλες και καλές αποκλίσεις αφού το σφάλμα πέφτει στο 6%.

7.5 Αξιολόγηση Χρόνου Επιλογής Αλγορίθμων

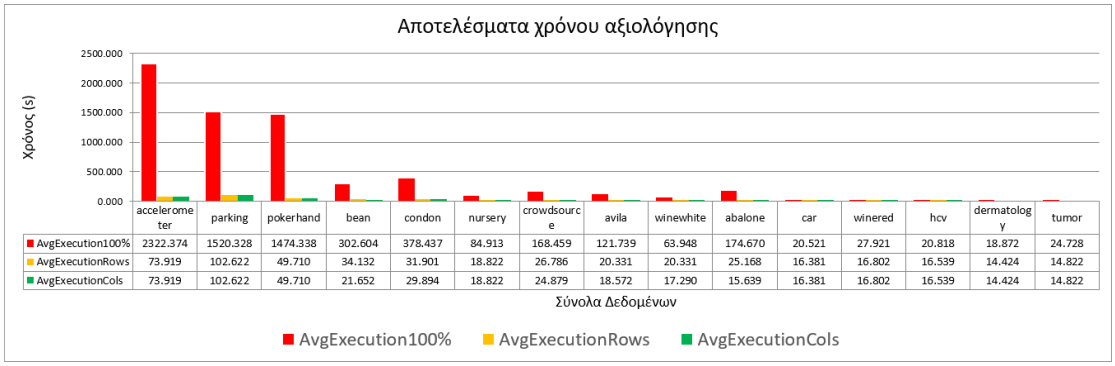
Στα παρακάτω αποτελέσματα γίνεται εμφανές η διάρκεια εκτέλεσης της επιλογής των αλγορίθμων ανάμεσα στο sample και complete σύνολο, με κόκκινο το complete dataset, με κίτρινο το sample dataset (rows) και με πράσινο το sample dataset (rows/cols).



	bank	adult	credit	occupancy	bankchurners	mushrooms	stroke	water	seismic	heart	diabetes	breastcancher	chronickidney	parkisons	hepatitis
AvgExecution100%	2501.711	1168.321	732.054	66.027	85.362	66.258	24.579	20.039	20.396	13.856	13.826	15.418	12.842	12.988	12.293
AvgExecutionRows	99.290	56.771	42.498	15.733	18.142	15.157	14.432	12.826	13.042	12.549	12.486	12.180	12.273	13.006	12.822
AvgExecutionCols	116.576	56.771	36.988	15.236	18.208	15.376	14.432	12.826	13.053	12.549	12.486	12.457	12.457	12.190	12.822

Εικόνα 49. Classification Binary χρόνος επιλογής αλγορίθμων

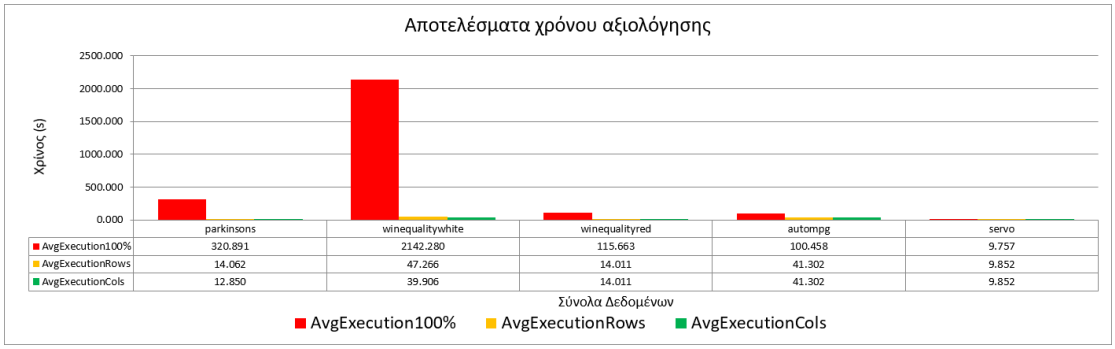
Για τα binary classification παρατηρείται μια τεράστια αύξηση για τα σύνολα δεδομένων τα οποία έχουν πάνω από 10.000 εγγραφές ανάλογα το μέγεθος του συνόλου.



