



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ  
ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

Πρόβλεψη θερμοκρασίας ρότορα ηλεκτροκινητήρα  
με τεχνικές μηχανικής μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ  
ΜΑΡΓΑΡΙΤΗ ΔΑΝΑΗ

**Επιβλέπων** : Κωνσταντίνος Κουσουρής  
Επίκουρος Καθηγητής Ε.Μ.Π.

ΑΘΗΝΑ, ΣΕΠΤΕΜΒΡΙΟΣ 2020





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ  
ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

Πρόβλεψη θερμοκρασίας ρότορα ηλεκτροκινητήρα  
με τεχνικές μηχανικής μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ  
ΜΑΡΓΑΡΙΤΗ ΔΑΝΑΗ

Επιβλέπων : Κωνσταντίνος Κουσουρής  
Επίκουρος Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την Οκτωβρίου 2020

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....

.....

.....

ΑΘΗΝΑ, ΣΕΠΤΕΜΒΡΙΟΣ 2020

*(Υπογραφή)*

.....

**Δανάη Μαργαρίτη**

Διπλωματούχος Εφαρμοσμένων Μαθηματικών και Φυσικών  
Επιστημών Ε.Μ.Π.

© 2020 – All rights reserved

## Περίληψη

Ο σκοπός της διπλωματικής εργασίας ήταν η κατασκευή ενός μοντέλου πρόβλεψης της θερμοκρασίας του ρότορα ενός ηλεκτροκινητήρα. Για το σκοπό αυτό, χρειάστηκε να δοκιμαστούν διάφορα μοντέλα πρόβλεψης, από απλή γραμμική παλινδρόμηση μέχρι μοντέλα παλινδρόμησης μηχανικής μάθησης, για να επιλεγεί το τελικό μοντέλο το οποίο να έχει όσο το δυνατόν μεγαλύτερη ακρίβεια πρόβλεψης χωρίς να χάσει την ιδιότητα της γενίκευσης.

*Λέξεις Κλειδιά* : Ηλεκτροκινητήρας, Σύγχρονες Μηχανές Μόνιμου Μαγνήτη, Παλινδρόμηση, Δέντρα Απόφασης, Τεχνητά Νευρωνικά Δίκτυα



## Abstract

The scope of this thesis was the construction of a regression model that would predict the rotor temperature of a Permanent Magnet Synchronous Motor. For that purpose multiple regression predictive models were tested, from a simple linear regression to artificial intelligence models, in order to choose the most accurate and precise model without losing its generalization ability.

*Keywords* : Electric Motor, Permanent Magnet Synchronous Motor, Regression, Decision Trees, Artificial Neural Networks





# Περιεχόμενα

## 1. Εισαγωγή

- 1.1 Εισαγωγή στη Μηχανική Μάθηση
- 1.2 Εισαγωγή στα Δεδομένα του Προβλήματος
- 1.3 Περιγραφή Μεταβλητών του Συνόλου Δεδομένων

## 2. Μέθοδοι που χρησιμοποιήθηκαν

- 2.1 Γραμμική Παλινδρόμηση
- 2.2 Μεθοδολογίες Μηχανικής Μάθησης
  - 2.2.1 Υπερπροσαρμογή Δεδομένων (Overfitting)
  - 2.2.2 K-Κοντινότεροι Γείτονες
  - 2.2.3 Δέντρα Απόφασης
  - 2.2.4 AdaBoost (Adaptive Boosting)
  - 2.2.5 Gradient Boosting
  - 2.2.6 Ομοιότητες & Διαφορές των μεθόδων AdaBoost, Gradient Boosting
- 2.3 Μέθοδοι Βελτιστοποίησης και Ελαχιστοποίησης
  - 2.3.1 Μέθοδος Απότομης Καθόδου
  - 2.3.2 Στοχαστική Μέθοδος Απότομης Καθόδου
- 2.4 Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Network)
  - 2.4.1 Η Έμπνευση των Νευρωνικών Δικτύων
  - 2.4.2 Τα Τεχνητά Νευρωνικά Δίκτυα
  - 2.4.3 Η Εκπαίδευση των Τεχνητών Νευρωνικών Δικτύων
  - 2.4.4 Υπερεκπαίδευση των Τεχνητών Νευρωνικών Δικτύων
  - 2.4.5 Σχεδιασμός Νευρωνικού Δικτύου και Αποτελέσματα

## 3. Επιλογή Μεταβλητών Για Την Κατασκευή Μοντέλου Πρόβλεψης

- 3.1 Σχέση Μεταβλητών

## 4. Κατασκευή Μοντέλου Πρόβλεψης κ' Αποτελέσματα

- 4.1 Γραμμική Παλινδρόμηση

4.2 Μη Γραμμική Παλινδρόμηση

4.3 Νευρωνικά Δίκτυα

4.4 Σύνοψη Αποτελεσμάτων

Βιβλιογραφία

# 1 | Εισαγωγή

## 1.1 | Εισαγωγή στη Μηχανική Μάθηση

Τα τελευταία χρόνια ο κόσμος του επιστημονικού τομέα αναφέρεται στη Μηχανική Μάθηση, για το πώς αλλάζει και θα αλλάξει τον κόσμο και για σενάρια που αφορούν τον ρόλο της στο μέλλον. Όμως, παρόλο που μόνο τα τελευταία χρόνια έχει αρχίσει να απασχολεί το κοινό, λόγω των πιο προσιτών σε αυτόν παραγώγων της, η αφετηρία της Μηχανικής Μάθησης είναι αρκετές δεκαετίες πίσω. Την δεκαετία του 1950, μία δεκαετία μετά την εφεύρεση του πρώτου ηλεκτρονικού υπολογιστή, γνωστού ως ENIAC (Electronic Numerical Integrator and Computer), ο Frank Rosenblatt, εφευρίσκει τον Perceptron, μια ηλεκτρονική συσκευή που κατασκευάστηκε σύμφωνα με τις βιολογικές αρχές και έδειξε ικανότητα εκμάθησης. Αρχικά προσομοιώθηκε σε έναν υπολογιστή IBM 704 το 1957.



Frank Rosenblatt, εργάζεται πάνω στο “perceptron”

Πρόκειται για έναν αδύναμο ταξινομητή που όταν συνδυαζόταν και με άλλους ίδιους ταξινομητές σε ένα δίκτυο, έδινε πρωτοφανή αποτελέσματα για την εποχή. Ο Perceptron τροφοδοτήθηκε με μια σειρά από κάρτες διάτρησης και μετά από 50 δοκιμές, ο υπολογιστής εκπαιδεύτηκε να διακρίνει τις κάρτες που ήταν σημειωμένες στα αριστερά από τις κάρτες που ήταν σημειωμένες στα

δεξιά. Ο Perceptron, ήταν στην πραγματικότητα το πρώτο νευρωνικό δίκτυο που εφευρέθηκε.

*Η πρώτη μηχανή που είναι ικανή να έχει μία πρωτότυπη ιδέα*  
– Frank Rosenblatt 1950

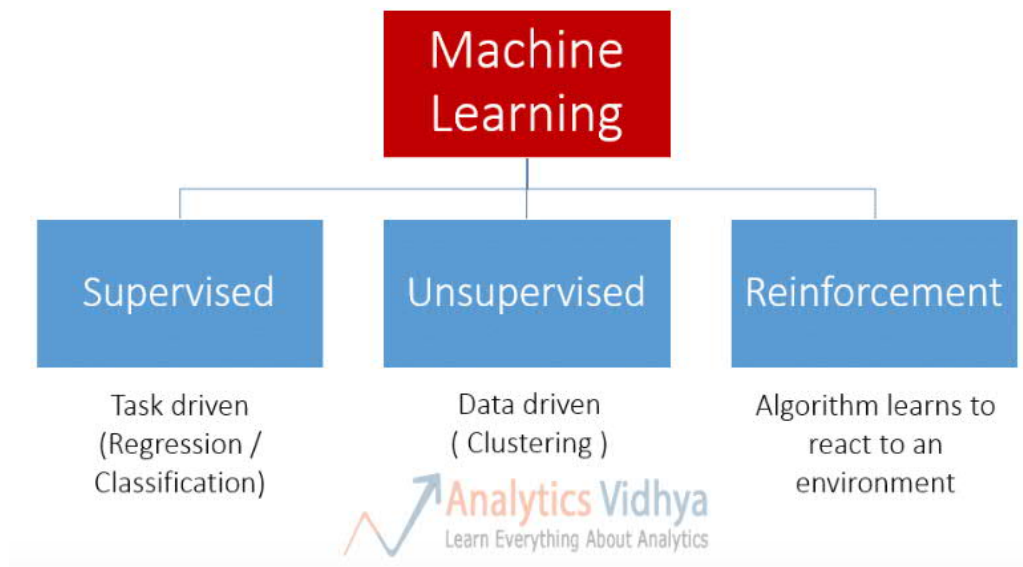
Στις αρχές του 1980, αναπτύχθηκαν τα expert systems προγράμματα που απάντησαν ερωτήματα χωρίς κάποια συγκεκριμένη γνώση αλλά χρησιμοποιώντας λογικούς κανόνες που έχουν προκύψει από τη γνώση κάποιων ειδικών. Ακολουθούν κάποια δύσκολα χρόνια για την επιστημονική εξέλιξη της τεχνητής νοημοσύνης, καθώς οι υπολογιστές της εποχής έχουν περιορισμένη μνήμη και υπολογιστική ταχύτητα και μπορούσαν να ανταπεξέλθουν μόνο στα πιο κοινά, χωρίς πραγματική αξία προβλήματα και έτσι κάθε προσπάθεια αλγορίθμου αντιμετωπιζόταν ως παιχνίδι και οι επιστήμονες δεν μπορούσαν να βρουν τις απαραίτητες χρηματικές πηγές για να συνεχίσουν τις έρευνες τους. Ευτυχώς αυτή η περίοδος κρίσης των ερευνών στον τομέα της μηχανικής μάθησης δεν κράτησε ούτε μία δεκαετία και στις αρχές της δεκαετίας του 1990, με πιο εξελιγμένους υπολογιστές και πίστη από τον επιχειρηματικό κόσμο, ο τομέας ήταν πιο δυναμικός από ποτέ.

Σύμφωνα με το αγγλικό λεξικό Oxford μηχανική μάθηση είναι η θεωρία και η ανάπτυξη υπολογιστικών συστημάτων ικανών να εκτελούν εργασίες που κανονικά απαιτούν ανθρώπινη νοημοσύνη, όπως οπτική αντίληψη, αναγνώριση ομιλίας, λήψη αποφάσεων και μετάφραση μεταξύ γλωσσών. Ο όρος της μηχανικής μάθησης περιλαμβάνει τους αλγορίθμους που παρέχουν τη δυνατότητα στους υπολογιστές να μαθαίνουν από δεδομένα και να βελτιώνουν την απόδοση και την ακρίβειά τους χωρίς προηγουμένως να έχουν προγραμματιστεί συγκεκριμένα για αυτό. Οι αλγόριθμοι λαμβάνουν δεδομένα, τα επεξεργάζονται χρησιμοποιώντας στατιστική ανάλυση και προβλέπουν την τιμή συγκεκριμένων μεταβλητών.

Μέχρι στιγμής, η μηχανική μάθηση έχει καταφέρει με επιτυχία πράγματα που πριν κάποια χρόνια ίσως να μην φανταζόμασταν. Εφαρμογές μαθαίνουν τις μουσικές προτιμήσεις του χρήστη, αλγόριθμοι προτείνουν άρθρα και διαφημίσεις σύμφωνα με τις αναζητήσεις του χρήστη, σκούπες αναγνωρίζουν τον χώρο που βρίσκονται και σκουπίζουν με το πάτημα ενός κουμπιού, αυτοκίνητα οδηγούν μόνο τους αναγνωρίζοντας λωρίδες και εμπόδια και πολλά ακόμη.

Οι αλγόριθμοι που περιλαμβάνονται στη μηχανική μάθηση ανήκουν σε τρεις κατηγορίες, την επιτηρούμενη μάθηση (supervised learning), τη μη

επιτηρούμενη μάθηση (unsupervised learning) και την ενισχυτική μάθηση (reinforcement learning).



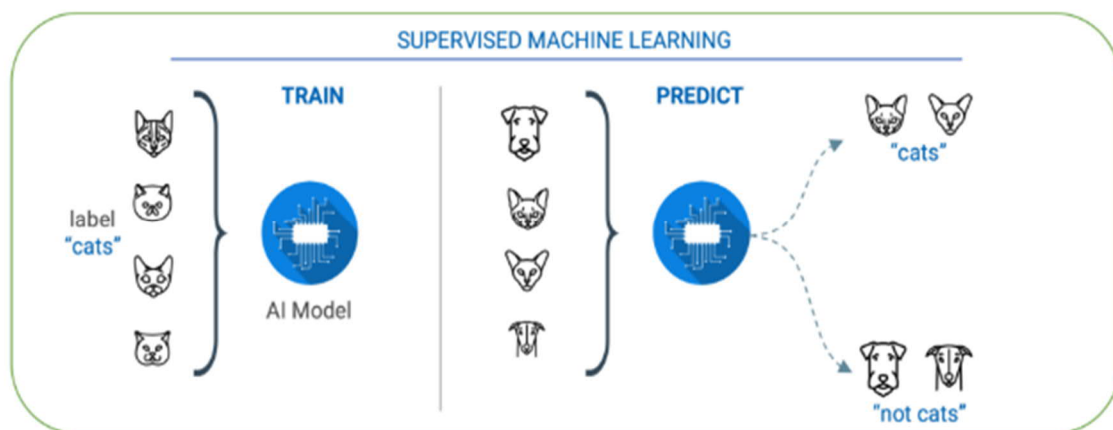
### Επιτηρούμενη Μάθηση

Η πλειοψηφία των αλγορίθμων μηχανικής μάθησης που εφαρμόζονται σε πραγματικά δεδομένα χρησιμοποιούν επιτηρούμενη μάθηση. Ονομάζεται επιτηρούμενη γιατί η διαδικασία εκμάθησης του αλγορίθμου από τα δεδομένα εκπαίδευσης θυμίζει μία δασκάλα που επιτηρεί τους μαθητές της. Οι σωστές απαντήσεις/προβλέψεις είναι γνωστές, ο αλγόριθμος επαναληπτικά προβλέπει τις απαντήσεις στα δεδομένα εκπαίδευσης και αυτές πρέπει να διορθωθούν. Η διαδικασία σταματά όταν ο αλγόριθμος πετυχαίνει μία επιτρεπτή απόδοση. Τα δεδομένα που διαχειρίζεται ένας επιτηρούμενος αλγόριθμος έχουν ετικέτα (labeled data), είναι χαρακτηρισμένα δηλαδή από μία πληροφορία, την οποία θέλουμε να προβλέψουμε. Ο σκοπός είναι να κατασκευάσει μία συνάρτηση ( $f(X)$ ) που να αντιστοιχεί ένα σύνολο δεδομένων ( $X$ ) με την ετικέτα τους ( $Y$ ). Ένα παράδειγμα επιτηρούμενης μηχανικής μάθησης είναι ένα μοντέλο που εκπαιδεύεται να αναγνωρίζει εάν τα δεδομένα αναφέρονται σε γάτα ή όχι. Δίνουμε ως δεδομένα εκπαίδευσης δεδομένα που ανήκουν σε γάτες, αυτά μπορεί να είναι εικόνες ή και χαρακτηριστικά τους σε μορφή τιμών, βάρος, ύψος, μέγεθος και το μοντέλο επιστρέφει την απάντηση του για το εάν τα

δεδομένα που δέχτηκε ανήκουν σε γάτα. Αυτό το πρόβλημα είναι τύπου ταξινόμησης.

Η επιτηρούμενη μάθηση χωρίζεται σε δύο είδη προβλημάτων, στα προβλήματα ταξινόμησης (Classification) και στα προβλήματα παλινδρόμησης (Regression). Εάν ο αλγόριθμος καλείται να προβλέψει ετικέτες που αφορούν κατηγορίες, τότε ανήκει στους αλγορίθμους ταξινόμησης, ενώ αν πρέπει να προβλέψει μία επικέτα που δέχεται συνεχείς τιμές, όπως για παράδειγμα το ύψος ενός ανθρώπου ή η κατανάλωση ρεύματος σε κιλοβατώρες ενός σπιτιού, τότε αναφερόμαστε σε πρόβλημα παλινδρόμησης.

Πηγή: mc.ai

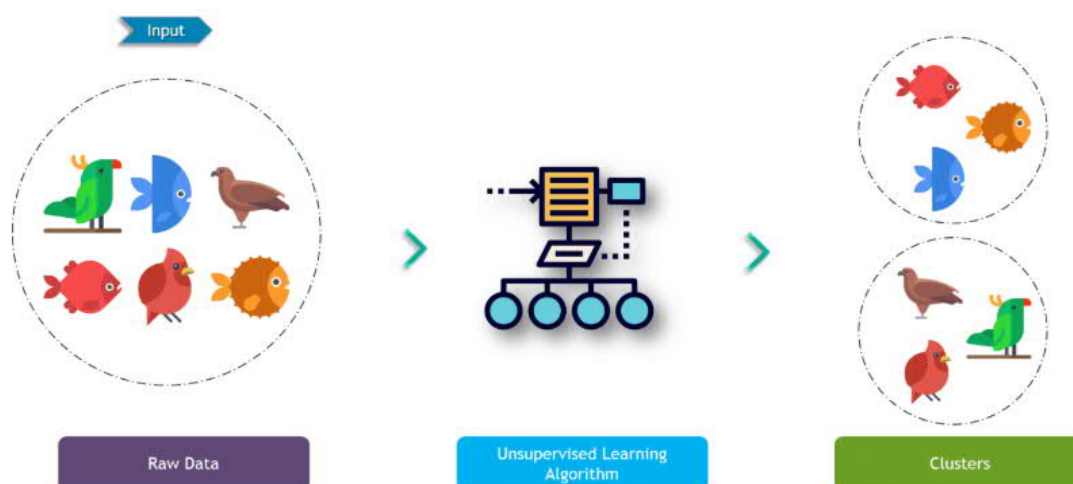


### Μη Επιτηρούμενη Μάθηση

Σε αυτή την κατηγορία ανήκουν αλγόριθμοι που δέχονται δεδομένα, που δεν είναι χωρισμένα σε κατηγορίες και προσπαθεί να διακρίνει κοινά χαρακτηριστικά μεταξύ υποομάδων των δεδομένων. Πρακτικά βρίσκει τις κατηγορίες στις οποίες ανήκουν τα δεδομένα, χωρίς να έχουν δοθεί από πριν, όπως γίνεται στην επιτηρούμενη μάθηση. Στο επόμενο παράδειγμα, εισάγουμε στον αλγόριθμο δεδομένα από διάφορα είδη ζώων, εκ των οποίων τα μισά ανήκουν στα ψάρια

και καλείται να αναγνωρίσει τα κοινά χαρακτηριστικά μεταξύ των ίδιων ζώων και να τα διαχωρίσει σε άλλη κατηγορία.

Σε αυτή την κατηγορία αλγορίθμων συναντάμε δύο τύπους, την ομαδοποίηση (Clustering) και τη συσχέτιση (Association). Το παράδειγμα που αναφερθήκαμε πριν, ανήκει στα προβλήματα ομαδοποίησης, όπου ο αλγόριθμος αναζητά ομάδες με κοινά χαρακτηριστικά. Στα προβλήματα συσχέτισης, σκοπός είναι η



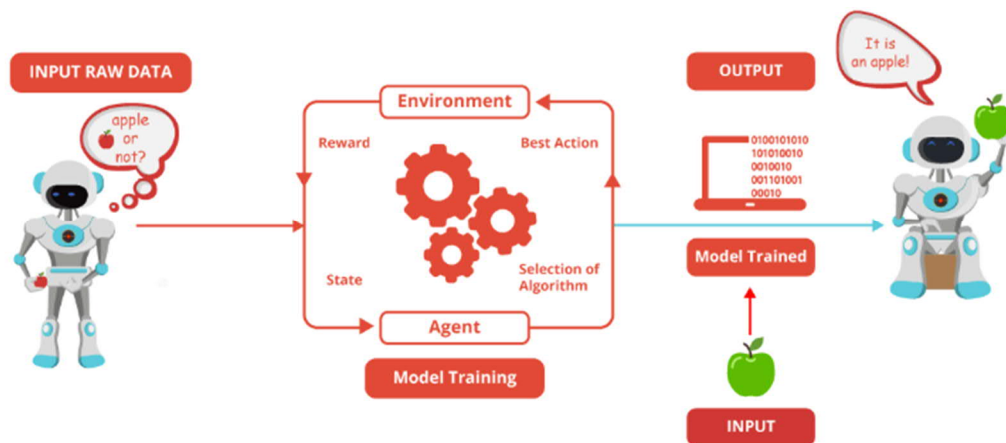
ανακάλυψη κανόνων που περιγράφουν μεγάλο μέρος των δεδομένων, όπως για παράδειγμα η συσχέτιση της αγοράς δύο προϊόντων, έχοντας εισάγει στον αλγόριθμο δεδομένα που αφορούν καταναλωτική συμπεριφορά. Μετά την χρήση μη επιτηρούμενων αλγορίθμων, χρειάζεται επαλήθευση των κατηγοριών που βρέθηκαν από τον χρήστη, καθώς ο αλγόριθμος μπορεί να βρει κοινά χαρακτηριστικά που δεν είναι χρήσιμα για την επίλυση του συγκεκριμένου προβλήματος που θέλουμε. Μία χρήσιμη λειτουργία αυτών των αλγορίθμων είναι ότι μπορούν να ανακαλύψουν ασυνήθιστα δεδομένα, κάτι το οποίο είναι χρήσιμο για εύρεση δόλιων συναλλαγών. Ένα αρνητικό αυτής της κατηγορίας σε σχέση με τους επιτηρούμενους αλγόριθμους είναι ότι έχουν πολύ μεγαλύτερη υπολογιστική δυσκολία, συνεπώς και κόστος.