	accelerometer	parking	pokerhand	bean	condon	nursery	crowdsource	avila	winewhite	abalone	car	winered	hcv	dermatology	tumor
AvgExecution100%	2322.374	1520.328	1474.338	302.604	378.437	84.913	168.459	121.739	63.948	174.670	20.521	27.921	20.818	18.872	24.728
AvgExecutionRows	73.919	102.622	49.710	34.132	31.901	18.822	26.786	20.331	20.331	25.168	16.381	16.802	16.539	14.424	14.822
AvgExecutionCols	73.919	102.622	49.710	21.652	29.894	18.822	24.879	18.572	17.290	15.639	16.381	16.802	16.539	14.424	14.822

Εικόνα 50. Classification Multiclass χρόνος επιλογής αλγορίθμων

Για τα multi classification παρατηρείται μια τεράστια αύξηση για τα σύνολα δεδομένων τα οποία έχουν πάνω από 10.000 εγγραφές ανάλογα το μέγεθος του συνόλου.



	parkinsons	winequalitywhite	winequalityred	autompg	servo
AvgExecution100%	320.891	2142.280	115.663	100.458	9.757
AvgExecutionRows	14.062	47.266	14.011	41.302	9.852
AvgExecutionCols	12.850	39.906	14.011	41.302	9.852

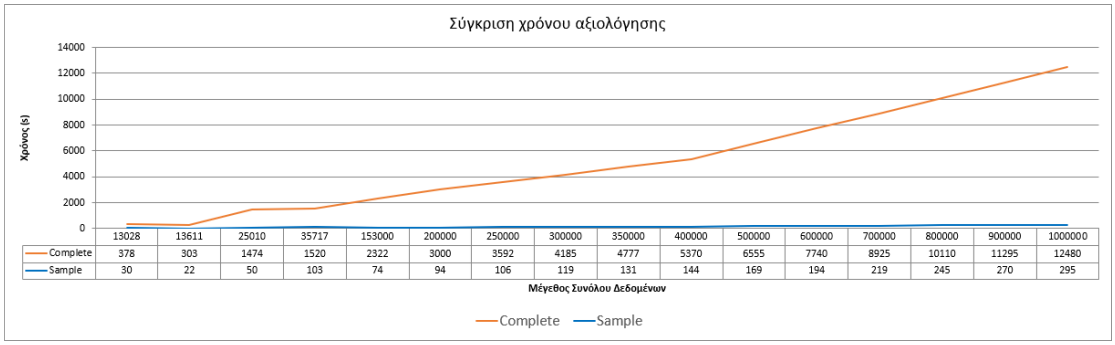
Εικόνα 51. Regression χρόνος επιλογής αλγορίθμων

Για τα regression παρατηρείται μια σχετική αύξηση για τα σύνολα δεδομένων τα οποία έχουν πάνω από 5.000 εγγραφές ανάλογα το μέγεθος του συνόλου αλλά δεν πραγματοποιήθηκαν δοκιμές με μεγαλύτερα σύνολα δεδομένων.

7.6 Αξιολόγηση Συγκριτικών Αποτελεσμάτων

Σε αυτό το σημείο, δεδομένων των αποτελεσμάτων που προέκυψαν, χρησιμοποιώντας αυτή την νέα τεχνική κατά την πειραματική διαδικασία, στην οποία φαίνεται πως η αρχική υπόθεση ότι ένα δείγμα του αρχικού συνόλου θα δουλέψει ικανοποιητικά φαίνεται να επαληθεύεται.

Επιπρόσθετα χρησιμοποιώντας τους χρόνους των πέντε μεγαλύτερων συνόλων γίνεται μία πρόβλεψη του χρόνου αναζήτησης για σύνολα δεδομένων για πάνω από 200.000 εγγραφές.

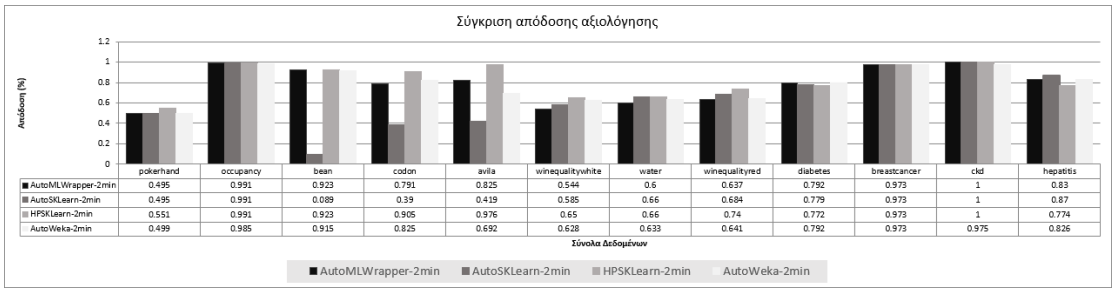


	13028	13611	25010	35717	153000	200000	250000	300000	350000	400000	500000	600000	700000	800000	900000	1000000
Complete	378	303	1474	1520	2322	3000	3592	4185	4777	5370	6555	7740	8925	10110	11295	12480
Sample	30	22	50	103	74	94	106	119	131	144	169	194	219	245	270	295

Εικόνα 52. Σύγκριση και πρόβλεψη χρόνου σε μεγαλύτερα dataset

Παρατηρούμε πως σε μεγάλα σύνολα δεδομένων ο χρόνος θα γίνει αισθητά μεγαλύτερος, όπως φαίνεται για το σύνολο με το 1 εκατομμύριο εγγραφές το οποίο θα χρειαστεί πάνω από 3,5 ώρες ενώ με sampling η αναζήτηση παραμένει εφικτή σε λογικά πλαίσια, κάτω των 5 λεπτών.

Τέλος, κρίνεται σκόπιμο να εξεταστεί πόσο καλά τα πηγαίνει σε τυχαία σύνολα δεδομένων συγκρίνοντας την απόδοση με άλλες ολοκληρωμένες λύσεις μηχανικής μάθησης. Οι μετρήσεις πραγματοποιούνται χρησιμοποιώντας τις προ-επιλεγμένες υπερ-παραμέτρους των ολοκληρωμένων λύσεων και σε χρονικό πλαίσιο των δύο λεπτών.



	pokerhand	occupancy	bean	codon	avila	winequalitywhite	water	winequalityred	diabetes	breastcancer	ckd	hepatitis
AutoMLWrapper-2min	0.495	0.991	0.923	0.791	0.825	0.544	0.6	0.637	0.792	0.973	1	0.83
AutoSKLearn-2min	0.495	0.991	0.089	0.39	0.419	0.585	0.66	0.684	0.779	0.973	1	0.87
HPSKLearn-2min	0.551	0.991	0.923	0.905	0.976	0.65	0.66	0.74	0.772	0.973	1	0.774
AutoWeka-2min	0.499	0.985	0.915	0.825	0.692	0.628	0.633	0.641	0.792	0.973	0.975	0.826

Εικόνα 53. Σύγκριση απόδοσης ανάμεσα σε ολοκληρωμένες λύσεις

Παρατηρούμε πως η AutoMLWrapper τα πάει αρκετά καλά με μικρές διαφορές στα περισσότερα σύνολα δεδομένων, ενώ παράλληλα φαίνεται να κινείται και πιο σταθερά καθώς σε όλα τα σύνολα απέδωσε ικανοποιητικά αποτελέσματα. Φυσικά το αποτέλεσμα αυτό είναι απλώς μια ένδειξη και δεν θα πρέπει να ληφθεί υπόψιν καθώς οι δύο βιβλιοθήκες έχουν διαφορετικά στάδια προ-επεξεργασίας αλλά και διαφορετική μοντελοποίηση με διαφορετικούς αλγόριθμους / υπερ-παραμέτρους. Η συγκεκριμένη μελέτη εστιάζει στο πόσο πιθανό χρησιμοποιώντας ένα δείγμα του συνόλου να ληφθούν αποφάσεις που θα λειτουργήσουν το ίδιο καλά σε ολόκληρο το σύνολο.