### Ενισχυτική Μάθηση

Η Ενισχυτική Μάθηση αποτελεί μία ειδική κατηγορία της επιτηρούμενης μάθησης, όπου ο χρήστης δίνει το ποσοστό ακρίβειας στο οποίο επιθυμεί ο αλγόριθμος να φτάσει. Δίνονται στον αλγόριθμο δεδομένα και το επιθυμητό αποτέλεσμα πρόβλεψης και σκοπός του αλγορίθμου είναι να φτάσει την ακρίβεια της πρόβλεψης όσο πιο κοντά γίνεται στην δοσμένη απόδοση, κάτι το οποίο γίνεται μέσω διαδικασιών εξερεύνησης και 'εκμετάλλευσης'. Το ενδιαφέρον με αυτή την κατηγορία αλγορίθμων είναι πως εάν το μοντέλο φτάσει σε λάθος πρόβλεψη, τότε επιβαρύνεται με μία ποινή, έτσι απομακρύνεται από την επιθυμητή απόδοση, ενώ αν φτάσει σε σωστή απόφαση

δίνεται μια επιβράβευση. Συνεπώς, στόχος είναι η ελαχιστοποίηση των ποινών και η μεγιστοποίηση των επιβραβεύσεων, μέσω διαδοχικών αποφάσεων όπου η επόμενη εξαρτάται από τον χρήστη ή το σύστημα, καθώς πρέπει να αξιολογηθεί η απάντησή του. Εξερεύνηση είναι όταν ο αλγόριθμος μαθαίνει από τις ποινές και τις επιβραβεύσεις του και 'εκμετάλλευση' όταν εκτελεί μία ενέργεια με τη γνώση που κατέκτησε από τη διαδικασία της εξερεύνησης.

Πηγή: edureka.co



Σήμερα η ενισχυτική μάθηση χρησιμοποιείται πολύ στη ρομποτική, καθώς μπορεί να επιλύσει πολυδιάστατα προβλήματα, στην εξόρυξη κειμένου, χρησιμοποιείται στην παραγωγή περιλήψεων από μεγάλα κείμενα, αλλά και στα ηλεκτρονικά παιχνίδια, αφού οι βασικοί αλγόριθμοι που λύνουν διάφορα παιχνίδια ή εκπαιδεύουν ηλεκτρονικούς αντιπάλους, χρησιμοποιούν τη τεχνολογία της ενισχυτικής μάθησης.

## 1.2 | Εισαγωγή στα Δεδομένα του Προβλήματος

Το 1881 άρχισε να λειτουργεί η πρώτη μονάδα παραγωγής ηλεκτρικής ενέργειας στο Godalming της Αγγλίας, μεταξύ Λονδίνου και Πόρτσμουθ, με ισχύ 746 kW. Η κίνηση της γεννήτριας προερχόταν από δύο υδρόμυλους και λειτουργούσε μόνο σε εποχή κανονικών βροχοπτώσεων, γιατί δεν ήταν δυνατόν να ελεγχθεί επαρκώς η ροή νερού στο ποτάμι που διατρέχει την πόλη. Το έτος 1885 κατασκεύασε ο William Stanley (Στάνλυ, 1858-1916) υπάλληλος της εταιρείας Westinghouse, ένα επαγωγικό πηνίο ή όπως λέμε σήμερα, ένα μετασχηματιστή ισχύος, με τον οποίο μετέβαλε κατ' επιθυμία την εναλλασσόμενη τάση. Με την αξιοποίηση του μετασχηματιστή επικράτησε οριστικά το εναλλασσόμενο ρεύμα έναντι του συνεχούς.



Η ικανότητα του ηλεκτροκινητήρα είναι να μετατρέπει την ηλεκτρική ενέργεια σε μηχανική ενέργεια. Αυτό γίνεται με την αλληλεπίδραση μαγνητικών πεδίων του ηλεκτροκινητήρα με ρευματοφόρους αγωγούς, καλώδια, για να παραχθεί δύναμη. Η δύναμη αυτή υλοποιείται με τη μορφή ροπής που εφαρμόζεται στον άξονα του ηλεκτροκινητήρα.

Στις μέρες μας, ηλεκτρικούς κινητήρες συναντάμε σε πολλές εφαρμογές στη βιομηχανία (συστήματα ανύψωσης-μεταφοράς, ανεμιστήρες, αντλίες κ.α.), σε καθημερινές εφαρμογές όπως οικιακές συσκευές, μικροϋπολογιστές, εκτυπωτές κ.α. αλλά και στον τομέα της αυτοκίνησης.

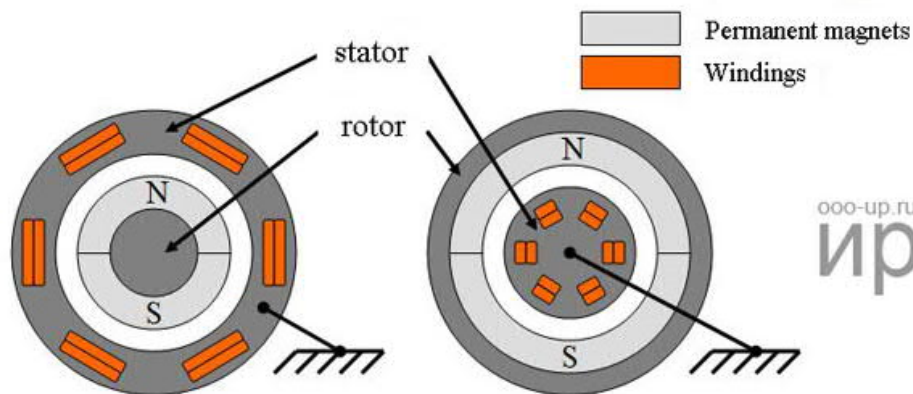
Υπάρχουν διάφορα είδη ηλεκτρικών κινητήρων, ωστόσο εμείς θα επικεντρωθούμε στις Σύγχρονες Μηχανές Μόνιμου Μαγνήτη ή ΣΜΜΜ (Permanent Magnet Synchronous Motor - PMSM) από τις οποίες προέρχονται τα δεδομένα που χρησιμοποιήθηκαν για το σκοπό της παρούσας διπλωματικής.

Οι ΣΜΜΜ έχουν στην πλειοψηφία τους κυλινδρικό σχήμα και αποτελούνται από δύο βασικά μέρη, το εξωτερικό μέρος αποτελείται από ένα σωληνωτό κυλινδρικό πλαίσιο, το οποίο, κατά κανόνα, αποτελεί το σταθερό μέρος του ηλεκτροκινητήρα και στερεώνεται στο χώρο της τοποθέτησης και το οποίο, καθώς είναι ακίνητο ονομάζεται στάτης (stator). Το άλλο κύριο μέρος βρίσκεται στο εσωτερικό το στάτη (stator) που φωλιάζει ο δρομέας (rotor), ένας κύλινδρος πάνω στον άξονα του κινητήρα που κι αυτός έχει στην εξωτερική του περιφέρεια τυλίγματα.

Ο στάτης, αποτελείται από ένα σωληνωτό κυλινδρικό πλαίσιο. Στο εσωτερικό του, σε κατάλληλα διαμορφωμένα αυλάκια παράλληλα με τον άξονα του, υπάρχουν ηλεκτρικοί αγωγοί, τα τυλίγματα/περιελίξεις (windings), που καταλήγουν σε ακροκιβώτιο σύνδεσης στο εξωτερικό μέρος. Η εξωτερική επιφάνεια του στάτη φέρει συνήθως πτερύγια για την καλύτερη ψύξη με φυσική κυκλοφορία αέρα ή περιβάλλεται από ειδικό πλαίσιο μέσα στο οποίο ρέει ψυκτικό υγρό (π.χ. νερό) στην περίπτωση που ο ηλεκτροκινητήρας είναι υδρόψυκτος.

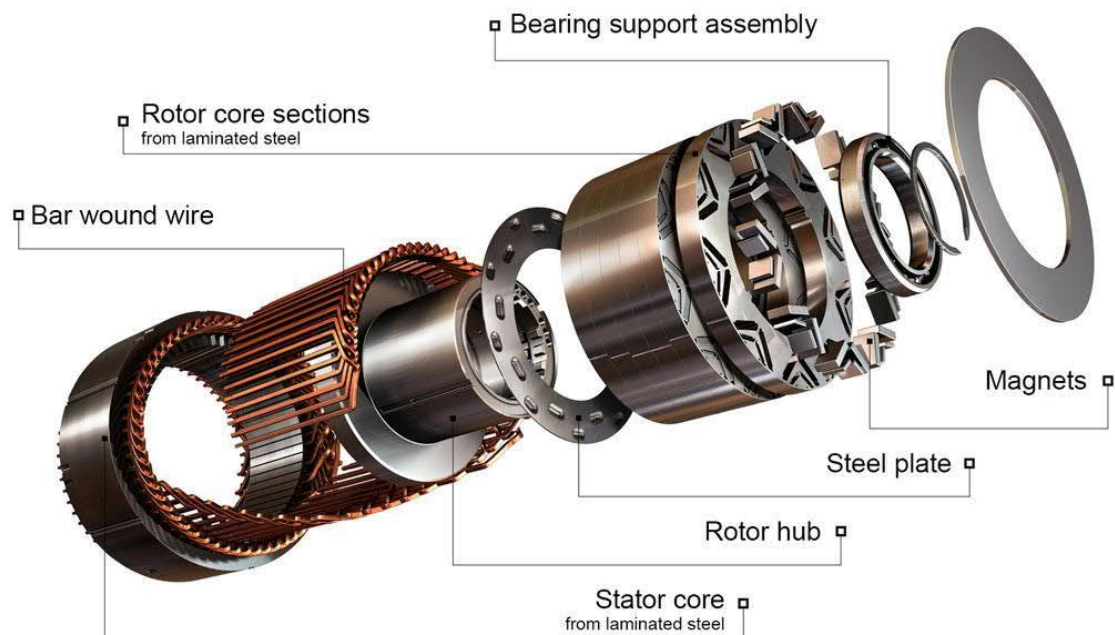
Ο ρότορας (rotor), είναι ένας κύλινδρος πάνω στον άξονα του κινητήρα που είτε έχει στην εξωτερική του επιφάνεια τυλίγματα είτε μόνιμους μαγνήτες (Μόνιμοι μαγνήτες ονομάζονται οι μαγνήτες που διατηρούν τις μαγνητικές τους ιδιότητες για μεγάλο χρονικό διάστημα και είναι συνήθως κατασκευασμένοι από χάλυβα ή ειδικά μαγνητικά κράματα). Ο ρότορας έχει ένα λεπτό διάκενο αέρα μεταξύ της εξωτερικής του επιφάνειας και του στάτη. Η τάξη μεγέθους του διακένου είναι συνήθως κλάσματα του χιλιοστού. Μόνο σε πολύ μεγάλους κινητήρες μπορεί το διάκενο να είναι μερικά χιλιοστά.

Η σχετική θέση των δύο βασικών μερών αλλάζει από κινητήρα σε κινητήρα ανάλογα με το σκοπό και την εφαρμογή του. Συνήθως, ο ρότορας βρίσκεται στο εσωτερικό τμήμα του στάτη, αλλά υπάρχουν και διατάξεις που ο ρότορας περιβάλλει το στάτη, όπως βλέπουμε την παρακάτω εικόνα:

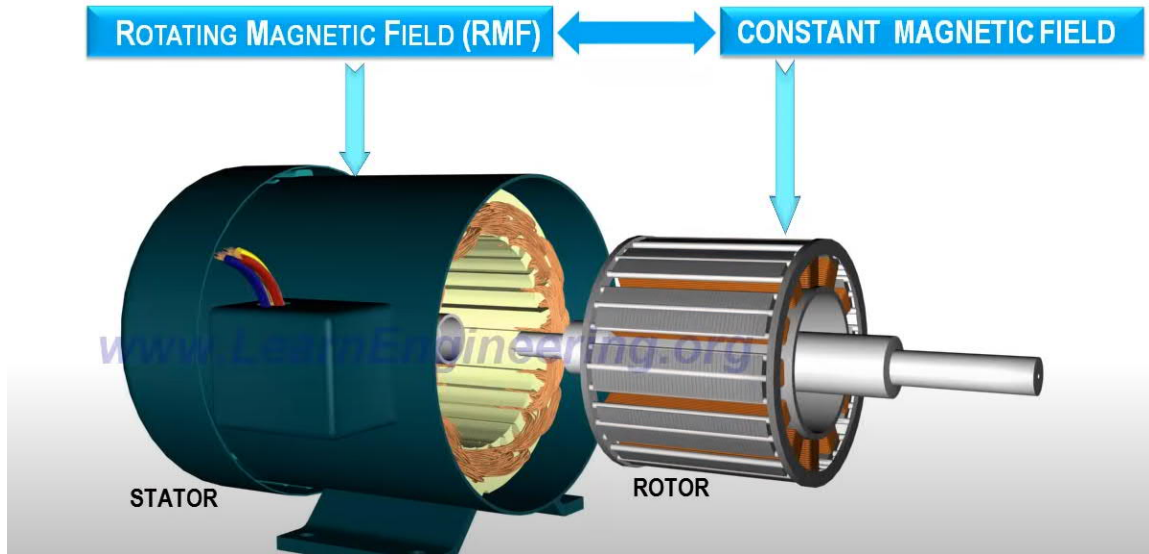


Τόσο ο στάτης όσο και ο δρομέας είναι κατασκευασμένοι από σιδηρομαγνητικό υλικό, έτσι ώστε να παρέχουν κατάλληλο δρόμο για την μαγνητική ροή. Οι περιελίξεις του στάτη και του δρομέα, επειδή είναι φορείς τάσης και ρεύματος και οι πυρήνες του στάτη και του δρομέα, επειδή είναι φορείς της μαγνητικής ροής αποτελούν τα ενεργά μέρη του κινητήρα.

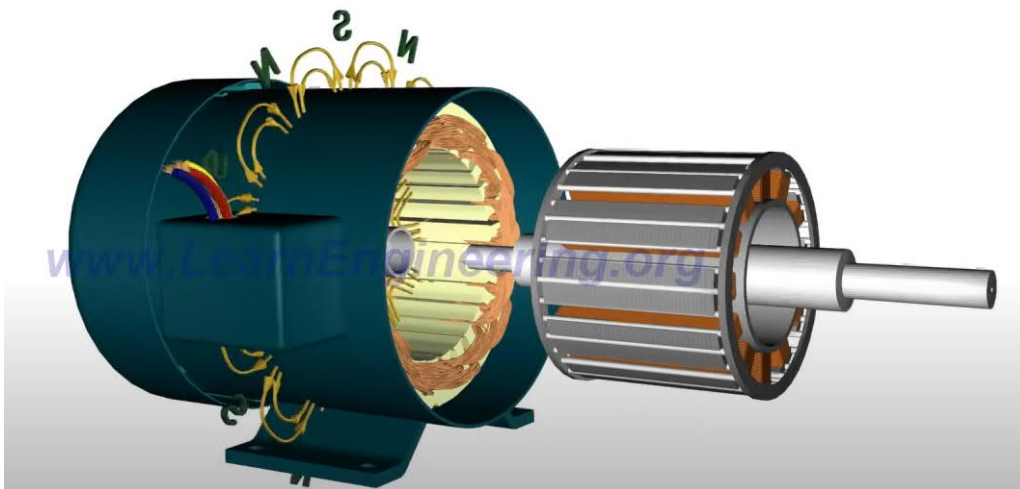
Στα παρακάτω σχήματα βλέπουμε αναλυτικά την εξωτερική και εσωτερική δομή ενός ΣΜΜ αντίστοιχα.



Το βασικό χαρακτηριστικό των ΣΜΜ είναι ότι έχουν μεγάλη απόδοση και ότι μπορούν να λειτουργούν και να διατηρούν μια σταθερή ταχύτητα ανεξάρτητα από το φόρτο που επιδρά πάνω τους. Η σταθερή ταχύτητα επιτυγχάνεται μέσω της αλληλεπίδρασης μεταξύ ενός σταθερού και ενός περιστρεφόμενου μαγνητικού πεδίου. Ο ρότορας ενός σύγχρονου κινητήρα παράγει σταθερό μαγνητικό πεδίο και ο στάτης παράγει περιστρεφόμενο μαγνητικό πεδίο.

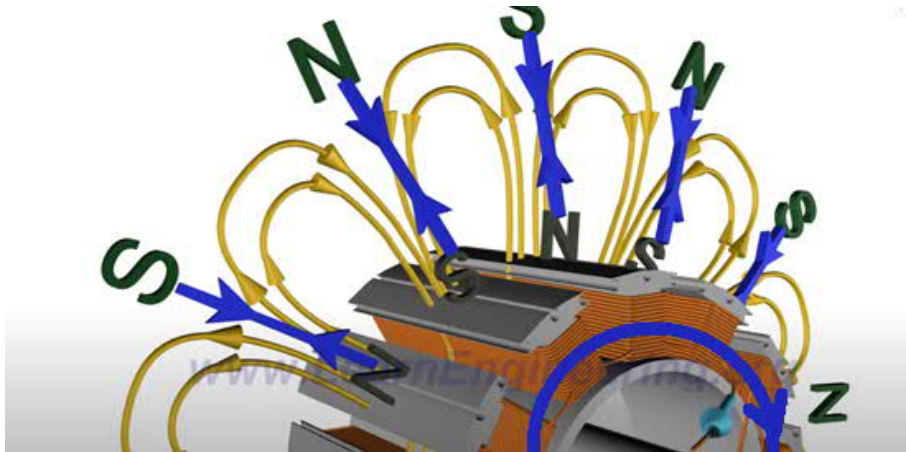


Ο στάτης διεγείρεται από μια τριφασική παροχή AC και κατά συνέπεια παράγει ένα περιστρεφόμενο μαγνητικό πεδίο που περιστρέφεται με σύγχρονη ταχύτητα.



Ο ρότορας διεγείρεται από μια τροφοδοσία DC και έτσι λειτουργεί σαν μόνιμος μαγνήτης. Σε πολλές περιπτώσεις, ο ρότορας μπορεί να είναι κατασκευασμένος από μόνιμο μαγνήτη. Η αλληλεπίδραση του ρότορα και του περιστρεφόμενου μαγνητικού πεδίου (Rotating Magnetic Field, RMF) λειτουργεί όπως περιγράφεται παρακάτω:

Ας υποθέσουμε ότι δίνεται μια αρχική περιστροφή στον ρότορα με κατεύθυνση ίδια με αυτή του RMF. Οι αντίθετοι πόλοι του RMF και του ρότορα θα προσελκύσουν ο ένας τον άλλον και θα κλειδώσουν μαγνητικά.

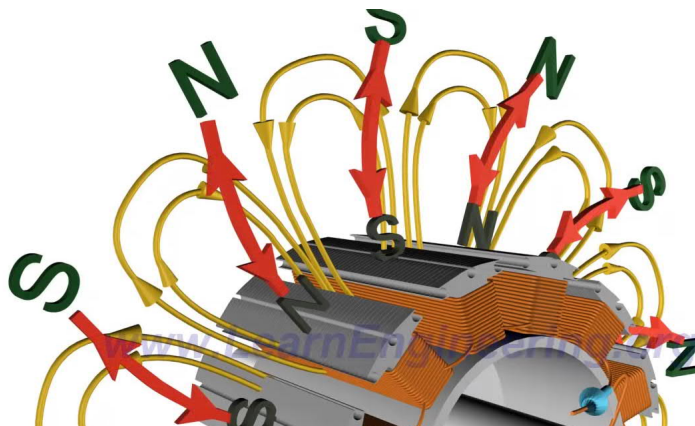


Αυτό σημαίνει ότι είτε ο ρότορας θα περιστρέφεται με την ίδια ταχύτητα με το RMF, είτε θα περιστρέφεται με σύγχρονη ταχύτητα. Η σύγχρονη ταχύτητα μπορεί εύκολα να προκύψει από την παρακάτω σχέση:

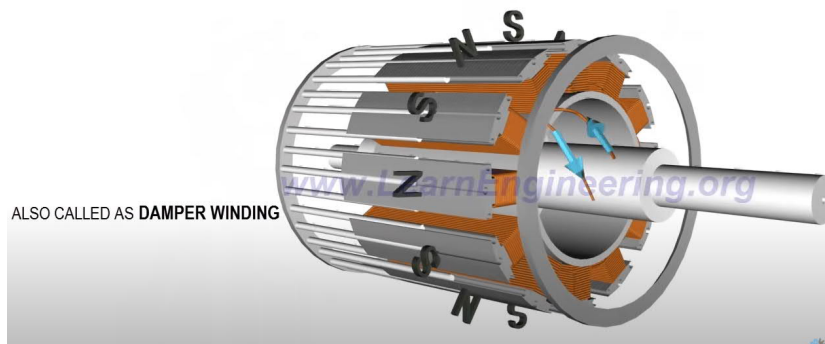
$$N_s = 120fP$$

όπου  $N_s$  η σύγχρονη ταχύτητα σε (rpm),  $f$  η συχνότητα του ηλεκτρικού ρεύματος και  $P$  ο αριθμός των πόλων. Αυτό σημαίνει ότι ελέγχοντας τη συχνότητα του ηλεκτρικού ρεύματος μπορούμε να ελέγξουμε με ακρίβεια την ταχύτητα του κινητήρα. Στην περίπτωση όμως που ο ρότορας δεν έχει αρχική περιστροφή, η κατάσταση είναι διαφορετική.

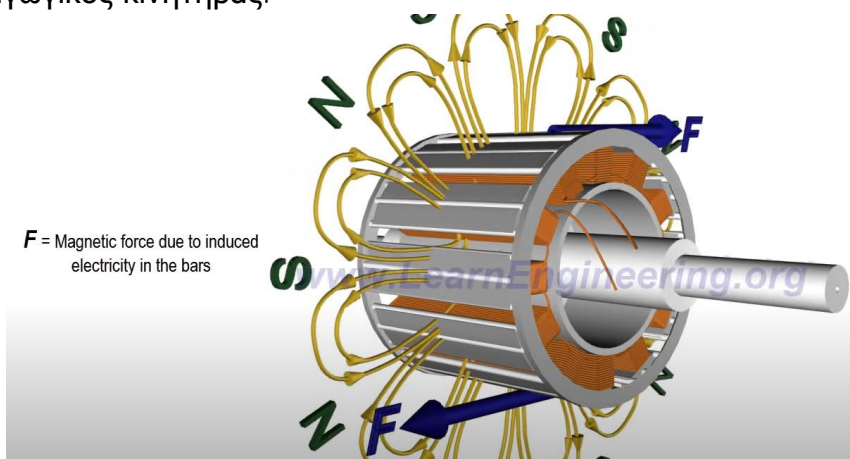
Ο βόρειος πόλος του ρότορα προσελκύεται από τον νότιο πόλο του RMF και αρχίζει να κινείται προς την ίδια κατεύθυνση, αλλά επειδή ο ρότορας έχει κάποια αδράνεια, αυτή η ταχύτητα εκκίνησης θα είναι πολύ χαμηλή. Έτσι ο νότιος πόλος του RMF θα αντικατασταθεί από έναν βόρειο πόλο ο οποίος θα δημιουργήσει απωστική δύναμη με το βόρειο πόλο του ρότορα με αποτέλεσμα ο ρότορας να μη μπορεί να ξεκινήσει και κατά συνέπεια ούτε και ο ίδιος ο κινητήρας.