8. Συμπεράσματα

Τα αποτελέσματα της πειραματικής διαδικασίας υπέδειξαν πως ένα 10% είναι αναγκαίο για δειγματοληψία στις γραμμές (για σύνολα δεδομένων όπου το 10% παρουσιάζει αρκετά δείγματα). Σε αυτήν την περίπτωση το αποτέλεσμα είναι ικανοποιητικό αφού η ίδια επιλογή των αλγορίθμων με χρήση του sample έναντι του complete φτάνει στο 71.5% (με το 11,5% από το υπόλοιπο 28,5% να οφείλεται σε λανθασμένη επιλογή λόγο ισοπαλίας) ενώ στην περίπτωση που ο αλγόριθμος επιλογής δεν είναι ο ίδιος υπάρχει πιθανότητα που ξεπερνά το 76.5% να είναι ο αμέσως καλύτερος.

Επιπρόσθετα αποτελέσματα υπέδειξαν πως και ένα 80% είναι αρκετό για δειγματοληψία στις στήλες (για σύνολα δεδομένων όπου το 80% παρουσιάζει αρκετά χαρακτηριστικά) με βάση την συσχέτιση. Σε αυτήν την περίπτωση το αποτέλεσμα είναι ικανοποιητικό αφού η ίδια επιλογή των αλγορίθμων με χρήση του sample έναντι του complete φτάνει στο 80% (με το 8,5% από το υπόλοιπο 20% να οφείλεται σε λανθασμένη επιλογή λόγο ισοπαλίας) ενώ στην περίπτωση που ο αλγόριθμος επιλογής δεν είναι ο ίδιος υπάρχει πιθανότητα που ξεπερνά το 70.1% να είναι ο αμέσως καλύτερος.

Ένα από τα πιο σημαντικά ευρήματα είναι η απόδοση σε χρόνο και απαιτήσεις συστήματος, καθώς στο ίδιο σύστημα ο μέσος χρόνος αναζήτησης του βέλτιστου αλγορίθμου / υπερ-παραμέτρων είναι έως και 15-20 φορές ταχύτερος όταν γίνεται στο sample σύνολο δεδομένων ενώ όταν αφορά, συγκεκριμένα, μεγάλα σύνολα δεδομένων, τότε ο λόγος είναι της τάξης 20-25 φορές. Αυτό σημαίνει πως ο χρόνος ή οι απαιτήσεις του συστήματος αυξάνονται με εκθετικό ρυθμό με την αύξηση των δεδομένων.

Συνοπτικά θα μπορούσε να ειπωθεί πως αν σε ένα μικρότερο σύνολο δεδομένων (δηλαδή με την επιλογή λιγότερων γραμμών και στηλών με έναν έξυπνο τρόπο) ληφθεί η απόφαση για την χρήση ενός αλγορίθμου, τότε αυτός ο αλγόριθμος είναι αρκετά πιθανό να λειτουργήσει το ίδιο καλά και σε όλο το σύνολο δεδομένων. Ως αποτέλεσμα, η διαδικασία ψηφοφορίας και επιλογής των αλγορίθμων γίνεται σε συντομότερο χρονικό διάστημα συγκριτικά με την χρήση ολόκληρου του συνόλου.

9. Μελλοντικές Επεκτάσεις

Η συγκεκριμένη έρευνα παρουσίασε μια συγκεκριμένη μεθοδολογία αλλά και πειραματική διαδικασία η οποία αποδεικνύει την υπόθεση ότι χρησιμοποιώντας ένα υποσύνολο δεδομένων και εξάγοντας κάποιες αποφάσεις για αυτό, οι αποφάσεις αυτές όπως η επιλογή αλγορίθμου λειτουργούν το ίδιο καλά σε όλο το σύνολο δεδομένων. Στην συγκεκριμένη έρευνα η πειραματική διαδικασία πραγματοποιήθηκε με 10% δειγματοληψία στις γραμμές με τυχαίο τρόπο και με 80% στις στήλες χρησιμοποιώντας την συσχέτιση.