Για να μπορέσουμε να εξασφαλίσουμε την αυτοεκκίνηση του ηλεκτρικού κινητήρα, ένα κυλινδρικό «σκίουρου-κλουβί» (damper winding), εφαρμόζεται μέσω των άκρων των πόλων του ρότορα.

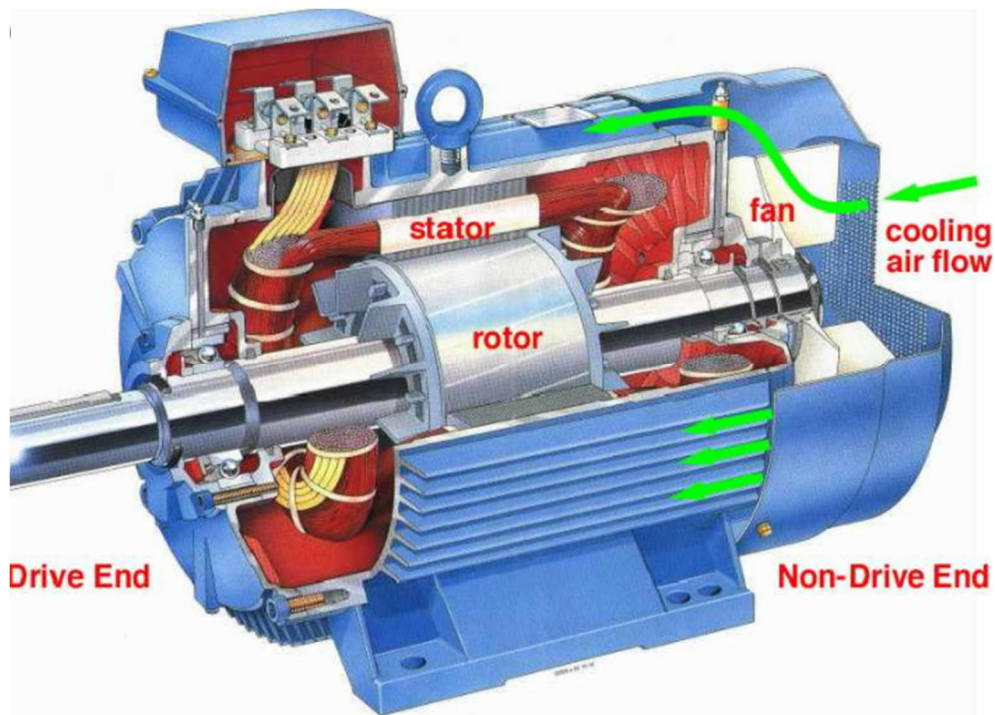


Αρχικά, τα πηνία του ρότορα δεν διεγείρονται από ηλεκτρικό ρεύμα. Ωστόσο, το περιστρεφόμενο μαγνητικό πεδίο δημιουργεί ηλεκτρικό ρεύμα στις ράβδους του κλουβιού-σκίουρου και ο ρότορας αρχίζει να περιστρέφεται όπως ένας επαγωγικός κινητήρας.



Όταν ο ρότορας έχει επιτύχει τη μέγιστη ταχύτητά του, τα πηνία του διεγείρονται από ηλεκτρικό ρεύμα και στη συνέχεια οι πόλοι του ρότορα κλειδώνονται με πόλους RMF με αποτέλεσμα να αρχίσει να περιστρέφεται με σύγχρονη ταχύτητα όπως περιγράφηκε στην παραπάνω διαδικασία. Όταν ο ρότορας περιστρέφεται με σύγχρονη ταχύτητα η σχετική κίνηση μεταξύ κλουβιού σκίουρου και RMF είναι μηδέν. Αυτό συνεπάγεται μηδενικό ρεύμα και δύναμη στις ράβδους κλουβιού σκίουρου, με αποτέλεσμα να μην επηρεάζεται τη συγχρονισμένη λειτουργία του κινητήρα.





Οι ΣΜΜΜ παράγουν σταθερή ταχύτητα ανεξάρτητα από το φορτίο του κινητήρα, μόνο εάν το φορτίο είναι εντός της ικανότητας του κινητήρα. Εάν το εξωτερικό φορτίο ροπής υπερβαίνει τη ροπή που παράγεται από τον κινητήρα, ο κινητήρας θα χάσει το συγχρονισμό του. Αντίστοιχα αποτελέσματα μπορεί να έχουμε και σε περιπτώσεις χαμηλής τάσης τροφοδοσίας και τάσης διέγερσης.

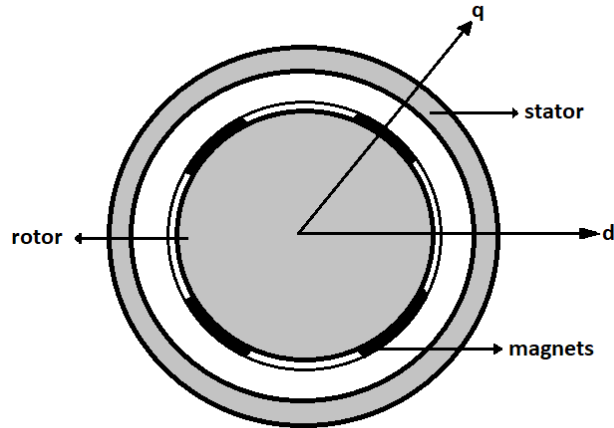
### 1.3 | Περιγραφή Μεταβλητών του Συνόλου Δεδομένων

Το σύνολο των δεδομένων που χρησιμοποιήθηκε στην παρούσα διπλωματική περιείχε τις παρακάτω μεταβλητές:

- Ambient Temperature: θερμοκρασία περιβάλλοντος. (Όπως αυτή μετρήθηκε από αισθητήρα θερμοκρασίας που τοποθετήθηκε κοντά στο στάτη.)
- Coolant: Θερμοκρασία ψυκτικού υγρού (Ο συγκεκριμένος ηλεκτρικός κινητήρας είναι υδρόψυκτος. Η μέτρηση της θερμοκρασίας γίνεται κατά την έξοδο του ψυκτικού υγρού από τον κινητήρα.)
- $U_d$ : d-συνιστώσα της τάσης  $V$
- $U_q$ : q-συνιστώσα της τάσης  $V$
- $I_d$ : d-συνιστώσα του ρεύματος  $I$

- $I_q$ : q-συνιστώσα του ρεύματος  $I$

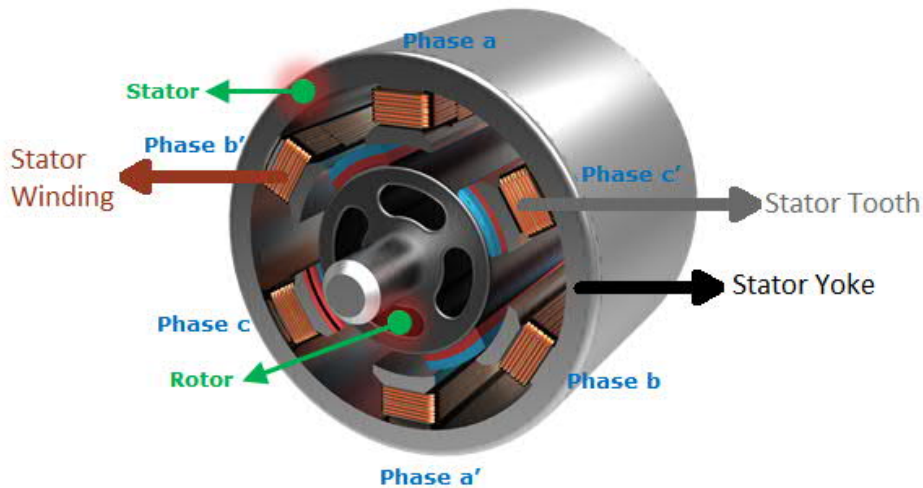
Οι άξονες d,q της τάσης και του ρεύματος φαίνονται στο παρακάτω γράφημα:



- Motor speed: Η ταχύτητα του ηλεκτρικού κινητήρα.
- Torque: Η στρεπτική ροπή που προκαλείται από το ηλεκτρικό ρεύμα.
- Pm: Θερμοκρασία του ρότορα (Όπως αυτή μετρήθηκε στην επιφάνεια του μόνιμου μαγνήτη από infrared μονάδα)
- Stator\_yoke: Θερμοκρασία του σιδήρου στάτη, το ελασματοποιημένο τμήμα του στάτη, χωρίς το μέρος των δοντιών που σχηματίζουν τις αυλακώσεις. (Όπως αυτή μετρήθηκε από αισθητήρα θερμοκρασίας)
- Stator\_tooth: Θερμοκρασία στα δόντια του στάτη (Όπως αυτή μετρήθηκε από αισθητήρα θερμοκρασίας)
- Stator\_winding: Θερμοκρασία στο τύλιγμα του στάτη, ένα σύνολο κυλινδρικών τμημάτων που αποτελούνται από συμπαγείς χάλκινους κυλινδρικούς αγωγούς και το μονωτικό υλικό μεταξύ αυτών και της αυλάκωσης (Όπως αυτή μετρήθηκε από αισθητήρα θερμοκρασίας)

Τα τρία παραπάνω μέρη του στάτορα μιας ΣΜΜΜ μπορούμε να διακρίνουμε στο παρακάτω σχεδιάγραμμα:

Profile\_Id: Το id κάθε μέτρησης που πραγματοποιήθηκε.



## 2 | Μέθοδοι που χρησιμοποιήθηκαν

Στις τρεις πρώτες υπο-ενότητες του κεφαλαίου αυτού, θα ασχοληθούμε με Γραμμική Παλινδρόμηση, Δέντρα Απόφασης και αλγορίθμους Boosting. Οι μέθοδοι αυτοί ανήκουν στην κατηγορία *Εποπτικών Μεθόδων* ή *Supervised Methods* και προσπαθούν να ανακαλύψουν τη σχέση μεταξύ των γνωρισμάτων εισόδου – ανεξάρτητες μεταβλητές – και της μεταβλητής απόκρισης – εξαρτημένη μεταβλητή – με χρήση μοντέλων. Τα μοντέλα περιγράφουν και εξηγούν φαινόμενα, τα οποία είναι ‘κρυμμένα’ στο σύνολο δεδομένων και μπορούν να προβλέψουν την τιμή της μεταβλητής ‘στόχου’, όταν είναι γνωστές οι τιμές των ανεξάρτητων μεταβλητών. Υπάρχουν δύο κύριες κατηγορίες εποπτικών μοντέλων ή supervised models, τα μοντέλα κατάταξης ( Ταξινομητές ή Classifiers ) και τα μοντέλα παλινδρόμησης ( Regression models ). Τα μοντέλα παλινδρόμησης αντιστοιχούν τον χώρο εισόδου σε ένα πραγματικό τομέα, σε αντίθεση με τους ταξινομητές οι οποίοι χαρτογραφούν τον χώρο εισόδου σε προκαθορισμένες κατηγορίες.

### 2.1 | Γραμμική Παλινδρόμηση

Ένα πολύ σημαντικό πρόβλημα της επιστημονικής κοινότητας είναι η ακριβής πρόβλεψη μίας μεταβλητής. Το πρόβλημα αυτό αφορά όλους τους τομείς της επιστήμης, όχι μόνο από ερευνητική άποψη αλλά και από την πλευρά των επιχειρήσεων, καθώς το να μπορεί να προβλέψει κάποιος την τιμή μίας μεταβλητής μπορεί να επιφέρει τεράστιες απολαβές. Από το χρηματιστήριο – πρόβλεψη μετοχής – το εμπόριο – πρόβλεψη επιτυχίας ενός προϊόντος – τη



μηχανική – πρόβλεψη αντοχής υλικού – την ιατρική – πρόβλεψη ασθένειας, και σε όλους τους άλλους τομείς που μπορεί ο καθένας να σκεφτεί, εφαρμόζεται και είναι απαραίτητη η πρόβλεψη μίας τιμής.

Η Ανάλυση Παλινδρόμησης είναι ένας όρος που περιγράφει μία σειρά αναλύσεων που αφορούν μερικούς από τους τρόπους που έχουν βρει οι επιστήμονες για να λύσουν το συγκεκριμένο πρόβλημα. Η Ανάλυση Παλινδρόμησης βοηθά να εξετάσουμε σχέσεις αιτιότητας μεταξύ των μεταβλητών, δηλαδή να ερμηνεύσουμε την τιμή της εξαρτημένης μεταβλητής – η μεταβλητή ‘στόχος’ που μας ενδιαφέρει να προβλέψουμε – σύμφωνα με μία ή περισσότερες ανεξάρτητες μεταβλητές.

Ένα μοντέλο παλινδρόμησης είναι μία συνάρτηση συσχέτισης της εξαρτημένης μεταβλητής από τις ανεξάρτητες. Συσχετίζει τη μεταβλητή πρόβλεψης  $Y$  σε μία συνάρτηση παλινδρόμησης ( regression ) των  $X$  και  $\beta$ , όπου  $\mathbf{X}$  είναι το διάνυσμα των ανεξάρτητων μεταβλητών σύμφωνα με τις οποίες θα προσπαθήσει να προβλέψει το  $Y$  και  $\beta$ , οι άγνωστοι συντελεστές συσχέτισης που θέλουμε να βρει το μοντέλο. Δηλαδή,

$$Y \cong F(X, \beta)$$

Ένα είδος παλινδρόμησης είναι η γραμμική παλινδρόμηση, που χωρίζεται σε Απλή Γραμμική Παλινδρόμηση και Πολλαπλή Γραμμική Παλινδρόμηση. Η απαίτηση του μοντέλου που θα παραχθεί είναι η εξαρτημένη μεταβλητή  $Y$  να είναι γραμμικός συνδυασμός των ανεξάρτητων μεταβλητών.

Στόχος της Απλής Γραμμικής Παλινδρόμησης είναι να βρεθεί η συνάρτηση που δημιουργεί μία ευθεία γραμμή που θα συσχετίζει καλύτερα την εξαρτημένη μεταβλητή, με τις μεταβλητές του συνόλου  $\mathbf{X}$ . Η ευθεία γραμμή θα έχει τη μορφή :

$$f(x) = Y = a + \beta X$$

Η Πολλαπλή Γραμμική Παλινδρόμηση είναι επέκταση της Απλής Γραμμικής Παλινδρόμησης για δύο ή περισσότερες ανεξάρτητες μεταβλητές. Δηλαδή, όταν πλέον θεωρούμε ότι η εξαρτημένη μεταβλητή είναι γραμμικά εξαρτημένη, επιπλέον και από μία δεύτερη μεταβλητή ( $X_2$ ) ή και από μία τρίτη ( $X_3$ ) ή από ένα σύνολο  $m$  μεταβλητών  $X$ . Τότε, η ευθεία γραμμή θα έχει τη μορφή :

$$f(x) = Y = a + \sum_{i=1}^m \beta_i X_i$$

### 2.1.1 | Μέθοδος Ελαχίστων Τετραγώνων

Για να βρεθεί η συνάρτηση της ευθείας, χρησιμοποιείται η ‘Μέθοδος Ελαχίστων Τετραγώνων’ η οποία αναζητά την ευθεία από όπου η απόσταση των σημείων ( $X_i, Y_i$ ) είναι η ελάχιστη. Γνωρίζοντας λοιπόν τις πειραματικές τιμές των

μεγεθών, τα σημεία δηλαδή  $(X_i, Y_i)$  προσπαθούμε να βρούμε τον μαθηματικό τύπο που τα συνδέει, ελαχιστοποιώντας το άθροισμα των τετραγώνων των αποκλίσεων μεταξύ των θεωρητικών και των πειραματικών τιμών. Θέλουμε να ελαχιστοποιήσουμε την παράσταση :

$$S(a, \beta_i) = \sum_{i=1}^m (Y - \alpha - \sum_{i=1}^m \beta_i X_i)^2$$

Παραγωγίζοντας την παράσταση ως προς  $a$  και ως προς  $\beta_1$  και θέτοντας τις μερικές παραγώγους ίσες με το μηδέν, καταλήγουμε σε ένα ζεύγος εξισώσεων που μας δίνουν τις εκτιμήσεις των συντελεστών  $a, \beta_1$ , οι οποίες λέγονται εκτιμήτριες ελαχίστων τετραγώνων.

### 2.1.2 | Ερμηνεία Παραμέτρων Πολλαπλής Γραμμικής Παλινδρόμησης

Η παράμετρος  $a$ , εκφράζει την αναμενόμενη τιμή της τ.μ.  $Y$  όταν κάθε συνιστώσα του τυχαίου δείγματος  $X$  είναι μηδέν. Η παράμετρος  $b_m$  εκφράζει το πόσο αναμένεται να μεταβληθεί η τ.μ.  $Y$ , αν η τ.μ.  $X_m$ , μεταβληθεί κατά μια μονάδα και όλες οι άλλες τυχαίες μεταβλητές  $X_i$  με  $m \neq i$ , παραμείνουν σταθερές. Το πρόσημο του  $b_m$ , εκφράζει τη σχέση εξάρτησης μεταξύ των  $Y$  και  $X_m$  όταν όλες οι  $X_i$  παραμείνουν σταθερές.

Οι ποσότητες  $\hat{\varepsilon}_i = y_i - \hat{y}_i$ , με  $\hat{y}_i = \hat{\alpha} + \hat{b}_1 x_1 + \hat{b}_2 x_2 + \dots + \hat{b}_k x_k$ , ( $i=1, \dots, n$ ), ονομάζονται υπόλοιπα και εκφράζουν τις εκτιμήσεις των σφαλμάτων των μετρήσεων, ενώ οι ποσότητες  $\hat{y}_i$  είναι οι προβλεπόμενες τιμές.

### 2.1.3 | Μετρικές Αξιολόγησης Μοντέλου Παλινδρόμησης

Οι τρόποι και οι μέθοδοι για τον έλεγχο της αποτελεσματικότητας ενός μοντέλου παλινδρόμησης ποικίλουν. Θα αναφερθούμε στους πιο χαρακτηριστικούς με κάποιους εκ των οποίων έχει αξιολογηθεί και το μοντέλο παλινδρόμησης της εργασίας.

- Συντελεστής Προσδιορισμού ( $R^2$ )

Ως συντελεστή προσδιορισμού ( coefficient of determination ) ορίζεται η ποσότητα :  $R^2 = 1 - \frac{\sum_{i=1}^n \{y_i - \hat{y}_i\}^2}{\sum_{i=1}^n \{y_i - \bar{y}\}^2}$ , η οποία παίρνει τιμές στο διάστημα  $[0,1]$

και εκφράζει το ποσοστό της διασποράς της τ.μ.  $Y$  που εκφράζεται μέσω ενός μοντέλου παλινδρόμησης. Αναφέρεται στο βαθμό στον οποίο οι ανεξάρτητες μεταβλητές  $X$  ερμηνεύουν το βαθμό μεταβλητότητας της εξαρτημένης μεταβλητής  $Y$ , μέσω της γραμμικής παλινδρόμησης. Όσο μεγαλύτερη είναι η τιμή του συντελεστή τόσο ισχυρότερη είναι η γραμμική σχέση εξάρτησης των  $Y$  και  $X$ , αν και μόνο αν το πολλαπλό γραμμικό μοντέλο πληροί τις απαραίτητες προϋποθέσεις και ορίζεται ως κατάλληλο.

Ωστόσο ο συντελεστής προσδιορισμού επηρεάζεται από το πλήθος των μεταβλητών, είναι καλό να μην χρησιμοποιείται ως μέτρο καλής προσαρμογής του μοντέλου στα δεδομένα ή ως μέτρο σύγκρισης δύο μοντέλων, διότι είναι ευαίσθητη στον αριθμό επεξηγηματικών μεταβλητών ενός μοντέλου. Για παράδειγμα, αν σε ένα γραμμικό μοντέλο προσθέσουμε μία επεξηγηματική μεταβλητή με ελάχιστη συνεισφορά στη μείωση της αβεβαιότητας μας για την τιμή της μεταβλητής απόκρισης, ο πολλαπλός συντελεστής προσδιορισμού θα αυξηθεί, δίνοντάς μας έτσι την εσφαλμένη εντύπωση ότι το νέο μοντέλο είναι καλύτερο, λόγω της νέας μεταβλητής απόκρισης που προστέθηκε, ανεξαρτήτως από το αν καλυτερεύει το μοντέλο ή όχι. Σε τέτοιες περιπτώσεις είναι προτιμότερο να υπολογίζουμε τον προσαρμοσμένο συντελεστή προσδιορισμού ( adjusted coefficient of determination ) ο οποίος εκφράζει το ποσοστό της διασποράς της τυχαίας μεταβλητής  $Y$  που εξηγείται με βάση το μοντέλο παλινδρόμησης λαμβάνοντας όμως υπόψη και την πολυπλοκότητα ενός μοντέλου, δηλαδή το πλήθος των επεξηγηματικών μεταβλητών. Εάν σε ένα μοντέλο προσθέσουμε μία επεξηγηματική μεταβλητή που δεν βελτιώνει το μοντέλο η τιμή του προσαρμοσμένου συντελεστή δεν θα είναι μεγαλύτερη από αυτή του αρχικού μοντέλου. Οι δύο συντελεστές προσδιορισμού συνδέονται μέσω της σχέσης:

$$\tilde{R}^2 = R^2 - \frac{(1 - R^2) p}{n - p - 1},$$

όπου  $p$  ο αριθμός των επεξηγηματικών μεταβλητών και  $n$  το πλήθος των παρατηρήσεων του δείγματος.

- Μέσο Τετραγωνικό Σφάλμα (Mean Squared Error)

Το μέσο τετραγωνικό σφάλμα(MSE) είναι το πιο γνωστό και ευρέως χρησιμοποιούμενο αλλά και ταυτόχρονα ένα από τα πιο χρήσιμα και αντικειμενικά κριτήρια καλής προσαρμογής για ένα μοντέλο παλινδρόμησης. Το μέσο τετραγωνικό σφάλμα είναι αμερόληπτος εκτιμητής της διασποράς του τυχαίου σφάλματος  $\sigma^2$  και δίνεται από τη σχέση:

$$MSE = \frac{1}{n} \sum_{i=1}^n \{y_i - \hat{y}_i\}^2.$$

Η θετική τετραγωνική ρίζα του MSE, καλείται τυπικό σφάλμα της παλινδρόμησης (standard error of the regression), και όσο μικρότερη η τιμή της τόσο καλύτερη προσαρμογή έχουμε στα δεδομένα μας. Συνήθως χρησιμοποιείτε το τυπικό σφάλμα, αυτό δικαιολογείται επειδή το μέσο τετραγωνικό σφάλμα αξιολογεί τις τετραγωνικές αποκλίσεις των τιμών ενός μοντέλου, ενώ το τυπικό σφάλμα είναι άμεσα συγκρίσιμο με τις μονάδες μέτρησης του μοντέλου. Η χρησιμότητα του μέσου τετραγωνικού σφάλματος ως εκτιμητή οφείλεται κυρίως στο ότι ο υπολογισμός του απαιτεί μόνο την εύρεση δύο βασικών παραμέτρων, της μέσης τιμής και της διασποράς του μοντέλου και όχι κατ' ανάγκη γνώση της κατανομής του.