Θα ήταν ενδιαφέρον να ερευνηθεί μελλοντικά η συμπεριφορά της μεθοδολογίας:

1. σε μεγαλύτερο ποσοστό για τις γραμμές (όπως 20%, 30%).
2. σε μεγαλύτερο ποσοστό για τις στήλες (όπως 85%, 90%).

Έτσι η συγκριτική μελέτη της απόδοσης σε χρόνο και της προβλεπτικής ικανότητας θα ήταν ενδιαφέρουσα σε μεγαλύτερα ποσοστά δειγματοληψίας πόσο μάλλον ο συνδυασμός δειγματοληψίας γραμμών και στηλών.

Τέλος ένα πρόβλημα που χρήζει έρευνας θα ήταν η αντιμετώπιση ισοπαλιών με διαφορετικό τρόπο καθώς η επίλυση αυτού θα μπορούσε να επιφέρει ακόμα καλύτερα αποτελέσματα της τάξης του 90%. Πιο συγκεκριμένα, τα αποτελέσματα αξιολόγησης για κάθε σύνολο δεδομένων θα μπορούσαν να μετατραπούν σε ένα καινούριο σύνολο δεδομένων προς ανάλυση και επιλογή από μηχανική μάθηση. Για παράδειγμα στον παρακάτω πίνακα φαίνεται η αξιολόγηση του hepatitis με sample 20% για στις γραμμές και 80% στις στήλες. Παρατηρείται πως τέσσερις αλγόριθμοι βρέθηκαν σε ισοπαλία με τον PassiveAggressiveClassifier να επιλέγεται λόγο χρόνου, ενώ το LogisticRegression ήταν το πρώτο για το πλήρες σύνολο δεδομένων.

hepatitis	model_ordr	model_scra	model_scrb	model_dura
RandomForestClassifier	9	0.8	0.8	67.49
DecisionTreeClassifier	5	0.8	0.8	2.47
KNeighborsClassifier	3	1	0.83	8.34
SVC	6	0.8	0.8	3.5
SGDClassifier	8	0.8	0.8	4.74
RidgeClassifier	7	0.95	0.8	4.53
LogisticRegression	4	0.83	0.83	18.99
Perceptron	2	0.81	0.83	3.83
PassiveAggressiveClassifier	1	0.87	0.83	1.51

Πίνακας 5. Χαρακτηριστικά αξιολόγησης

Σημείωση:

* model_scra, το train score (acc), model_scrb, το test score (acc), model_dura, το fit duration (sec)

Θα μπορούσε λοιπόν να γίνει μια μελέτη με διάφορα χαρακτηριστικά αξιολόγησης και να επιλεχθεί ο κατάλληλος αλγόριθμος ανάμεσα στους αντιπάλους για την πρώτη θέση με άλλα κριτήρια όπως η τυπική απόκλιση του training / testing, ή ακόμη και κάποιου είδους successive clustering το οποίο θα ανίχνευε την καλύτερη ομάδα αλγορίθμων για την επόμενη φάση που ίσως πραγματοποιούνταν μεγαλύτερο ποσοστό δειγματοληψίας και στο τέλος θα έδινε την καλύτερη δυνατή επιλογή γρηγορότερα σε σύγκριση με το successive halving καθώς θα εξυπηρετούσε μια μέθοδο aggressive pruning. Φυσικά δεν θα μπορούσε να λείψει και ο συνδυασμός της μεθόδου meta-search για την αντιμετώπιση ισοπαλιών.

Βιβλιογραφία

Brent Komer, James Bergstra και Chris Eliasmith. 2019. *Hyperopt-Sklearn.* s.l. : Automated Machine Learning, 2019. σσ. 97-111.

Frank Hutter, Holger H. Hoos και Kevin Leyton-Brown. 2011. *Sequential Model-Based Optimization for General Algorithm Configuration.* s.l. : LION, 2011. σσ. 507-523.

Frank Hutter, Lars Kotthoff και Joaquin Vanschoren. 2019. *Automated Machine Learning - Methods, Systems, Challenges.* s.l. : Springer, 2019. 978-3-030-05317-8.

Hadi S. Jomaa, Lars Schmidt-Thieme και Josif Grabocka. 2021. Dataset2Vec: learning dataset meta-features. *Data Mining and Knowledge Discovery.* February 2021, Τόμ. 35, σσ. 964-985.

Hector Mendoza, και συν. 2019. *Towards Automatically-Tuned Deep Neural Networks.* s.l. : Automated Machine Learning, 2019. σσ. 135-149.

Iqbal H. Sarker. 2021. Machine Learning Algorithms, Real-World Applications and Research. *SN Computer Science.* 3, January 2021, Τόμ. 2, σ. 160.

Jason Brownlee. 2016. *Master Machine Learning Algo From Scratch.* s.l. : Machine Learning Mastery, 2016. 1.

Jasper Snoek, Hugo Larochelle και Ryan P. Adams. 2012. *Practical Bayesian Optimization of Machine Learning.* s.l. : NIPS, 2012. σσ. 2960-2968.

Joaquin Vanschoren. 2019. *Meta-Learning.* s.l. : Automated Machine Learning, 2019. σσ. 35-61.

Jonathan Waring, Charlotta Lindvall και Renato Umeton. 2020. Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artificial Intelligence in Medicine.* April 2020, 104, σ. 101822.

Lars Kotthoff, και συν. 2019. *Auto-WEKA: Automatic Model Selection and Hyperparameter Optimization in WEKA.* s.l. : Automated Machine Learning, 2019. σσ. 81-95.

Li Yang και Abdallah Shami. 2020. *On hyperparameter optimization of machine learning algorithms: Theory and practic.* s.l. : Neurocomputing, 2020. σσ. 295-316. Τόμ. 415.

Marion Neumann. 2019. AI profiles: an interview with Thomas Dietterich. May 2019, Τόμ. 5, σσ. 7-9.

Maroua Bahri, και συν. 2022. AutoML: State Of The Art With A Focus On Anomaly Detection, Challenges, And Research Directions. *International Journal of Data Science and Analytics.* 2022.

Matthias Feurer, και συν. 2020. Auto-Sklearn 2.0: The Next Generation. *CoRR.* CoRR 2020, Τόμ. abs/2007.04074.

Matthias Feurer, και συν. 2019. *Auto-sklearn: Efficient and Robust Automated Machine Learning.* s.l. : Automated Machine Learning, 2019. σσ. 113-134.

Petro Liashchynskyi και Pavlo Liashchynskyi. 2019. *Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS.* s.l. : CoRR, 2019. Τόμ. abs/1912.06059.

Philipp Probst, Anne-Laure Boulesteix και Bernd Bischl. 2019. *Tunability: Importance of Hyperparameters of Machine Learning Algorithms.* s.l. : J. Mach. Learn. Res., 2019. σσ. 53:1-53:32. Τόμ. 20.

Quanming Yao, και συν. 2018. Taking the Human out of Learning Applications: A Survey on Automated Machine Learning. *CoRR.* December 2018, Τόμ. abs/1810.13306.

Randal S. Olson και Jason H. Moore. 2016. *TPOT: A Tree-based Pipeline Optimization Tool for Automating Machine Learning.* s.l. : Automated Machine Learning, 2016. σσ. 66-74.

S. Banumathi και Dr. A. Aloysius. 2016. Bayesian Design of Special Type of Double Sampling Plans for Compliance Testing. *Communications in Statistics - Simulation and Computation.* August 2016, Τόμ. 45.

Sah, S. 2020. *Machine Learning: A Review of Learning Types.* s.l. : Preprints, 2020.

Thomas Dietterich. 2017. *Machine learning challenges and impact: an interview with Thomas.* May 2017.

Włodzisław Duch και Karol Grudzi ´nski. 2018. Meta-learning: searching in the model space. *CoRR.* 1806.06207, June 2018, Τόμ. abs/1806.06207.

Yang Li, και συν. 2020. *Efficient Automatic CASH via Rising Bandits.* s.l. : CoRR, 2020. Τόμ. abs/2012.04371.