- Μέσο Απόλυτο Σφάλμα (Mean Absolute Error)

Το μέσο απόλυτο σφάλμα, εκφράζει το μέσο μέγεθος των σφαλμάτων σε ένα σύνολο προβλέψεων, ενός μοντέλου παλινδρόμησης, αγνοώντας την κατεύθυνση τους (δηλαδή το αν τα  $\hat{y}$  είναι μεγαλύτερα ή μικρότερα από τα  $y$ ). Πρόκειται για το μέσο όρο των απόλυτων διαφορών μεταξύ των προβλέψεων του  $y$  και των πραγματικών τιμών του, και δίνεται από τη σχέση:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$$

Το μέσο απόλυτο σφάλμα είναι πιο ανθεκτικό απέναντι σε ακραίες προβλέψεις (outliers), σε αντίθεση με το μέσο τετραγωνικό σφάλμα, που λόγω του τετραγώνου στον τύπο τους, δίνει μεγάλο βάρος στις ακραίες τιμές στις προβλέψεις  $\hat{y}$ .

- Ποσοστιαίο – Σχετικό Σφάλμα (Relative Error)

Το ποσοστιαίο – σχετικό σφάλμα της παλινδρόμησης, εκφράζει τη διαφορά των προβλέψεων του μοντέλου παλινδρόμησης και των πραγματικών τιμών που αντιστοιχούν σε αυτές, προς τις πραγματικές τιμές και ορίζεται από τον τύπο:

$$\text{Relative Error} = \frac{y_{i \text{ predicted}} - y_{i \text{ true}}}{y_{i \text{ true}}}$$

Μπορεί να μην αποτελεί μετρική για την αξιολόγηση ενός μοντέλου παλινδρόμησης όπως οι προηγούμενες μετρικές, αλλά η μέση τιμή του και η διασπορά του σχετικού σφάλματος αποτελούν δύο καλές μετρικές για την ακρίβεια των προβλέψεων ενός μοντέλου πρόβλεψης. Εάν πετύχουμε η μέση τιμή σε όλο το εύρος του συνόλου τιμών να πλησιάζουν το μηδέν και η διασπορά να τείνει σε αυτό όσο, αυξάνεται η τιμή της μεταβλητής απόκρισης  $y$ , έχουμε κατασκευάσει ένα ικανοποιητικό μοντέλο πρόβλεψης, με ικανοποιητική ακρίβεια σε όλο το πεδίο ορισμού της  $y$ .

## 2.1.4 | Προϋποθέσεις Πολλαπλού Γραμμικού Μοντέλου

Αναφερθήκαμε πριν στις μετρικές αξιολόγησης γραμμικού μοντέλου, οι οποίες έχουν νόημα αν και μόνο αν το μοντέλο μας πληροί κάποιες προϋποθέσεις. Οι προϋποθέσεις αυτές αφορούν αρχικά το αν η σχέση μεταξύ της δεσμευμένης μέσης τιμής της μεταβλητής απόκρισης και των επεξηγηματικών μεταβλητών μπορεί να περιγραφεί από ένα γραμμικό μοντέλο, 'Υπόθεση Γραμμικότητας'. Έπειτα, πριν αναφερθούμε σε στατιστική συμπερασματολογία, οφείλουμε να προβούμε σε έλεγχο και των υπόλοιπων προϋποθέσεων, δηλαδή την 'Κανονικότητα των Σφαλμάτων', την 'Ομοσκεδαστικότητα' και την 'Ανεξαρτησία των Σφαλμάτων'. Η κάθε προϋπόθεση ελέγχεται με έναν συγκεκριμένο τρόπο και σε αυτούς θα αναφερθούμε στη συνέχεια.

**Γραμμικότητα:** Για τον έλεγχο αυτής της προϋπόθεσης υπάρχουν δύο τρόποι, ο ένας αφορά την περίπτωση που οι επεξηγηματικές μεταβλητές είναι μεταξύ τους ασυσχέτιστες και ο δεύτερος όταν οι επεξηγηματικές μεταβλητές σχετίζονται μεταξύ τους. Η πρώτη περίπτωση είναι και η πιο απλή καθώς τότε αρκεί για κάθε επεξηγηματική μεταβλητή να ελέγξουμε αν υπάρχει γραμμική σχέση με τη μεταβλητή απόκρισης, αυτό γίνεται γραφικά σχεδιάζοντας το διάγραμμα διασποράς των σημείων  $(x_i, y_i)$  - όπου  $i = 1, \dots, n$  - βλέποντας αν τα σημεία στο διάγραμμα είναι κοντά στην ευθεία  $y = x$ . Στην δεύτερη περίπτωση, πρέπει να ελέγξουμε αν η τιμή της επεξηγηματικής μεταβλητής  $X_j, j=1, \dots, p$ , συνδέεται γραμμικά με τη δεσμευμένη μέση τιμή της  $Y$ , όταν οι τιμές των υπολοίπων επεξηγηματικών μεταβλητών, συνδέονται γραμμικά με τη δεσμευμένη μέση τιμή της  $Y$ . Για τον έλεγχο αυτό, χρειαζόμαστε τον ορισμό των μερικών υπολοίπων  $P_{ij}$ . Από τη σχέση  $y \approx \hat{a} + \hat{b}_1 x_{i1} + \dots + p_j(x_{ij}) + \dots + \hat{b}_p x_{ip}$ ,  $i=1, \dots, n$ , όπου αποδεικνύεται εύκολα ότι,  $p_j(x_{ij}) \approx \hat{b}_j x_{ij} + \hat{e}_i \equiv P_{ij}$ , ορίζονται οι όροι  $P_{ij}$ , τα  $j$ -μερικά υπόλοιπα (partial residuals) τα οποία χρησιμοποιούμε για να ελέγξουμε την προϋπόθεση της γραμμικότητας όταν έχουμε συσχετισμένες μεταξύ τους μεταβλητές, σχεδιάζοντας τα διαγράμματα διασποράς των σημείων  $(x_{ij}, P_{ij})$ , για κάθε μεταβλητή.

**Κανονικότητα των σφαλμάτων:** Για τον έλεγχο αυτής της προϋπόθεσης, θεωρούμε τα υπόλοιπα  $\hat{e}_i = y_i - \hat{a} - \hat{b}_1 x_{i1} - \dots - \hat{b}_p x_{ip}$ , δηλαδή τις εκτιμήσεις των τυπικών σφαλμάτων. Στη συνέχεια, είτε μέσα από το ιστόγραμμα τους είτε από κανονικό διάγραμμα (normal Q-Q plot), ελέγχουμε την κανονικότητα των σφαλμάτων, βλέποντας αν τα σημεία βρίσκονται περίπου σε μια ευθεία γραμμή.

**Ομοσκεδαστικότητα:** Ο όρος ομοσκεδαστικότητα, αφορά τον έλεγχο της διασποράς των τυχαίων σφαλμάτων  $e_i$  για το αν παραμένει σταθερή για τις διάφορες τιμές  $x$  του τυχαίου διανύσματος. Χρησιμοποιείται το διάγραμμα διασποράς, μεταξύ των υπολοίπων  $\hat{e}_i$ , και των προβλεπόμενων τιμών  $\hat{y}_i$ . Για να ικανοποιείται η προϋπόθεση θα πρέπει στο διάγραμμα τα σημεία να μην εμφανίζουν κάποιο συστηματικό τρόπο συμπεριφοράς. Διαφορετικά, αποτελεί

ένδειξη ότι δεν ικανοποιείται η ομοσκεδαστικότητα, και πρέπει να μετασχηματίσουμε τη μεταβλητή απόκρισης.

Ανεξαρτησία των σφαλμάτων: Τα τυχαία σφάλματα θα πρέπει να είναι ανεξάρτητα μεταξύ τους, σε αντίθετη περίπτωση λέμε ότι παρουσιάζεται το πρόβλημα της αυτοσυσχέτισης.

### 2.1.5 | **Συσχέτιση Μεταβλητών Γραμμικού Μοντέλου**

Για την κατασκευή ενός σωστού μοντέλου παλινδρόμησης απαιτείται η κατάλληλη επιλογή των επεξηγηματικών μεταβλητών. Θέλοντας να προβλέψουμε ή/υε μία μεταβλητή απόκρισης με βάση κάποιες επεξηγηματικές μεταβλητές, η επιλογή τους είναι κρίσιμη και ένα στοιχείο που θα πρέπει να προσέξουμε είναι η μεταξύ τους συσχέτιση. Η κάθε επεξηγηματική μεταβλητή έχει καθοριστικό ρόλο για το μοντέλο και δίνει πληροφορία που θα οδηγήσει εν τέλει στην μέγιστη απόδοση του μοντέλου. Συνεπώς, θέλουμε κάθε μεταβλητή να δίνει ουσιαστική πληροφορία που να μην είναι παρόμοια με μία άλλη που θα χρησιμοποιήσουμε στο μοντέλο, άρα θέλουμε όσο το δυνατόν μικρότερη συσχέτιση γίνεται. Ταυτόχρονα οι μεταβλητές θέλουμε να είναι συσχετισμένες με την μεταβλητή απόκρισης και οι πληροφορίες που δίνουν να είναι χρήσιμες για την πρόβλεψή της, χωρίς όμως να είναι όμοιες ή να έχουν κάποια συστηματική σχέση, όπως για παράδειγμα να είναι γραμμικά συσχετισμένες καθώς αυτό μπερδεύει το μοντέλο και οδηγεί σε λάθος συμπεράσματα. Όλα αυτά αποτελούν ιδανικές συνθήκες τις οποίες προσπαθούμε να φτάσουμε με τη σωστή επεξεργασία και ανάλυση του συνόλου δεδομένων που διαθέτουμε.

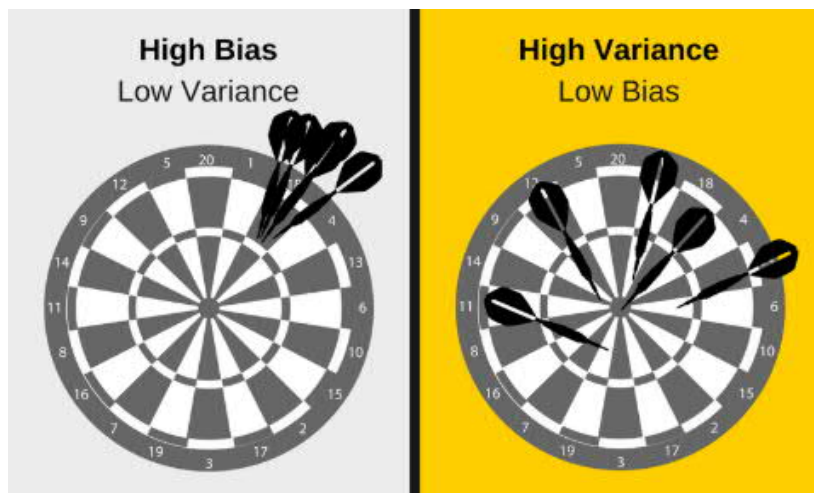
## 2.2 | **Μεθοδολογίες Μηχανικής Μάθησης**

### 2.2.1 | **Υπερπροσαρμογή Δεδομένων (Overfitting)**

Πριν αναλύσουμε μεθόδους μηχανικής μάθησης πρέπει να αναφερθούμε στην έννοια της υπερπροσαρμογής ή υπερεκπαίδευσης δεδομένων, η οποία προκαλείται όταν ένα μοντέλο πρόβλεψης είναι απόλυτα εξαρτημένο από τα δεδομένα με τα οποία εκπαιδεύτηκε και δεν διαχειρίζεται με την ίδια απόδοση ένα οποιοδήποτε άλλο σύνολο δεδομένων. Για την καλύτερη κατανόηση της έννοιας θα αναφερθούμε στο αντίθετο πρόβλημα, την υποπροσαρμογή του μοντέλου, η οποία προκαλείται όταν ένα μοντέλο είναι πολύ απλό, αποτελείται

δηλαδή από πολύ λίγες επεξηγηματικές μεταβλητές, κάτι που καθιστά το μοντέλο άκαμπτο και ανίκανο στο να μάθει και να εκπαιδευτεί μέσα από το σύνολο εκπαίδευσης. Απλά μοντέλα εκπαίδευσης τείνουν να έχουν λιγότερο διακύμανση στις προβλέψεις τους και περισσότερη μεροληψία απέναντι σε λάθος προβλέψεις. Από την άλλη, περίπλοκοι εκπαιδευτές τείνουν να έχουν περισσότερη διακύμανση στις προβλέψεις τους. Σε αλγορίθμους εκμάθησης τόσο η διακύμανση, όσο και η μεροληψία αποτελούν μορφές σφάλματος πρόβλεψης και η εύρεση ισορροπίας μεταξύ ενός απλού αλγορίθμου, με μεγάλη μεροληψία, και ενός πολύπλοκου, με μεγάλη διασπορά, αποτελεί αντικείμενο ενδιαφέροντος για τους τομείς της στατιστικής και της μηχανικής μάθησης.

### Ανίχνευση Υπερπροσαρμογής Δεδομένων



Μία πρόκληση με την υπερεκπαίδευση είναι ότι δεν γνωρίζουμε πόσο καλά το μοντέλο πρόβλεψης θα λειτουργήσει μέχρι να το δοκιμάσουμε σε ένα νέο σύνολο δεδομένων. Για αυτό το λόγο ο πιο απλός και γρήγορος τρόπος για να το ελέγξουμε είναι να χωρίσουμε το σύνολο δεδομένων που διαθέτουμε σε δύο υποσύνολα δεδομένων, το υποσύνολο εκπαίδευσης ( training set ) και το υποσύνολο επαλήθευσης ( testing set ). Έτσι, εφαρμόζοντας το μοντέλο που έχει εκπαιδευτεί με το σύνολο εκπαίδευσης, στο σύνολο επαλήθευσης και συγκρίνοντας τα αποτελέσματα, μπορούμε να καταλάβουμε αν υπάρχει υπερπροσαρμογή ή όχι. Αν για παράδειγμα στο σύνολο εκπαίδευσης έχουμε 99% ακρίβεια στις προβλέψεις μας, ενώ στο σύνολο επαλήθευσης έχουμε 55% ακρίβεια, στην ίδια μετρική, τότε είναι ξεκάθαρο σημάδι υπερπροσαρμογής του μοντέλου στα δεδομένα εκπαίδευσης. Επίσης, εάν δύο μοντέλα με διαφορετική αρχιτεκτονική μας δίνουν ίδια αποτελέσματα, καλό είναι να προτιμούμε το πιο απλό μοντέλο και να μην αυξάνουμε την πολυπλοκότητα του παρά μόνο εάν βελτιώνεται αισθητά η ακρίβεια της πρόβλεψης.

## Πρόληψη Υπερπροσαρμογής Δεδομένων

Το να είμαστε σε θέση να ανιχνεύσουμε την υπερπροσαρμογή κατά τη διαδικασία κατασκευής ενός μοντέλου πρόβλεψης είναι απαραίτητο, αλλά το επόμενο βήμα είναι η λύση του προβλήματος. Στη συνέχεια θα μιλήσουμε για κάποιους από τους πιο γνωστούς και χρήσιμους τρόπους αντιμετώπισης αυτού του φαινομένου.

- Εκπαίδευση με περισσότερα δεδομένα

Είναι η πιο απλή μορφή αντιμετώπισης της υπερπροσαρμογής των δεδομένων και δεν φέρει αποτέλεσμα σε κάθε περίπτωση, παρόλα αυτά η εκπαίδευση ενός μοντέλου με περισσότερα δεδομένα μπορεί να βοηθήσει τον αλγόριθμο να ανιχνεύσει καλύτερα την πληροφορία. Ένα σημαντικό στοιχείο στην μέθοδο αυτή είναι ότι θα πρέπει η προσθήκη περισσότερων δεδομένων να είναι όχι μόνο ποσοτική, αλλά και ποιοτική, διότι η προσθήκη 'θορύβου' ( noise ) δεν θα βοηθήσει στην καλύτερη εκπαίδευση του μοντέλου. Ως θορυβώδη δεδομένα, ορίζουμε εκείνα τα οποία είναι ακραίες ή εσφαλμένες τιμές που δεν βοηθούν στην εκπαίδευση του μοντέλου αλλά αντιθέτως το μπερδεύουν και επηρεάζουν την απόδοσή του.

- Αφαίρεση Μεταβλητών ( Feature Removal )

Σε ένα μοντέλο πρόβλεψης όσες περισσότερες επεξηγηματικές μεταβλητές έχει τόσο πιο περίπλοκο είναι και υπάρχει μεγαλύτερη πιθανότητα να εμφανιστεί το φαινόμενο της υπερπροσαρμογής. Έτσι, είναι πολύ σημαντικό κατά την κατασκευή του να έχουμε πλήρη γνώση για το τι θέλουμε να πετύχουμε, ποια είναι η μεταβλητή που θέλουμε να προβλέψουμε και ποιες μεταβλητές την περιγράφουν καλύτερα. Για όλες τις μεταβλητές που χρησιμοποιούμε θα πρέπει να είμαστε σίγουροι ότι προσφέρουν αξία στο μοντέλο μας και βελτιώνουν την πρόβλεψη, αν δεν είμαστε σίγουροι για αυτό ή δεν καταλαβαίνουμε τι ακριβώς πληροφορία προσφέρει μία μεταβλητή, είναι καλύτερα να την αφαιρούμε από τις επεξηγηματικές μεταβλητές από το να χρησιμοποιούμε κάτι που δεν καταλαβαίνουμε τι προσφέρει στο μοντέλο. Επίσης, υπάρχουν αυτόματες μέθοδοι που μας δείχνουν σε τι ποσοστό χρησιμοποιείται η κάθε μεταβλητή προκειμένου το μοντέλο να καταλήξει στη πρόβλεψη, έτσι, οι μεταβλητές που προσφέρουν λίγη ή και καθόλου πληροφορία στο μοντέλο και δεν τις αξιοποιούν θα πρέπει να αφαιρούνται από το σύνολο των επεξηγηματικών μεταβλητών. Δεν θα πρέπει να ξεχνάμε ότι ένα μοντέλο κατασκευάζεται με σκοπό να χρησιμοποιηθεί σε αληθινά δεδομένα που είτε τα έχουν ήδη συλλέξει, είτε θα συλλεχθούν ξανά στο μέλλον, οπότε όσο πιο λίγες μεταβλητές χρειάζεται το μοντέλο για να προβλέψει τη ζητούμενη τιμή τόσο πιο εύκολη θα είναι και η συλλογή των απαιτούμενων δεδομένων στο μέλλον.

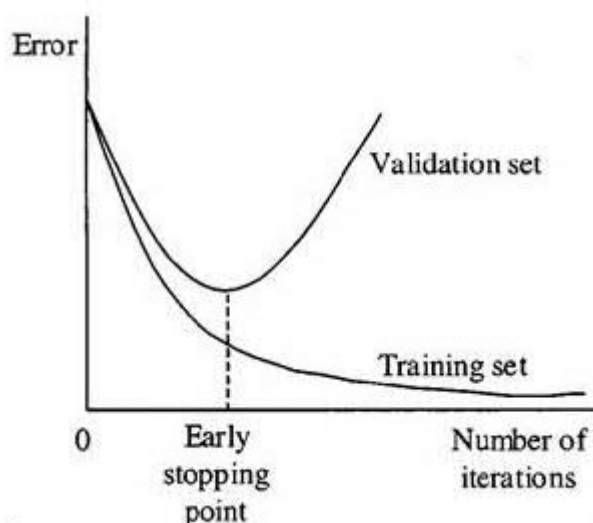


- Διασταυρωμένη Επικύρωση ( Cross - Validation )

Η μέθοδος Cross-Validation είναι ένα ισχυρό προληπτικό μέτρο απέναντι στην υπερπροσαρμογή. Θα χαρακτηριζόταν κανείς την ιδέα πίσω από τη μέθοδο έξυπνη, καθώς χρησιμοποιούμε τα δεδομένα από το υποσύνολο εκπαίδευσης για να δημιουργήσουμε πολλαπλά μικρά υποσύνολα και να 'παίξουμε' με το μοντέλο μας. Στην k-fold cross-validation χωρίζουμε τα δεδομένα σε k υποσύνολα, που λέγονται πτυχές ( folds ). Έπειτα, εκπαιδεύουμε επαναληπτικά τον αλγόριθμο χρησιμοποιώντας k-1 πτυχές ( υποσύνολα ) και έχοντας το υπολειπόμενο υποσύνολο για επαλήθευση ( holdout fold ) και όλα τα υποσύνολα θα έχουν χρησιμοποιηθεί ως υποσύνολο επαλήθευσης και εκπαίδευσης. Σε κάθε μια από τις k επαναλήψεις υπολογίζουμε την απόδοση του μοντέλου και στο τέλος παίρνουμε το μέσο όρο αυτής, έτσι πετυχαίνουμε μεγαλύτερη ακρίβεια και αποτελεσματικότητα στο μοντέλο καθώς η απόδοση του δεν εξαρτάται πλέον από την επιλογή των δεδομένων, καθώς ξαναχρησιμοποιούμε τα δεδομένα εκπαίδευσης, χωρίζοντας τα μέρη και κάνοντας κύκλους δοκιμών σε αυτά.

- Διακοπή Αλγορίθμου ( Early Stopping )

Σε μία επαναληπτική εκπαίδευση αλγορίθμου μπορούμε σε κάθε επανάληψη να μετράμε πόσο καλά προβλέπει το μοντέλο, γνωρίζοντας λοιπόν ότι το μοντέλο βελτιώνεται μέχρι ένα συγκεκριμένο αριθμό επαναλήψεων και ότι από ένα σημείο και μετά, εξασθενείται η ικανότητά του να προβλέπει ένα γενικό σύνολο δεδομένων και αρχίζει να υπερπροσαρμόζεται στα δεδομένα εκπαίδευσης. Αυτή η επανάληψη όπου αρχίζει η υπερπροσαρμογή λέγεται όριο υπερπροσαρμογής και στόχος είναι να σταματήσουμε τη διαδικασία εκπαίδευσης, πριν ο αλγόριθμος περάσει αυτό το σημείο.



Ένας απλός, αποτελεσματικός και σίγουρος τρόπος για την εύρεση του σημείου υπερπροσαρμογής και της αποφυγής του, είναι καθώς μεταβάλλουμε τις

παραμέτρους του αλγορίθμου που έχουμε επιλέξει για την εκπαίδευση του μοντέλου μας και έχοντας στο μυαλό μας μία ή και περισσότερες μετρικές, σύμφωνα με τις οποίες αξιολογούμε την ακρίβεια του μοντέλου, να τις αναπαραστήσουμε γραφικά και αναλόγως με το πώς μεταβάλλεται το σφάλμα του υποσυνόλου εκπαίδευσης και επαλήθευσης, να βρούμε το αντίστοιχο όριο υπερπροσαρμογής. Κοιτώντας το παραπάνω σχήμα, καταλαβαίνουμε που βρίσκεται το όριο υπερπροσαρμογής, παρατηρώντας ότι από εκείνο το σημείο και μετά το σφάλμα του συνόλου εκπαίδευσης αρχίζει και μειώνεται εκθετικά, ενώ αντίθετα το σφάλμα του συνόλου επαλήθευσης αρχίζει και αυξάνεται παραβολικά. Συνεπώς, το μοντέλο έχει εκπαιδευτεί ήδη σε σημείο που να είναι αποτελεσματικό για δεδομένα παρόμοια με αυτά που εκπαιδεύεται, αλλά συνεχίζοντας τις επαναλήψεις εκπαίδευσης θα γινόταν μεροληπτικό απέναντι στο σύνολο δεδομένων εκπαίδευσης. Είναι πολύ σημαντικό κατά την κατασκευή ενός μοντέλου πρόβλεψης να επιτυγχάνουμε παρόμοια απόδοση στα δεδομένα με τα οποία εκπαιδεύτηκε, αλλά και σε παρόμοια νέα δεδομένα που το μοντέλο συναντά για πρώτη φορά. Αυτό δεν είναι εύκολο, και πολλές φορές γίνεται το λάθος να πετυχαίνεται υψηλή ακρίβεια σε πρόβλεψη, από μοντέλα που δεν είναι γενικευμένα και στην εισαγωγή νέων δεδομένων η ακρίβεια πέφτει κατακόρυφα, καθώς το μοντέλο είναι μεροληπτικό και απόλυτα εξαρτημένο από τα δεδομένα στα οποία εκπαιδεύτηκε λόγω του υψηλού αριθμού επαναλήψεων που χρησιμοποιήθηκαν.

### 2.2.2 | **K-Κοντινότεροι Γείτονες**

Ο αλγόριθμος των κ-κοντινότερων γειτόνων κατέχει τη θέση του απλούστερου αλγορίθμου μηχανικής μάθησης και μπορεί να χρησιμοποιηθεί σε προβλήματα κατηγοριοποίησης και παλινδρόμησης. Προβλέπει μία τιμή σύμφωνα με το πόσο κοντά μπορεί να αντιστοιχηθεί σε μία από τις τιμές του συνόλου εκπαίδευσης.

Ας δούμε πως λειτουργεί ο μη παραμετρικός αυτός αλγόριθμος.

- Αρχικά, υπολογίζει για το σημείο που θέλουμε να προβλέψουμε, τις αποστάσεις μεταξύ κάθε σημείου του συνόλου εκπαίδευσης, σύμφωνα με μία συνάρτηση απόστασης.
- Επιλέγονται τα κ-κοντινότερα σημεία και είτε υπολογίζεται ο μέσος όρος τους και θέτεται ως πρόβλεψη, είτε υπολογίζεται ένας αντίστροφα βεβαρυμμένος μέσος όρος, σύμφωνα με την απόστασή τους και θέτεται ως πρόβλεψη.

Όσον αφορά τον τρόπο υπολογισμού της απόστασης μεταξύ των σημείων, η προκαθορισμένη για τον βιβλιοθήκη που χρησιμοποιούμε ( `sklearn.neighbors.KNeighborsRegressor` ) είναι η Ευκλείδεια απόσταση που δίνεται από τον τύπο :

$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

Μπορεί να χρησιμοποιηθούν και οι αποστάσεις :

- Manhattan απόσταση :  $\sum_{i=1}^k |x_i - x'_i|$
- Hamming απόσταση :  $D_H = \sum_{i=1}^k |x_i - x'_i|$  όπου αν  $x = y \rightarrow D = 0$ , ενώ αν  $x \neq y \rightarrow D = 1$ , για κατηγορικές μεταβλητές.
- Minkowski απόσταση :  $\left( \sum_{i=1}^k (|x_i - x'_i|)^q \right)^{\frac{1}{q}}$

όπου  $x, x'$  δύο διαφορετικές παρατηρήσεις από το σύνολο δεδομένων μας και  $k$ , ο αριθμός των επεξηγηματικών μεταβλητών.

Πριν την εφαρμογή της μεθόδου είναι καλό να κανονικοποιούμε ή να τυποποιούμε τα δεδομένα μας, για να αποφεύγονται οι επιρροές στην απόσταση λόγω διαφοράς μετρικής κλίμακας μεταξύ των επεξηγηματικών μεταβλητών.

Ένα σημαντικό ερώτημα στον αλγόριθμο αυτό είναι η επιλογή της μεταβλητής  $K$ , δηλαδή από πόσους κοντινότερους γείτονες θα υπολογιστεί η τιμή της μεταβλητής που ψάχνουμε. Όπως μπορούμε να φανταστούμε, αν επιλέξουμε έναν μόνο γείτονα θα οδηγηθούμε σε υπερ εκπαίδευση στο σύνολο των δεδομένων εκπαίδευσης, ενώ με μία μεγάλη τιμή του  $K$ , για παράδειγμα 20, ο αλγόριθμος δεν θα λειτουργεί καλά σε κανένα σύνολο δεδομένων. Δηλαδή, όσο περισσότερο πλησιάζει στη μονάδα ο αριθμός  $K$ , τόσο λιγότερο ευσταθείς γίνονται οι προβλέψεις, επηρεάζονται από τα δεδομένα και τυχόν διακυμάνσεις σε αυτά.

Η μέθοδος των  $K$ - κοντινότερων γειτόνων, μπορεί να είναι απλή στην κατανόηση και την εφαρμογή, έχει όμως μεγάλη υπολογιστική πολυπλοκότητα καθώς καλείται να υπολογίζει την απόσταση μεταξύ ενός σημείου και όλων των υπολοίπων στο σύνολο δεδομένων. Η μέθοδος αυτή, καλείται *brute-force* - γνωστή και ως εξερεύνηση φασικού χώρου - χρειάζεται χρόνο  $O[DN^2]$  για να υλοποιηθεί, όπου  $D$  : ο αριθμός των επεξηγηματικών μεταβλητών και  $N$  : το πλήθος του συνόλου δεδομένων. Για την εύρεση της κατάλληλης τιμής μπορεί να χρησιμοποιηθεί ένα διάγραμμα οπτικοποίησης της βέλτιστης τιμής γνωστό ως 'elbow curve', το οποίο μας δείχνει το κατάλληλο αριθμό γειτόνων που πρέπει να χρησιμοποιήσουμε για τα δεδομένα μας. Η *brute-force* είναι πολύ ανταγωνιστική και αποτελεσματική για μικρά σύνολα δεδομένων αλλά για μεγάλο σύνολο δεδομένων ( μεγάλο  $N$  ) η μέθοδος γίνεται υπολογιστικά αδύνατη.

Για να αντιμετωπιστούν οι υπολογιστικές ανεπάρκειες της προσέγγισης *brute-force*, έχουν επινοηθεί ποικίλες δομές δεδομένων που βασίζονται σε δέντρα και επιδιώκουν να μειώσουν τον απαιτούμενο αριθμό υπολογισμό αποστάσεων, με την αποτελεσματική κωδικοποίηση πληροφοριών σχετικά με την απόσταση για

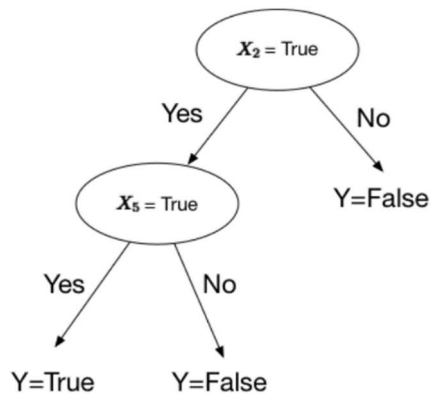
το δείγμα. Μία τέτοια δομή είναι η *K-D tree* που βασίζεται στην ιδέα ότι αν ένα σημείο A είναι απομακρυσμένο από το σημείο B και το B είναι πολύ κοντά σε ένα σημείο Γ, τότε ξέρουμε ήδη ότι τα σημεία A και Γ είναι απομακρυσμένα, χωρίς να χρειάζεται να το υπολογίσουμε. Έτσι το υπολογιστικό κόστος μειώνεται στο  $O[D \log(N)]$  και η βελτίωση σε σχέση με την *brute-force* είναι αισθητή. Αυτό γίνεται μέσω της δυαδικής δομής δέντρου που χωρίζει αναδρομικά το χώρο των παραμέτρων σε περιοχές με συγκεκριμένες παρατηρήσεις και έτσι η κατασκευή ενός KD-δέντρου είναι πολύ γρήγορη γιατί η διαμέριση του χώρου εκτελείται κατά μήκος των αξόνων δεδομένων και δεν χρειάζεται ο υπολογισμός D- διαστάσεων και η κάθε αναζήτηση απαιτεί χρόνο  $O[\log(N)]$ . Παρόλα αυτά η δομή γίνεται αναποτελεσματική για διαστάσεις μεγαλύτερες από 20 ( $D > 20$ )<sup>[8]</sup>.

Για την αντιμετώπιση της ανεπάρκειας της δομής των KD- δέντρων σε μεγάλες διαστάσεις, σχεδιάστηκε η *ball tree* μέθοδος. Σε αντίθεση με τα KD- δέντρα, που χωρίζουν τα δεδομένα κατά μήκος των αξόνων, τα *ball trees* χωρίζουν τα δεδομένα σε μία σειρά από εμφωλευμένες σφαίρες. Αυτό τα κάνει την κατασκευή τους πιο ακριβή από τα KD- δέντρα, αλλά τα είναι αποδοτικά σε δομημένα δεδομένα, ακόμα και σε μεγάλες διαστάσεις. Ένα *ball tree* χωρίζει αναδρομικά τα δεδομένα σε κόμβους που καθορίζονται από ένα κέντρο C και μία ακτίνα r, έτσι ώστε κάθε σημείο του κόμβου να βρίσκεται μέσα στη σφαίρα (C,r). Ο αριθμός των υποψήφιων σημείων για μία αναζήτηση γειτόνων περιορίζεται μέσω της τριγωνικής ανισότητας :  $|x + y| \leq |x| + |y|$ . Έτσι, η απόσταση μεταξύ ενός σημείου και του κέντρου της σφαίρας είναι επαρκής για να καθοριστούν ένα άνω και κάτω όριο των αποστάσεων όλων των σημείων που βρίσκονται στον ίδιο κόμβο.

### 2.2.3 | Δέντρα Απόφασης

Τα δέντρα απόφασης ανήκουν όχι μόνο στα εργαλεία των τεχνικών κατηγοριοποίησης (classification) αλλά και στα εργαλεία παλινδρόμησης (regression) και αποτελούν ένα από τα πιο αποτελεσματικά εργαλεία για την αντιμετώπιση της μη γραμμικότητας σε ένα μοντέλο παλινδρόμησης. Ο στόχος είναι η ανάπτυξη ενός μοντέλου, το οποίο θα μπορεί να κατηγοριοποιήσει μελλοντικά δεδομένα. Πρόκειται για ισχυρά εργαλεία που εκφράζονται ως αναδρομική διχοτόμηση του χώρου γεγονότων και λειτουργούν με διαδοχικές εντολές «If-Then». Ξεκινώντας από ένα σύνολο δεδομένων, μπορούν να ταξινομήσουν ή να προβλέψουν τη μεταβλητή απόκρισης, σύμφωνα με τις υπόλοιπες παραμέτρους, με χρήση δενδροδιαγράμματος. Ένα δέντρο απόφασης έχει κατεύθυνση που ξεκινά από έναν κόμβο που ονομάζεται ρίζα, ο οποίος δεν έχει καμία εισερχόμενη ακμή. Όλοι οι άλλοι κόμβοι έχουν ακριβώς μία εισερχόμενη ακμή. Κάθε κόμβος με εξερχόμενη ακμή, αναφέρεται ως ένας

κόμβος «εσωτερικός» ή «εξετάσης». Όλοι οι άλλοι κόμβοι ονομάζονται "φύλλα". Στο δέντρο απόφασης, κάθε εσωτερικός κόμβος χωρίζει το χώρο του γεγονότος σε δύο ή περισσότερους υπό-χώρους σύμφωνα με μία διακριτή συνάρτηση των αξιών των χαρακτηριστικών εισόδου. Στην περίπτωση των αριθμητικών χαρακτηριστικών, η κατάσταση αναφέρεται σε ένα εύρος. Σε κάθε φύλλο έχει αντιστοιχηθεί μία κατηγορία η οποία αναπαριστά την κατάλληλη αξία της μεταβλητής στόχου. Κάθε κόμβος αποτελεί έλεγχο γνωρισμάτων και κάθε κλαδί που τον συνδέει με τα υπόλοιπα αντιστοιχεί σε μία πιθανή τιμή για τη



μεταβλητή απόκρισης. Η διαδικασία που ακολουθείται σε ένα δέντρο απόφασης είναι συγκεκριμένη. Ξεκινώντας από τη ρίζα του δέντρου διασπούμε σε κάθε κόμβο τα δεδομένα με βάση τη βέλτιστη ιδιότητα του κόμβου που υπολογίζεται με κάποια κριτήρια όπως το Gini index. Έτσι, σε κάθε επόμενο κόμβο αποκτούμε συνεχώς μεγαλύτερη ομοιογένεια καταλήγοντας τελικά στα φύλλα που περιέχουν τις κλάσεις του προβλήματος. Σε περίπτωση αριθμητικών χαρακτηριστικών,

τα δέντρα απόφασης μπορούν να ερμηνευτούν γεωμετρικά ως μια συλλογή από υπερεπίπεδα, κάθε ένα ορθογώνιο προς έναν από τους άξονες. Τα φύλλα στα οποία φτάνει είναι οι ελάχιστες τιμές στις οποίες έχει καταλήξει, μετά από αυτόματο κλάδεμα ή pruning του αλγορίθμου. Για να ελέγξουμε την απόδοση ενός δέντρου απόφασης δίνουμε ως είσοδο τις τιμές του διανύσματος τιμών και περιμένουμε τις απαντήσεις. Το πλήθος των σωστών απαντήσεων καθορίζει την ακρίβεια του δέντρου.

Μία από τις πιο ευρέως χρησιμοποιούμενες μεθόδους για κατασκευή δέντρων αποφάσεων είναι η Classification And Regression Tree (CART) του Leo Breiman το 1984.

Ας δούμε λίγο αναλυτικά πώς δουλεύει ένα δέντρο απόφασης σε ένα πρόβλημα παλινδρόμησης. Όπως προαναφέρθηκε ένα δέντρο απόφασης είναι χτισμένο από την κορυφή προς τα κάτω, από έναν κόμβο ρίζα προς τα φύλλα. Ξεκινάει από τη ρίζα και διαιρεί τα δεδομένα σε όσο το περισσότερο ομοιογενή υποσύνολα. Για τον υπολογισμό της ομοιογένειας, σε διάφορους αλγορίθμους χρησιμοποιείται ο υπολογισμός της τυπικής απόκλισης, καθώς εάν το αριθμητικό δείγμα είναι ομοιογενές, η τυπική απόκλιση είναι μηδέν. Ταυτόχρονα, κάθε φορά που δημιουργείται ένα επίπεδο, υπολογίζεται και ο συντελεστής διακύμανσης, που θα χρησιμοποιηθεί σαν κριτήριο τερματισμού. Στο τέλος τα φύλλα του δέντρου δημιουργούνται από τη μέση τιμή των εμπλεκόμενων παρατηρήσεων. Η κατασκευή ενός δέντρου απόφασης αφορά την εύρεση μεταβλητών που επιστρέφουν την υψηλότερη μείωση της τυπικής απόκλισης – δηλαδή τα πιο ομοιογενή κλαδιά. Αρχικά, υπολογίζεται η τυπική απόκλιση της

μεταβλητής απόκρισης. Μετά τη δημιουργία ενός επιπέδου διασπώντας ένα χαρακτηριστικό, υπολογίζεται η τυπική απόκλιση για κάθε ένα και αφαιρείται από την υπολογισμένη τυπική απόκλιση πριν τον διαχωρισμό. Το αποτέλεσμα είναι η μείωση της τυπικής απόκλισης ή Standard Deviation Reduction (SDR). Το χαρακτηριστικό με την μεγαλύτερη μείωση τυπικής απόκλισης επιλέγεται για φύλλο του δέντρου απόφασης. Τα δεδομένα διαιρούνται με βάση τις τιμές του επιλεγμένου χαρακτηριστικού. Η διαδικασία εκτελείται αναδρομικά στα κλαδιά, μέχρι όλα τα δεδομένα να επεξεργαστούν. Πρακτικά, χρειαζόμαστε κάποια κριτήρια τερματισμού, για παράδειγμα όταν ο συντελεστής συσχέτισης είναι μικρότερος από ένα συγκεκριμένο κατώφλι ή/και όταν έχουν μείνει πολύ λίγα ανεπεξεργαστα δεδομένα σε κάθε κλαδί.

Πώς όμως φτιάχνεται ένα δέντρο απόφασης από το μηδέν; Έστω ότι έχουμε ένα σύνολο δεδομένων όπου θέλουμε να προβλέψουμε τη μεταβλητή απόκρισης, σύμφωνα με μία επεξηγηματική μεταβλητή. Ξεκινώντας από τις δύο μικρότερες παρατηρήσεις, υπολογίζουμε τον μέσο όρο τους και σύμφωνα με αυτόν χωρίζουμε τις μεταβλητές σε δύο κατηγορίες, εκείνες που είναι μικρότερες από τον μέσο όρο τους και εκείνες που είναι μεγαλύτερες. Έτσι, δημιουργούμε ένα δέντρο απόφασης βάθους ένα, το οποίο σαν ρίζα έχει τον μέσο όρο τους και σαν φύλλα, τον μέσο όρο των παρατηρήσεων της κάθε ομάδας που χωρίσαμε, δηλαδή οι πρώτες προβλέψεις. Στη συνέχεια, υπολογίζουμε το άθροισμα των τετραγωνικών υπολοίπων κάθε παρατήρησης, το οποίο δίνεται από τον τύπο :

$$\sum_{i=1}^n (y_i - \hat{y})^2$$

όπου  $\hat{y}$  , η τιμή πρόβλεψης, δηλαδή ο μέσος όρος των παρατηρήσεων της ομάδας στην οποία ανήκει η παρατήρηση  $y_i$  .

Στη συνέχεια επικεντρωνόμαστε στις επόμενες δύο μικρότερες παρατηρήσεις και επαναλαμβάνουμε την διαδικασία. Από κάθε ζευγάρι παρατηρήσεων κρατάμε το άθροισμα των τετραγωνικών υπολοίπων που υπολογίστηκε. Αφού υπολογίσαμε το ζητούμενο άθροισμα για όλα τα ζευγάρια παρατηρήσεων, βρίσκουμε το ελάχιστο και αυτό θα είναι η ρίζα του τελικού μας δέντρου. Αφού βρούμε τη ρίζα του δέντρου, βλέπουμε πόσες παρατηρήσεις θα υπάρχουν σε κάθε κόμβο, σύμφωνα με τη ρίζα για να δούμε αν θα διασπάσουμε τον κόμβο και άλλο ή αν θα γίνει φύλλο. Εκεί ανάλογα με τον χρησιμοποιούμενο αλγόριθμο υπάρχει ένα κατώφλι σύμφωνα με το οποίο θα διασπαστεί ή όχι ο κόμβος. Αυτό είναι το κριτήριο των ελαχίστων παρατηρήσεων ανά κόμβο (minimum samples leaf). Εάν δεν χρησιμοποιούνταν κάποιο κατώφλι και κάθε κόμβος διασπώταν ανεξαρτήτως του πλήθους των παρατηρήσεων που του αντιστοιχούν, θα καταλήγαμε σε ένα δέντρο με μεγάλο βάθος και ακρίβεια, δηλαδή θα είχαμε ένα αμερόληπτο (unbiased) δέντρο με μεγάλη διακύμανση (Variance). Θα οδηγούμασταν, δηλαδή, σε υπερπροσαρμογή των δεδομένων, αφού στα φύλλα

θα υπήρχαν ελάχιστες παρατηρήσεις και το μοντέλο θα είχε μεγάλο σφάλμα στην προσπάθεια πρόβλεψης των τιμών, λόγω του ότι θα βασιζόταν σε πολύ συγκεκριμένες παρατηρήσεις για την πρόβλεψη. Εάν, οι παρατηρήσεις που αντιστοιχούν σε έναν κόμβο είναι περισσότερες από το κατώφλι, τις διασπούμε και πάλι με την προηγούμενη διαδικασία, υπολογίζοντας το ελάχιστο από τα αθροίσματα τετραγωνικών υπολοίπων ανά ζευγάρι παρατηρήσεων και θέτοντάς το ως ερώτημα διάσπασης. Η διαδικασία τερματίζεται όταν το δέντρο έχει λιγότερες από το κατώφλι παρατηρήσεις σε κάθε κλαδί, δηλαδή όταν το δέντρο έχει καταλήξει σε φύλλα απόφασης στο κατώτερο επίπεδο του. Το κατώφλι μπορεί να προσδιοριστεί ελέγχοντας για διάφορες τιμές αν το ποσοστό ακρίβειας της πρόβλεψης για τα δεδομένα εκπαίδευσης και επαλήθευσης είναι κοντινό.

Στην περίπτωση που έχουμε παραπάνω από μία επεξηγηματική μεταβλητή στο σύνολο δεδομένων μας, η παραπάνω διαδικασία επαναλαμβάνεται για κάθε μεταβλητή ξεχωριστά. Ξεκινώντας από μία, βρίσκουμε το ελάχιστο από τα αθροίσματα των τετραγωνικών υπολοίπων ανά ζευγάρι και ο μέσος όρος παρατηρήσεων που μας οδήγησε σε αυτό, θα είναι υποψήφιο για ρίζα του δέντρου. Το ίδιο γίνεται και για τις υπόλοιπες επεξηγηματικές μεταβλητές και συγκρίνοντας τα αθροίσματα τετραγωνικών υπολοίπων του κάθε υποψηφίου, καταλήγουμε στον υποψήφιο με το ελάχιστο, ο οποίος θα είναι η ρίζα του δέντρου. Μεγαλώνουμε το δέντρο με την ίδια διαδικασία μόνο που πλέον θα πρέπει σε κάθε βήμα να συγκρίνουμε τα αθροίσματα των τετραγωνικών υπολοίπων κάθε μεταβλητής και να επιλέγουμε το μικρότερο για κόμβο διάσπασης.

Άλλη μία τεχνική που χρησιμοποιείται για τη βελτιστοποίηση ενός δέντρου απόφασης και την αποφυγή της υπερεκπαίδευσης είναι το κλάδεμα (pruning). Μετά την κατασκευή του δέντρου, οι κόμβοι που αν απομακρυνθούν μειώνουν το μέσο τετραγωνικό σφάλμα ή οι τελικοί κόμβοι με πολύ λίγες παρατηρήσεις, αφαιρούνται από το δέντρο. Στο τελικό δέντρο η μέθοδος υπολογίζει στατιστικά για κάθε φύλλο και ως τελική πρόβλεψη της μεταβλητής απόκρισης, θεωρείται η μέση τιμή του κόμβου.

Στις επόμενες δύο υπο-ενότητες θα μιλήσουμε για τη μεθοδολογία ώθησης ή boosting και δύο από τους πιο γνωστούς αλγορίθμους της.

## Μεθοδολογία Boosting - Ενδυνάμωσης

Στην προσπάθεια να προβλέψουμε την επιθυμητή μεταβλητή με οποιαδήποτε τεχνική τεχνητής νοημοσύνης υπάρχουν κάποιοι κύριοι λόγοι που οδηγούν στην απόκλιση της πραγματικής από την προβλεπόμενη τιμή της. Τέτοιοι λόγοι

μπορεί να είναι ο θόρυβος (noise), η διακύμανση (variance) και η μεροληψία (bias). Ένας τρόπος ελαχιστοποίησης της διακύμανσης και της μεροληψίας είναι η μέθοδοι συνόλων στις οποίες γίνεται ομαδοποίηση κάποιων χαρακτηριστικών τιμών όπως η μέση τιμή για να οδηγήσουν στην τελική πρόβλεψη. Μία τέτοια τεχνική είναι και η Boosting, στην οποία οι μεταβλητές πρόβλεψης δεν φτιάχνονται ανεξάρτητα αλλά με εξάρτηση ο ένας από τον άλλον. Συγκεκριμένα, αυτή η τεχνική χρησιμοποιεί τη λογική στην οποία οι επόμενες μεταβλητές πρόβλεψης μαθαίνουν από τα λάθη των προηγούμενων στη σειρά και σκοπό έχει να συνδυάσει το αποτέλεσμα ενός εκτιμητή για την κατασκευή ενός άλλου πιο ισχυρού. Συνεπώς, οι παρατηρήσεις του συνόλου δεδομένων έχουν διαφορετική πιθανότητα να εμφανιστούν στο επόμενο μοντέλο και μάλιστα μεγαλύτερη πιθανότητα να εμφανιστούν έχουν οι παρατηρήσεις με το μεγαλύτερο σφάλμα. Με αυτήν την τεχνική όπου οι επόμενες μεταβλητές πρόβλεψης μαθαίνουν από τα λάθη των προηγούμενων, η σωστή πρόβλεψη βρίσκεται σε λιγότερο χρόνο και λιγότερες επαναλήψεις. Ένα βασικό μειονέκτημα της τεχνικής, είναι ότι ο χρήστης πρέπει να επιλέξει με προσοχή το κριτήριο τερματισμού της, διότι τα δεδομένα εκπαίδευσης μπορούν να οδηγηθούν στην υπερεκπαίδευση. Η μέθοδος μηχανικής μάθησης boosting ή ώθησης, χρησιμοποιεί έναν πληθυσμό από ασθενείς κατηγοριοποιητές (weak classifiers/learners) με στόχο έναν ισχυρό κατηγοριοποιητή (strong classifier/learner) (Freund & Schapire, 1999· Zhou, 2012), όπως περιγράφεται παρακάτω. Ένας κατηγοριοποιητής καλείται ασθενής, όταν δίνει περίπου τόσες σωστές απαντήσεις, όσες θα έδινε μια τυχαία επιλογή των απαντήσεων, δηλαδή με ποσοστό κοντά στο 50%, ενώ ένας κατηγοριοποιητής καλείται ισχυρός, όταν μπορεί να προσεγγίσει το σύνολο των ορθών απαντήσεων με ακρίβεια. Ένας ευρέως χρησιμοποιούμενος ασθενής κατηγοριοποιητής για τις μεθόδους ώθησης, είναι για παράδειγμα ένα ρηχό δέντρο – μέχρι 5 κόμβους βάθος. Ο ισχυρός κατηγοριοποιητής που προκύπτει έτσι συνήθως μειώνει όχι μόνο τη μεροληψία (bias), αλλά και τη διασπορά (variance) της ικανότητας γενίκευσης ενός μεμονωμένου κατηγοριοποιητή.

Τα πιο συνηθισμένα μοντέλα τα οποία συνδυάζονται με την τεχνική Boosting για την επιλογή των μεταβλητών πρόβλεψης είναι τα δέντρα απόφασης. Βασικοί αλγόριθμοι της Boosting τεχνικής είναι η AdaBoost και η Gradient Boosting με την οποία θα ασχοληθούμε και θα εφαρμόσουμε στα δεδομένα μας.

#### 2.2.4 | AdaBoost (Adaptive Boosting)

Η AdaBoost μέθοδος για να φτάσει στην τελική πρόβλεψη, χρησιμοποιεί ένα δάσος από δέντρα ή Forest of Trees, τα οποία είναι αδύναμοι ταξινομητές και αποτελούνται συνήθως από τη ρίζα και δύο φύλλα, τα οποία λέγονται Stumps ή κούτσουρα. Προφανώς τα stumps δεν είναι καλοί ταξινομητές, διότι



καλούνται να πάρουν μία απόφαση μόνο με μία ερώτηση. Στα δέντρα από stumps που φτιάχνονται με την Adaboost μέθοδο, κάποια ρηχά δέντρα έχουν περισσότερο αντίκτυπο στην τελική απόφαση, δηλαδή η απόφαση τους έχει μεγαλύτερο βάρος και όπως αναφέρθηκε, τα επόμενα δέντρα μαθαίνουν από τα λάθη των προηγούμενων στη σειρά.

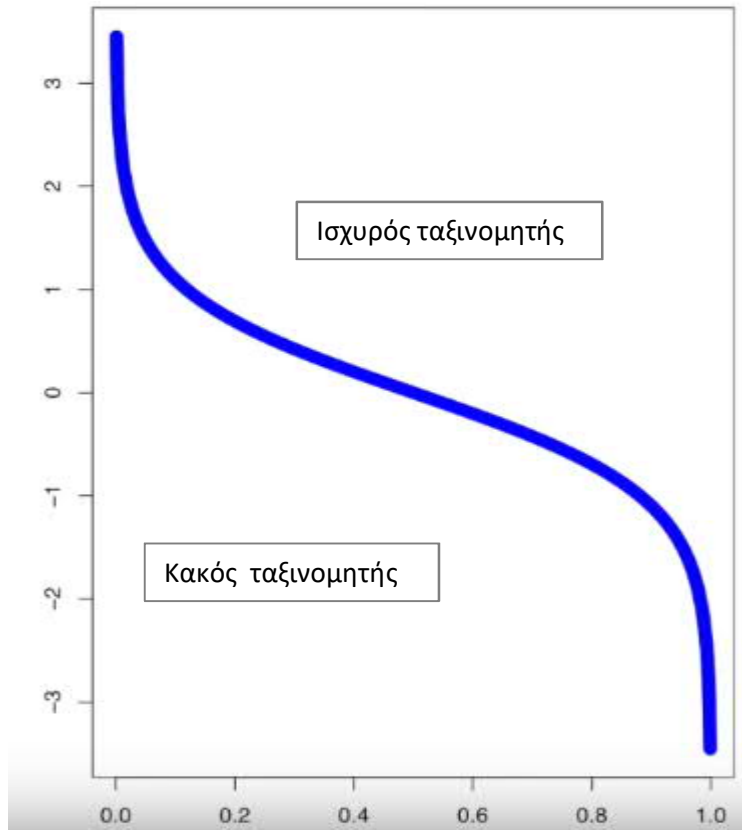
Για να κατασκευάσουμε ένα δάσος από ρηχά δέντρα σε ένα σύνολο δεδομένων και να χρησιμοποιήσουμε τον αλγόριθμο ώθησης της Adaboost, πρέπει αρχικά να ορίσουμε τα βάρη σε κάθε μεταβλητή πρόβλεψης. Ξεκινάμε, με ίσα βάρη σε όλες τις μεταβλητές ίσο με  $1/n$  (πλήθος των παρατηρήσεων, οπότε όλες οι μεταβλητές σε αυτό το σημείο είναι εξίσου σημαντικές για να παρθεί η τελική απόφαση της πρόβλεψης).

$$W_t^1 = 1/n$$

Για να υπολογίσουμε τα βάρη, μετράμε για κάθε μεταβλητή απόκρισης τα λάθη της πρόβλεψης σε σχέση με τα σωστά. Υπολογίζουμε τον δείκτη Gini, για κάθε ρηχό δέντρο κάθε μεταβλητής απόκρισης και το δέντρο με τον μικρότερο δείκτη Gini θα είναι το πρώτο δέντρο του 'Ρηχού' Δάσους μας. Υπολογίζουμε το συνολικό σφάλμα του κάθε ρηχού δέντρου ή Total Error ως το άθροισμα του βάρους του που σχετίζεται με τα λάθη της πρόβλεψης που υπολογίσαμε πριν. Όλα τα βάρη των παρατηρήσεων πρέπει να αθροίζονται στη μονάδα, με τις παρατηρήσεις που προέβλεψε λάθος ένα ρηχό δέντρο να έχουν βάρος 0, καθώς δεν προσφέρουν στην μετέπειτα εκπαίδευση του δάσους και ένα τις παρατηρήσεις που ταξινομήσε λάθος να έχουν βάρος 1. Βέβαια στο πρώτο βήμα τα βάρη είναι ίσα μεταξύ τους και αθροίζονται στη μονάδα αλλά για τα υπόλοιπα βήματα είναι απαραίτητη η κανονικοποίηση. Ο τύπος που δίνει στην Adaboost τα σφάλματα είναι :

$$W_i^j = \frac{1}{2} \log \frac{1 - \sum_{k=1}^l W_i^k}{\sum_{k=1}^l W_i^k}, \text{ όπου } k \text{ το πλήθος των λάθος προβλέψεων}$$

Το πόσο θα επηρεάσει το δάσος το συγκεκριμένο ρηχό δέντρο θα φανεί από το βάρος που υπολογίστηκε παραπάνω. Όταν έχει κάνει καλή ταξινόμηση, δηλαδή τα λάθη του είναι λίγα, το βάρος θα είναι σχετικά μεγάλος θετικός αριθμός, αντίστοιχα αν το δέντρο έχει πετύχει 50% επιτυχία πρόβλεψης, το βάρος του θα είναι μηδενικό και τέλος αν τα λάθη του είναι πολλά σε σχέση με τις προβλέψεις του, τότε το βάρος θα είναι ένας μεγάλος αρνητικός αριθμός και η πρόβλεψή του θα δίνεται στο δάσος ως η αντίθετη από αυτή που έδωσε – δηλαδή ως η σωστή.



Όπως αναφέρθηκε κάθε παρατήρηση που ταξινομήθηκε λάθος, θα εκπαιδεύσει το επόμενο ρηχό δέντρο, συνεπώς το βάρος του θα είναι μεγαλύτερο από τις υπόλοιπες παρατηρήσεις. Αυτό γίνεται χρησιμοποιώντας τον τύπο :

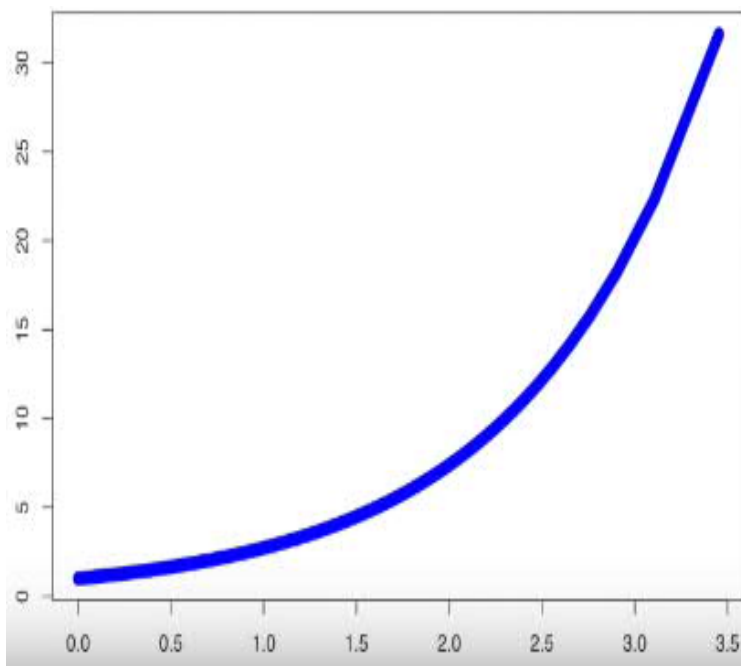
$$W_i^{j'} = W_i^j e^{W_i^j}$$

Η μετάφραση του τύπου είναι πως όταν το προηγούμενο ρηχό δέντρο έκανε καλή πρόβλεψη, συνεπώς το βάρος του θα είναι ένας μεγάλος θετικός αριθμός, τότε το νέο βάρος της παρατήρησης που δεν προέβλεψε καλά ο – κατά τα άλλα - καλός ταξινομητής θα πολλαπλασιαστεί με έναν μεγάλο θετικό αριθμό και θα γίνει μεγαλύτερο από το παλιό της βάρος. Αντίθετα, αν το προηγούμενο ρηχό δέντρο δεν προέβλεψε καλά τις παρατηρήσεις το νέο βάρος των λανθασμένων παρατηρήσεων θα πολλαπλασιαστεί με έναν μικρό αριθμό και θα αυξηθεί κατά πολύ λίγο. Έτσι οι παρατηρήσεις που προέβλεψε λάθος ένας καλός ταξινομητής θα έχουν μεγάλο βάρος, ενώ αντίθετα οι παρατηρήσεις που προέβλεψε λάθος ένας όχι και τόσο καλός ταξινομητής θα έχουν λιγότερο μεγάλο βάρος. Για τις παρατηρήσεις που προέβλεψαν σωστά οι ταξινομητές, θέλουμε να μειώσουμε

το βάρος τους καθώς δεν προσφέρουν κάτι στη συνέχεια της εκπαίδευσης, κάτι που γίνεται με τον τύπο :

$$W_i^{j'} = W_i^1 e^{-w_i^j}$$

Αντίστοιχα με πριν, αν ο ταξινομητής ήταν καλός θα πολλαπλασιάσουμε με μεγάλο αριθμό και το βάρος της σωστής παρατήρησης θα μειωθεί πολύ, ενώ αντίθετα αν δεν είναι θα πολλαπλασιάσουμε με μικρό αριθμό και το βάρος θα μειωθεί λίγο.



Πριν επεξεργαστεί ο επόμενος ταξινομητής τα νέα βάρη, σύμφωνα με τον προηγούμενο, πρέπει να κανονικοποιηθούν έτσι ώστε να αθροίζονται στη μονάδα. Στη συνέχεια, είτε με χρήση βεβαρημένου δείκτη Gini, ο οποίος θα επικεντρωθεί στη παρατήρηση με το μεγαλύτερο βάρος – δηλαδή εκείνη που ταξινομήθηκε λάθος, είτε δημιουργώντας ένα νέο σύνολο δεδομένων ίδιων διαστάσεων το οποίο θα περιέχει πολλαπλές φορές την παρατήρηση με το μεγαλύτερο βάρος, η διαδικασία συνεχίζεται. Εν τέλει, κάθε ένα σύνολο ρηχών δέντρων καταλήγουν σε μία πρόβλεψη, όπου αξιολογούνται σύμφωνα με το άθροισμα των βαρών των ρηχών δέντρων που περιέχει το σύνολο και επιλέγεται το μεγαλύτερο.

## 2.2.5 | Gradient Boosting

Η Gradient Boosting τεχνική παράγει ένα μοντέλο πρόβλεψης βασισμένο σε ένα σύνολο αδύναμων μοντέλων, όπως τα δέντρα απόφασης. Όπως σε όλες τους εποπτευόμενους αλγορίθμους έτσι και η Gradient Boosting έχει ως στόχο την ελαχιστοποίηση μιας ορισμένης συνάρτησης απώλειας, όπως το μέσο τετραγωνικό σφάλμα (MSE). Η κύρια ιδέα είναι να χρησιμοποιήσουμε μία αδύναμη μέθοδο εκμάθησης αρκετές φορές για να πάρουμε μια σειρά υποθέσεων και μοτίβων, τα οποία επικεντρώνονται εκ νέου στα παραδείγματα που προηγουμένως έχουν ταξινομηθεί λανθασμένα. Όπως και στο γραμμικό μοντέλο, η βασική προϋπόθεση για να θεωρήσουμε πετυχημένη την εφαρμογή ενός αλγορίθμου στα δεδομένα μας, είναι η μη συστηματική κατανομή των υπολοίπων γύρω από το μηδέν. Ωστόσο, αν παρατηρήσουμε ότι τα υπόλοιπα που έχουν παραχθεί από το μοντέλο πρόβλεψης μας, έχουν μία συστηματική

συμπεριφορά γύρω από το μηδέν ή ακολουθούν κάποιο μοτίβο, τότε μπορούμε να το αξιοποιήσουμε για να ταιριάξουμε τα δεδομένα καλύτερα στο προβλεπτικό μοντέλο. Η τεχνική πίσω από τον αλγόριθμο της Gradient Boosting, είναι η συνεχής αξιοποίηση των μοτίβων των υπολοίπων και η ενδυνάμωση του μοντέλου με αδύναμες προβλέψεις. Ένα κριτήριο τερματισμού του αλγορίθμου είναι όταν τα υπόλοιπα δεν έχουν κάποιο μοτίβο το οποίο μπορούμε να χρησιμοποιήσουμε για την βελτίωση του μοντέλου, διότι μετά οδηγούμαστε στην υπερεκπαίδευση του μοντέλου.

Στην Gradient Boost μέθοδο, σκοπός είναι να δημιουργήσουμε μια συνάρτηση  $F(x)$ , που θα αντιστοιχίζει το σύνολο των επεξηγηματικών μεταβλητών  $x$  με τη μεταβλητή απόκρισης  $y$ , έτσι ώστε να ελαχιστοποιείται μια συγκεκριμένη συνάρτηση κόστους (loss function), όπως το μέσο τετραγωνικό σφάλμα

$$\frac{1}{n} \sum_{i=1}^n \{y_i - \hat{y}_i\}^2,$$

όπου  $n$  ο αριθμός των συνολικών παρατηρήσεων που διαθέτουμε στο σύνολο εκπαίδευσης,  $y_i$  οι εκτιμήσεις της μεταβλητής απόκρισης που έδωσε η  $F(x)$ , για δεδομένα  $x$ , και  $\hat{y}_i$ , οι πραγματικές τιμές των εκτιμήσεων αυτών.

Σε κάθε επανάληψη  $m$ , όπου  $1 \leq m \leq M$ , η μέθοδος, υποθέτει ότι υπάρχει ένα αδύναμο – με μικρή ακρίβεια(μεγάλη απόκλιση), μοντέλο  $F_m$ , που προβλέπει τη μεταβλητή  $y$ , και προχωράει στη βελτίωση του μοντέλου αυτού, κατασκευάζοντας ένα νέο καλύτερο με μικρότερο σφάλμα μοντέλο  $F_{m+1}$  χάρη σε έναν επιπλέον εκτιμητή  $h$

$$F_{m+1}(x) = F_m(x) + h(x)$$

Για τον επιπλέον εκτιμητή  $h$ , θέλουμε να ισχύει

$$F_{m+1}(x) = F_m(x) + h(x) = y \text{ ή ισοδύναμα } h(x) = y - F_m(x).$$

Συνεπώς, η gradient boosting προσαρμόζει τον όρο  $h$  με βάση το υπόλοιπο  $y - F_m(x)$ , έτσι ώστε κάθε όρος  $F_{m+1}$  να προσπαθεί να διορθώσει το σφάλμα του προηγούμενου όρου  $F_m$ , βασιζόμενη στην ιδέα ότι τα υπόλοιπα  $y - F_m(x)$ , είναι αρνητικές κλίσεις του τετραγωνικού σφάλματος (συνάρτηση κόστους – loss function),  $\frac{1}{2}(y - F_m(x))^2$ . Έτσι η gradient boosting βασίζεται στην έννοια της απότομης καθόδου (gradient descent).

Ένα σημαντικό κομμάτι της μεθόδου, είναι η κανονικοποίηση ( regularization ) μέσω της συρρίκνωσης ( shrinkage ), δηλαδή τη μετατροπή του μοντέλου σύμφωνα με τον παρακάτω τρόπο

$$F_m(x) = F_{m-1}(x) + v \sum_{j=1}^{J_m} \gamma_{jm} 1_{R_{jm}}(x), 0 \leq v \leq 1$$

όπου η παράμετρος  $v$ , ονομάζεται βαθμός εκμάθησης(learning rate). Έχει παρατηρηθεί ότι σε μοντέλα της Gradient Boosting στα οποία έχει εφαρμοστεί συρρίκνωση για μικρούς βαθμούς εκμάθησης,  $v < 0.1$ , τείνουν να είναι αρκετά

πιο γενικευμένα σε σχέση με μοντέλα στα οποία δεν έχει εφαρμοστεί συρρίκνωση,  $v=1$ . Με άλλα λόγια, η συρρίκνωση για μικρά  $v$ , συμβάλλει στην αποφυγή του φαινομένου της υπερπροσαρμογής (overfitting) στα δεδομένα εκπαίδευσης. Έτσι, ένα μοντέλο στο οποίο έχει εφαρμοστεί συρρίκνωση, μπορεί να είναι πολύ πιο αποδοτικό σε καινούρια δεδομένα από ότι ένα μοντέλο στο οποίο δεν έχει εφαρμοστεί συρρίκνωση. Ωστόσο, η εφαρμογή της συρρίκνωσης στη μέθοδο, οδηγεί σε αύξηση του υπολογιστικού χρόνου, αφού απαιτούν περισσότερες επαναλήψεις. Ένα Gradient Boosting μοντέλο επηρεάζεται από δύο παραμέτρους, το μέγεθος ή βάθος των δέντρων και ο αριθμός των δέντρων, δηλαδή των αδύναμων εκτιμητών που θα κατασκευαστούν. Λόγω του ότι θέλουμε εκπαίδευση μόνο αδύναμων ταξινομητών με τη μέθοδο, το βάθος δεν θα πρέπει να ξεπερνάει τα 5. Όμοια, ο αριθμός των εκτιμητών-δέντρων, δεν πρέπει να είναι πολύ μεγάλος. Εάν επιλέξουμε μεγάλο αριθμό για αυτές τις παραμέτρους, είναι πιθανό να οδηγηθούμε σε φαινόμενο υπερπροσαρμογής γι' αυτό πρέπει να υπάρχει κατάλληλος έλεγχος. Ένας από τους πιο ασφαλής τρόπους για την επιλογή των παραμέτρων της μεθόδου, είναι ο έλεγχος για το ποιες οδηγούν σε καλύτερες προβλέψεις εξασφαλίζοντας κάθε φορά, ότι δεν υπάρχει υπερπροσαρμογή στα δεδομένα εκπαίδευσης.

Η μέθοδος ξεκινάει υπολογίζοντας το μέσο όρο της μεταβλητής απόκρισης  $y$  και τον θέτει ως αρχική εκτίμηση  $\hat{y}$ . Στη συνέχεια, υπολογίζει τα ψευδοϋπόλοιπα ( $y - \hat{y}$ ) για κάθε παρατήρηση και εκπαιδεύει πάνω σε αυτά το πρώτο δέντρο απόφασης. Αφού κατασκευαστεί το δέντρο, η πρόβλεψη για κάθε παρατήρηση θα είναι ο μέσος όρος των εκτιμήσεων του δέντρου (ψευδοϋπολοίπων) που βρίσκονται στον κόμβο που αντιστοιχεί η παρατήρηση αυτή. Στη συνέχεια οι τιμές των αρχικών ψευδοϋπολοίπων αντικαθίστανται από τις προβλέψεις τους. Η πρόβλεψη του μοντέλου για κάθε  $y$  προκύπτει ως άθροισμα του μέσου όρου των  $y$  και του ψευδοϋπόλοιπου που αντιστοιχεί στην παρατήρηση αυτή. Ωστόσο για να αποφύγουμε το overfitting προσθέτουμε έναν όρο, το ρυθμό εκμάθησης ο οποίος καθορίζει τη συνεισφορά του νέου δέντρου στην τελική πρόβλεψη. Έτσι η νέα πρόβλεψη δίνεται από το άθροισμα της μέσης τιμής της  $y$  με την τιμή του ψευδοϋπόλοιπου επί το ρυθμό εκμάθησης. Με τον τρόπο αυτό παίρνουμε κάθε φορά μία λίγο καλύτερη πρόβλεψη χωρίς να κινδυνεύουμε από υπερεκπαίδευση. Η διαδικασία συνεχίζεται, χτίζοντας το επόμενο δέντρο το οποίο θα εκπαιδευτεί στα ψευδοϋπόλοιπα που προκύπτουν ως η διαφορά των πραγματικών τιμών της μεταβλητής απόκρισης και των προβλέψεων που έκανε το προηγούμενο δέντρο. Η τελική πρόβλεψη κάθε φορά προκύπτει ως το άθροισμα του μέσου όρου των πραγματικών τιμών με τα ψευδοϋπόλοιπα του κάθε δέντρου για την παρατήρηση αυτή πολλαπλασιασμένα κάθε φορά με το ρυθμό εκμάθησης. Η διαδικασία σταματάει όταν φτάσουμε έναν καθορισμένο αριθμό δέντρων ή όταν τα ψευδοϋπόλοιπα δεν αλλάζουν σημαντικά.

## 2.2.6 | **Ομοιότητες & Διαφορές των μεθόδων AdaBoost, Gradient Boosting**

Τόσο η Adaboost, όσο και η Gradient Boost χρησιμοποιούν ως βάση αδύναμους εκτιμητές (weak learners), για τη δημιουργία δυνατών εκτιμητών (strong learners) και προσπαθούν να ενισχύσουν την απόδοση τους με την επαναληπτική μετατόπιση της προσοχής τους στα προβληματικά σημεία που δεν προέβλεψαν σωστά.

Η διαφορά τους ωστόσο εντοπίζεται στον τρόπο με τον οποίο κατασκευάζουν και βελτιώνουν τους εκτιμητές μέσα από την επαναληπτική διαδικασία. Η Gradient Boosting κατασκευάζει το τελικό μοντέλο σταδιακά, βελτιώνοντας μέσα από το κάθε βήμα το μοντέλο σύμφωνα με το προηγούμενο, ενώ η Adaboost συνδυάζει τα αποτελέσματα όλων των εκτιμητών για να καταλήξει στο τελικό μοντέλο.

Η Adaboost αυξάνει την απόδοση-ακρίβεια τους επιβαρύνοντας με μεγαλύτερα βάρη τις παρατηρήσεις που προβλέπονται δυσκολότερα (παρατηρήσεις που το μοντέλο έχει μεγαλύτερο σφάλμα) σε κάθε ένα από τους εκτιμητές και στο τέλος συνδυάζονται και με βάση το βάρος τους συνεισφέρουν ανάλογα στην κατασκευή του ισχυρού εκτιμητή. Η Gradient Boosting, κατασκευάζει κάθε εκτιμητή εκπαιδεύοντας τον με τα υπόλοιπα του προηγούμενου εκτιμητή και κάθε φορά η τελική πρόβλεψη του μοντέλου προκύπτει από το αποτέλεσμα του κάθε εκτιμητή πολλαπλασιασμένο επί το ρυθμό εκμάθησης και κάθε εκτιμητής βελτιώνει σταδιακά την ακρίβεια του μοντέλου αποφεύγοντας την υπερπροσαρμογή.

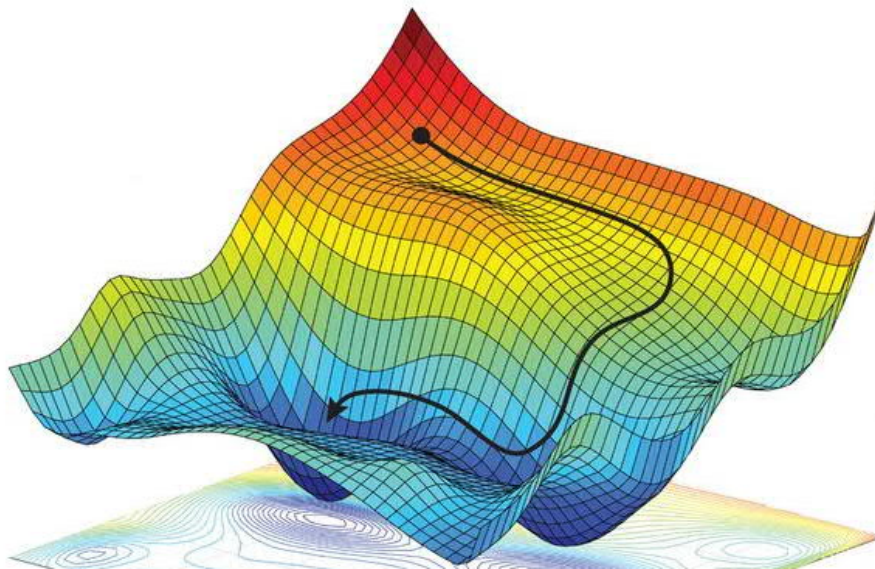
## 2.3 | **Μέθοδοι Βελτιστοποίησης κ Ελαχιστοποίησης**

### 2.3.1 | **Μέθοδος Απότομης Καθόδου ( Gradient Descent )**

Η μέθοδος Gradient Descent χρησιμοποιείται για να βελτιστοποιήσει παραμέτρους, δηλαδή να βρεθούν οι βέλτιστες παράμετροι μιας συνάρτησης. Βασίζεται σε επαναληπτικό αλγόριθμο και έχει σκοπό την ελαχιστοποίηση μιας συνάρτησης.

Χρησιμοποιώντας το παρακάτω γράφημα που αναπαριστά μία συνάρτηση, θα περιγράψουμε τη λειτουργία της μεθόδου. Θέλουμε να βρούμε το ελάχιστο της συνάρτησης με χρήση της απότομης καθόδου. Ξεκινάμε από το κόκκινο σημείο,

που αποτελεί μέγιστο της συνάρτησης και κατευθυνόμαστε προς τη κατεύθυνση που υποδεικνύει η αρνητική κλίση, επιλέγουμε άλλο σημείο και υπολογίζουμε την αρνητική κλίση και συνεχίζουμε επαναληπτικά μέχρι να φτάσουμε στο ελάχιστο σημείο.



Η Gradient Descent βασίζεται στο ότι αν έχουμε μία πολυδιάστατη συνάρτηση  $F(x)$ , ορισμένη και παραγωγίσιμη σε μία γειτονιά σημείου  $a$  τότε η  $F(x)$  μειώνεται γρηγορότερα αν μετακινηθούμε κατά την αρνητική κλίση της στο  $a$ .

Δηλαδή, αν:

$$a_{n+1} = a_n - \gamma \nabla F(a_n), \text{ για } \gamma \in R \text{ αρκετά μικρό,}$$

τότε  $F(a_n) \geq F(a_{n+1})$ . Ο όρος,  $\gamma \nabla F(a)$  έχει αφαιρεθεί από το  $a$ , γιατί, θέλουμε να προχωρήσουμε αντίθετα με την κλίση προς το ελάχιστο της συνάρτησης. Ξεκινάμε από ένα αρχικό  $x_0$ , το οποίο θεωρούμε ως τοπικό ελάχιστο της  $F$ , και θεωρούμε τα  $x_1, x_2, \dots$ , τέτοια ώστε

$$x_{n+1} = x_n - \gamma_n \nabla F(x_n), n \geq 0.$$

Έτσι προκύπτει μια μονότονη ακολουθία,  $F(x_0) \geq F(x_1) \geq F(x_2) \geq \dots$ , η οποία συγκλίνει στο τοπικό ελάχιστο που θέλουμε να βρούμε.

Με πιο απλά λόγια, η Gradient Descent κάνει λίγους υπολογισμούς όταν βρίσκεται μακριά από την βέλτιστη λύση και αυξάνει τον αριθμό των υπολογισμών όταν φτάνει πιο κοντά σε αυτή. Σε αντίθεση με τη μέθοδο ελαχίστων τετραγώνων δεν μηδενίζει τη παράγωγο της συνάρτησης για την εύρεση της βέλτιστης λύσης, αλλά υπολογίζοντας την τιμή της συνάρτησης ξεκινώντας από μία αρχική τιμή, συνεχίζει με χρήση βήματος για να φτάσει στην καλύτερη. Αυτό κάνει την Gradient Descent πολύ χρήσιμη για περιπτώσεις που δεν είναι δυνατόν να βρεθεί η τιμή που μηδενίζει την παράγωγο της. Δοκιμάζοντας λοιπόν, τιμές για τον μηδενισμό της παραγώγου όσο πιο κοντά

φτάνουμε στο μηδέν τόσο πιο κοντινές τιμές δοκιμάζονται από τη μέθοδο. Η τιμή του βήματος που καθορίζει εν τέλει τις τιμές που θα δοκιμαστούν, προσδιορίζεται πολλαπλασιάζοντας την τιμή της παραγώγου που βρέθηκε, με μία μικρή τιμή εκμάθησης ( $\gamma$  : learning rate). Ο ρυθμός εκμάθησης καθορίζει το μέγεθος κάθε βήματος που θα κάνει η μέθοδος. Ένας μεγάλος ρυθμός εκμάθησης, σημαίνει ότι σε κάθε βήμα μπορούμε να καλύψουμε μεγαλύτερο έδαφος, όμως ρισκάρουμε να προσπεράσουμε το ελάχιστο, καθώς η πλαγιά-κλίση (slope) σε κάθε σημείο αλλάζει. Από την άλλη, ένας μικρός ρυθμός εκμάθησης, σημαίνει ότι θα έχουμε μεγαλύτερη ακρίβεια στον υπολογισμό του ελαχίστου, όμως ο υπολογισμός της κλίσης καταναλώνει κάποιο χρόνο και χαμηλοί ρυθμοί εκμάθησης απαιτούν πολύ χρόνο για να βρεθεί το ελάχιστο της συνάρτησης.

Η νέα τιμή που θα δοκιμάσει η μέθοδος βρίσκεται αν από την παλιά τιμή που χρησιμοποιήθηκε αφαιρέσουμε το βήμα. Βέβαια στην πράξη όλες οι μέθοδοι υπολογίζουν αυτόματα το βέλτιστο βήμα, ξεκινώντας από ένα μεγάλο και κάνοντάς το μικρότερο σε κάθε επανάληψη.

Η μέθοδος Gradient Descent σταματάει όταν το βήμα είναι πολύ κοντά στο μηδέν, άρα όταν η παράγωγος της συνάρτησης που ψάχνουμε είναι πολύ κοντά στο μηδέν. Συνήθως η μέθοδος σταματάει όταν το βήμα είναι μικρότερο από 0.001. Ταυτόχρονα υπάρχει και όριο στα βήματα που η μέθοδος θα εκτελέσει πριν σταματήσει την προσπάθεια για εύρεση της τιμής που ελαχιστοποιεί την συνάρτηση.

Ας δούμε τη μέθοδο με βήματα:

- 1) Υπολογισμός παραγώγου συνάρτησης ελαχιστοποίησης που έχουμε επιλέξει, συναρτήσει κάθε παραμέτρου που περιλαμβάνει (Gradient of the loss function)
- 2) Επιλογή τυχαίων τιμών για τις μεταβλητές
- 3) Αντικατάσταση των παραπάνω τιμών στις μερικές παραγώγους που υπολογίσαμε
- 4) Υπολογισμός βήματος (βήμα = τιμή παραγώγου (αρχικά σε τυχαίο αρχικό σημείο) \* τιμή εκμάθησης)
- 5) Υπολογισμός νέων παραμέτρων (νέα παράμετρος = παλιά παράμετρος - βήμα)
- 6) Επανάληψη βημάτων 3, 4, 5 μέχρις ότου το βήμα να είναι πολύ μικρό ή μέχρι να φτάσουμε το μέγιστο επιτρεπτό αριθμό βημάτων



### 2.3.2 | **Στοχαστική Μέθοδος Απότομης Καθόδου** (Stochastic Gradient Descent)

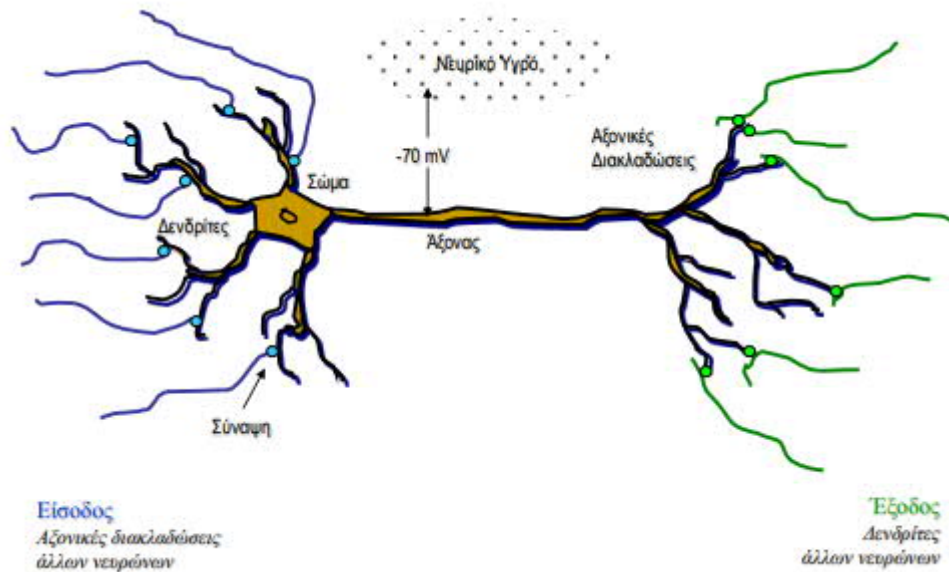
Για πολύπλοκα μοντέλα με πολλές παραμέτρους που εφαρμόζονται σε ένα μεγάλο σύνολο δεδομένων, η Gradient Descent δεν συμφέρει υπολογιστικά, καθώς υπολογίζει σε κάθε βήμα για όλα τα σημεία τις τιμές των μερικών παραγώγων για όλες τις παραμέτρους της συνάρτησης. Έτσι, χρησιμοποιείται η Stochastic Gradient Descent που επιλέγει ένα υποσύνολο των δεδομένων που έχουμε, αντί όλων των σημείων. Αν πρέπει να βρούμε το ελάχιστο μιας συνάρτησης με πολλές παραμέτρους για 1.000.000 δείγματα, τότε θα έπρεπε σε κάθε βήμα να υπολογίζαμε τους όρους για 1.000.000 σημεία για κάθε μερική παράγωγο της συνάρτησης ελαχιστοποίησης. Η Stochastic Gradient Descent επιλέγει τυχαία ένα ικανοποιητικό υποσύνολο σημείων του δείγματος, μάλιστα αν τα σημεία είναι ομαδοποιημένα τότε θα επιλέξει σωστό αριθμό σημείων από κάθε ομάδα σε κάθε βήμα. Έτσι στην πραγματικότητα χρησιμοποιείται η Stochastic Gradient Descent, αντί της απλής Gradient Descent.

## 2.4 | **Τεχνητά Νευρωνικά Δίκτυα** (Artificial Neural Network)

### 2.4.1 | **Η Έμπνευση των Νευρωνικών Δικτύων**

Η έμπνευση για κάθε μορφής νευρωνικό δίκτυο ξεκινά από την βιολογία και τον ανθρώπινο εγκέφαλο. Το νευρικό σύστημα του ανθρώπου είναι υπεύθυνο για διεργασίες όπως η επαφή με τον εξωτερικό κόσμο, η μάθηση, η μνήμη κλπ. Το νευρικό σύστημα του ανθρώπου αποτελείται από πολλά νευρωνικά δίκτυα τα οποία είναι εξειδικευμένα στις διεργασίες αυτές και το κέντρο του είναι ο εγκέφαλος. Κάθε νευρωνικό δίκτυο αποτελείται από νευρώνες, που είναι η πιο

μικρή ανεξάρτητη μονάδα του νευρικού δικτύου και επεξεργάζονται πληροφορίες, λαμβάνοντας και στέλνοντας σήματα σε άλλους νευρώνες.



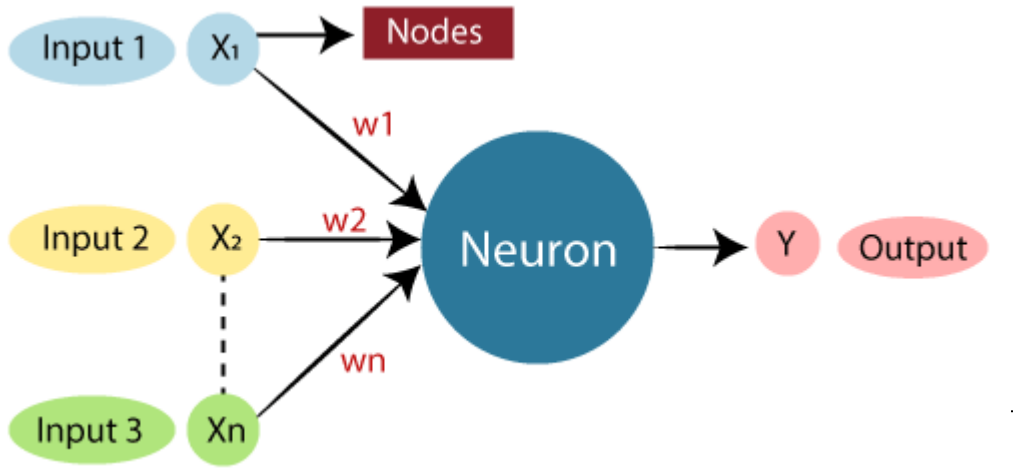
Οι επιστήμονες θέλουν και προσπαθούν να κατανοήσουν πλήρως τις λειτουργίες του εγκεφάλου και να καταφέρουν οι ηλεκτρονικοί υπολογιστές να κάνουν ό,τι και εκείνο. Η αναγνώριση φωνής ή εικόνας, είναι πράξεις που το μυαλό κάνει πολύ εύκολα και οι υπολογιστές έχουν πλέον καταφέρει σε ένα ικανοποιητικό σημείο. Πώς όμως έχουμε φτάσει σε αυτό το σημείο; Προσπαθώντας να φτιάξουμε έναν υπολογιστή με τέτοια εσωτερική δομή που να μοιάζει με την δομή του εγκεφάλου. Έτσι, δημιουργήθηκαν κάποια μοντέλα του ανθρώπινου νευρικού συστήματος, τα οποία μπορούν από μόνα τους να επιτελέσουν τις εργασίες αυτές, με τον ίδιο τρόπο που γίνονται στα βιολογικά νευρωνικά δίκτυα. Αυτά είναι τα τεχνητά νευρωνικά δίκτυα (artificial neural networks, ANN).

## 2.4.2 | Τα Τεχνητά Νευρωνικά Δίκτυα

### Η Λειτουργία του Νευρώνα

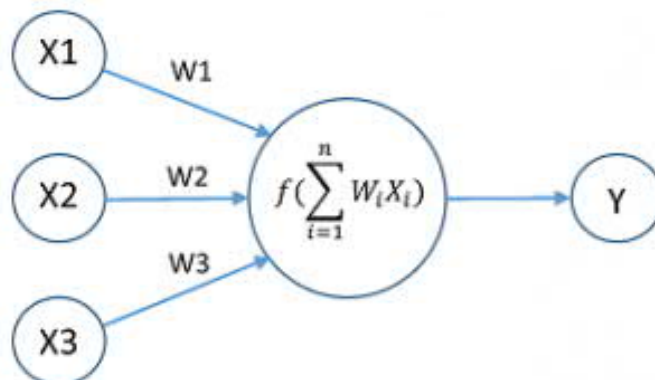
Ο πρωταρχικός σκοπός της λειτουργίας ενός τεχνητού νευρικού δικτύου είναι αφού εκπαιδευτεί με τη βοήθεια κατάλληλου συνόλου δεδομένων, να μπορεί να επιτελεί αυτόματα ορισμένες διεργασίες. Εφόσον η έμπνευση των τεχνητών νευρικών δικτύων είναι ο άνθρωπος εγκεφάλος, η αρχιτεκτονική τους έχει κοινά χαρακτηριστικά με αυτή του εγκεφάλου. Στον ανθρώπινο εγκέφαλο συναντάμε ένα σύνολο νευρώνων που συνδέονται μεταξύ τους και με κάθε ερέθισμα μεταδίδουν πληροφορίες, με σκοπό τα κατάλληλα εκτελεστικά όργανα να ενεργήσουν σύμφωνα με αυτές. Έτσι και τα τεχνητά νευρωνικά δίκτυα αποτελούνται από ένα σύνολο νευρώνων που αλληλεπιδρούν και μεταφέρουν

πληροφορίες σύμφωνα με μια προκαθορισμένη διαδρομή. Κάθε δίκτυο δέχεται συγκεκριμένες εισόδους και δίνει ορισμένες εξόδους (input-output). Η πληροφορία εισέρχεται στο τεχνητό νευρωνικό δίκτυο μέσω των σημάτων εισόδου (Input Layer), που αντιστοιχούν σε επεξηγηματικές μεταβλητές του μοντέλου που έχουμε αποφασίσει να εκπαιδεύσουμε, αντίστοιχο ρόλο στον εγκέφαλο έχουν οι δενδρίτες με τις αισθήσεις να έχουν τον ρόλο των μεταβλητών του μοντέλου.



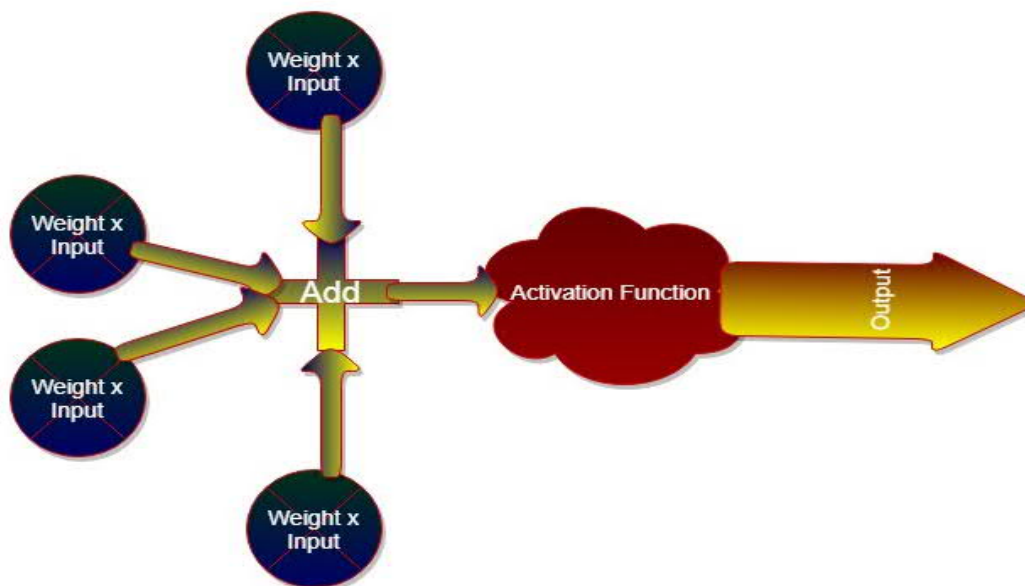
Έπειτα, η πληροφορία επεξεργάζεται από τον νευρώνα και παράγεται το αποτέλεσμα από το σήμα εξόδου, το οποίο μπορεί να έχει πολλές μορφές αναλόγως το πρόβλημα στο οποίο εργαζόμαστε.

Η επεξεργασία γίνεται ως εξής. Τα σήματα εισόδου αρχικοποιούν τη πληροφορία με κάποια βάρη και από εκεί οδηγούνται στο νευρώνα όπου υπολογίζεται το άθροισμα του γινομένου των βαρών επί των τιμών των επεξηγηματικών μεταβλητών, δηλαδή το  $\sum_{i=1}^n w_i x_i$ , όπου  $x$  η τιμή της επεξηγηματικής μεταβλητής και  $w$  το αντίστοιχο βάρος της. Στη συνέχεια, υπολογίζεται η τιμή  $f(\sum_{i=1}^n w_i x_i)$  μιας συνάρτησης ενεργοποίησης  $f$  για την τιμή του υπολογισμένου αθροίσματος και αναλόγως θα ενεργοποιηθεί ή όχι το νευρώνα. Αν ενεργοποιηθεί ο νευρώνας, τότε η τιμή  $y_i$  θα καταλήξει στο στρώμα εξόδου (Output Layer).



## Συναρτήσεις Ενεργοποίησης ( Activation Functions )

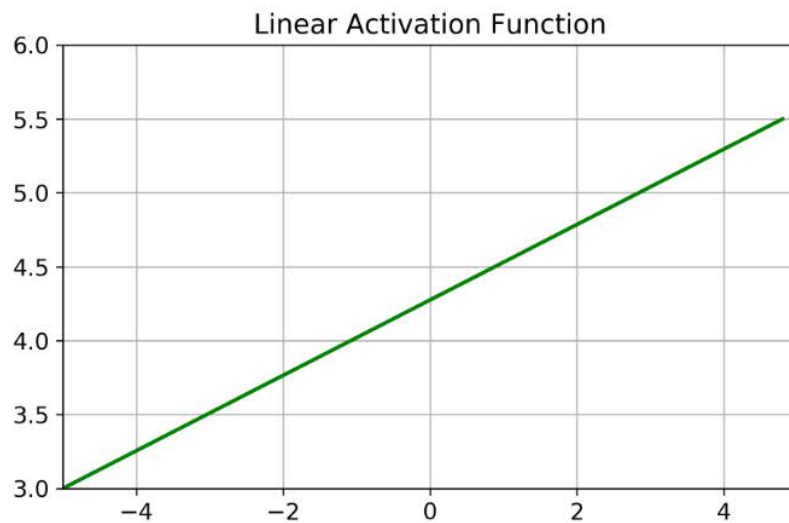
Βασικό κριτήριο για τη λειτουργία και την αποτελεσματικότητα του νευρωνικού δικτύου είναι η επιλογή της συνάρτησης ενεργοποίησης, που καθορίζει αν θα παραχθεί το αποτέλεσμα εξόδου για ένα συγκεκριμένο σύνολο επεξηγηματικών μεταβλητών  $x_i$ . Οι επιλογή της συνάρτησης ενεργοποίησης εξαρτάται κυρίως από το λόγο που χρησιμοποιούμε το νευρωνικό δίκτυο, δηλαδή το πρόβλημα που προσπαθούμε να λύσουμε.



Οι βασικές συναρτήσεις ενεργοποίησης, είναι η γραμμική συνάρτηση (linear), η βηματική συνάρτηση (threshold), η ανορθωτική γραμμική (rectifier), η λογιστική σιγμοειδής (sigmoid) και η υπερβολική εφαπτομένη (hyperbolic tangent).

- Γραμμική (linear) Συνάρτηση

Ξεκινάμε με την πιο απλή συνάρτηση ενεργοποίησης, τη γραμμική συνάρτηση  $y=x$ , η οποία για κάθε τιμή του αθροίσματος  $\sum w_i x_i$ , επιστρέφει το ίδιο το άθροισμα.



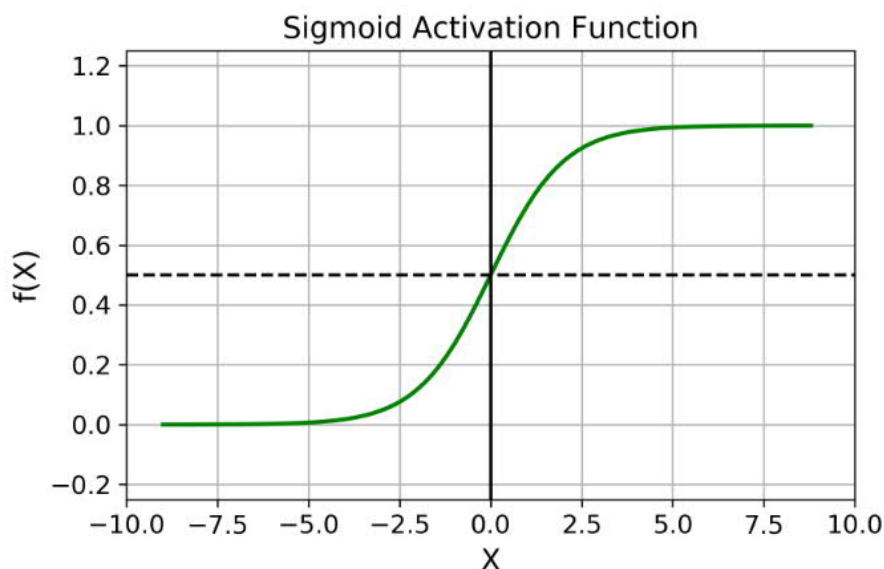
Συνήθως χρησιμοποιείται ως συνάρτηση ενεργοποίησης νευρωνικών δικτύων που χρησιμοποιούνται σε προβλήματα παλινδρόμησης, αφού σε τέτοιες περιπτώσεις, θέλουμε να έχουμε ως έξοδο του δικτύου (Output Layer), μια τιμή που να μην έχει όρια. Δηλαδή η πρόβλεψη του μοντέλου να μην περιορίζεται σε ένα σύνολο τιμών, κάτι που το πεδίο τιμών της γραμμικής συνάρτησης μπορεί να προσφέρει  $(-\infty, \infty)$ . Το βασικό πρόβλημα της γραμμικής συνάρτησης ενεργοποίησης είναι ότι η παράγωγος της είναι πάντα σταθερός αριθμός και ανεξάρτητη από τις τιμές εισόδου, δηλαδή την τιμή του  $X$ . Έτσι η μέθοδος απότομης καθόδου (gradient descent), που χρησιμοποιεί τις παραγώγους δεν λειτουργεί με τον τρόπο που θα έπρεπε καθώς η κλίση είναι πάντα σταθερή και ανεξάρτητη από το  $X$  και η εκπαίδευση του δικτύου γίνεται μέσω μίας σταθερής τιμής. Πηγαίνοντας σε πιο πολύπλοκα νευρωνικά δίκτυα που περιλαμβάνουν πολλά κρυφά στρώματα με πολλούς νευρώνες που συνδέονται μεταξύ τους, τότε αν χρησιμοποιούσαμε τη γραμμική συνάρτηση για την ενεργοποίηση κάθε στρώματος, τότε θα είχαμε μία γραμμική συνάρτηση της αρχικής τιμής να μεταφέρεται από στρώμα σε στρώμα και όλα τα κρυφά στρώματα θα μπορούσαν να αντικατασταθούν από ένα μοναδικό. Αυτό θα γινόταν γιατί ο συνδυασμός γραμμικών συναρτήσεων είναι γραμμική συνάρτηση και κατά συνέπεια η γραμμική συνάρτηση δε χρησιμοποιείται ως συνάρτηση ενεργοποίησης σε νευρωνικά δίκτυα με κρυφά στρώματα.

- Σιγμοειδής (sigmoid) Συνάρτηση

Η σιγμοειδής συνάρτηση είναι αρκετά διαφορετική από τις υπόλοιπες συναρτήσεις ενεργοποίησης και για κάθε βήμα δίνει την ένδειξη για το αν το

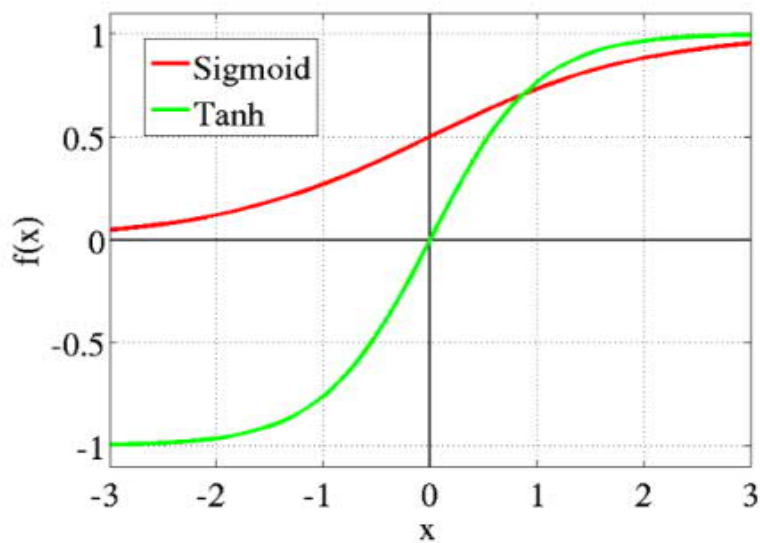
δίκτυο κινείται προς το στρώμα εξόδου ή όχι και για αυτό το λόγο είναι κατάλληλο για μεθόδους οπισθοδιάδοσης (backpropagation). Επίσης, λόγω της καμπυλότητας της είναι κατάλληλη για την περιγραφή πιθανοτήτων, αφού εκτός από τις τιμές 0, 1 παίρνει και όλες τις ενδιάμεσες τιμές. Σαν λειτουργία η σιγμοειδής συνάρτηση, για τιμές μεγαλύτερες του μηδενός συγκλίνει στον άσσο, ενώ για τιμές από το μηδέν και κάτω συγκλίνει στο μηδέν. Η σιγμοειδής χρησιμοποιείται συνήθως, στο στρώμα εξόδου ενός νευρωνικού δικτύου, σε πρόβλημα ταξινόμησης (classification) γιατί σε αυτές τις περιπτώσεις θέλουμε η έξοδος του νευρωνικού δικτύου να αντιστοιχεί στην πιθανότητα μια συγκεκριμένη εγγραφή να ανήκει σε κάθε μια από τις υπάρχουσες κλάσεις και η σιγμοειδής συνάρτηση λειτουργεί ακριβώς με αυτό τον τρόπο. Σήμερα όμως αυτή η συνάρτηση ενεργοποίησης έχει χάσει τη χρησιμότητά της στις περισσότερες περιπτώσεις, και αυτό γιατί η σιγμοειδής συνάρτηση από κάποιες τιμές του αθροίσματος  $\sum w_i x_i$  και μετά τείνει να επιστρέφει την ίδια σχεδόν τιμή, αυτό είναι ένα πρόβλημα γνωστό ως εξαφανιζόμενη κλίση (vanishing gradient).

Πρακτικά για πολύ υψηλές και πολύ χαμηλές τιμές του  $X$ , η  $f(X)$  δεν αλλάζει και ότι από ένα σημείο και μετά η κλίση (gradient) της συνάρτησης κόστους με βάση την οποία αξιολογείται η απόδοση του δικτύου σταθεροποιείται, δηλαδή το μοντέλο είναι πολύ δύσκολο να εκπαιδευτεί σε αυτά τα σημεία ή εκπαιδεύεται με πολύ αργό ρυθμό.



- Υπερβολική Εφαπτομένη (Tanh)

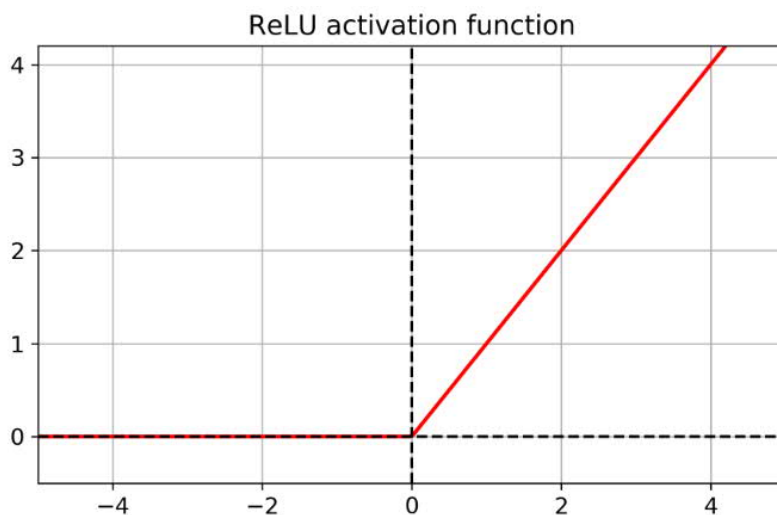
Η υπερβολική συνάρτηση ενεργοποίησης μοιάζει με την σιγμοειδή συνάρτηση ενεργοποίησης, αλλά σε μία καλύτερη μορφή. Το σύνολο τιμών της είναι το  $(-1, 1)$  και λόγω αυτού, είναι επίσης κατάλληλη στην έκφραση πιθανοτήτων. Υπερτερεί σε σχέση με την σιγμοειδή στο ότι είναι πιο αποτελεσματική στην



εκπαίδευση τιμών που είναι κοντά στο μηδέν, λόγω της αυξημένης κλίσης της σε αυτό το σημείο, σε σχέση με τη σιγμοειδή που θα έστειλε τις τιμές στο μηδέν. Χρησιμοποιείται κυρίως σε προβλήματα ταξινόμησης.

- Ανορθωτική (rectifier - ReLU) Συνάρτηση

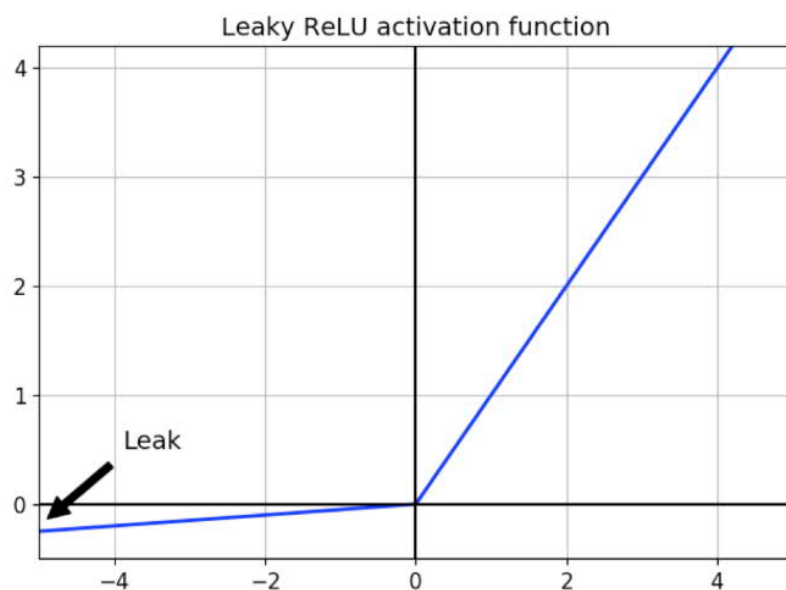
Η πιο διαδεδομένη συνάρτηση ενεργοποίησης σήμερα είναι η ανορθωτική



γραμμική συνάρτηση ReLU (Rectified Linear Unit) και αυτό γιατί δίνει καλύτερα αποτελέσματα και είναι υπολογιστικά γρήγορη, οπότε είναι ευρέως διαδεδομένη για νευρωνικά δίκτυα με πολλά κρυφά στρώματα. Η λειτουργία της είναι απλή, μηδενίζει το άθροισμα  $\sum_1^n w_i x_i$  όταν αυτό έχει τιμή μικρότερη του μηδενός και το επιστρέφει αυτούσιο όταν είναι μεγαλύτερο του μηδενός. Ο μηδενισμός για αρνητικές τιμές έχει ως βασικό πρόβλημα, ότι η εκπαίδευση του μοντέλου δεν γίνεται σωστά, καθώς δεν διαχειρίζεται σωστά και συνεπώς δεν μαθαίνει από

αυτές τις τιμές. Αυτό στη συνέχεια επηρεάζει τη μέθοδο απότομης καθόδου (Gradient Descent) αφού η κλίση παραμένει μηδενική και η μέθοδος μένει σε ένα τοπικό ελάχιστο και το επιθυμητό ολικό ελάχιστο δεν βρίσκεται ποτέ.

Η λύση σε αυτό το πρόβλημα, γνωστό και ως 'dying ReLU' δόθηκε μέσω μίας παραλλαγής της μεθόδου, που διαμορφώνει την παράμετρο της κλίσης της συνάρτησης ενεργοποίησης έτσι ώστε να έχει μία κλίση από τον οριζόντιο άξονα και να μην μηδενίζονται τα αθροίσματα των αρνητικών τιμών. Η



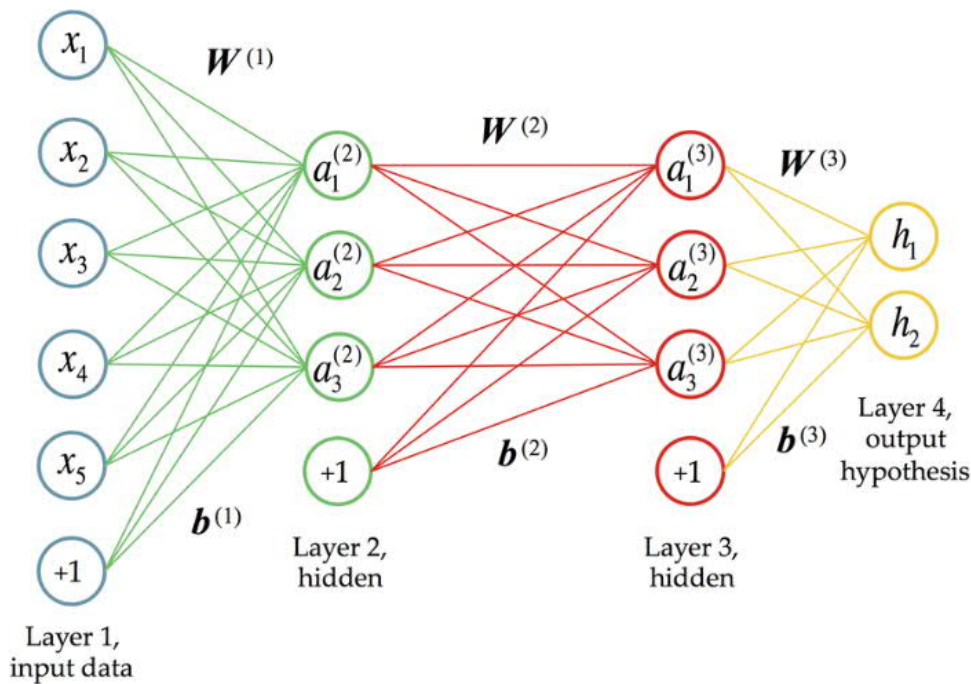
παραλλαγή αυτή για την τιμή της κλίσης να είναι  $\alpha = 0.01$  λέγεται Leaky ReLU, διαφορετικά λέγεται Randomized ReLU.

### 2.4.3 | Η Εκπαίδευση των Τεχνητών Νευρωνικών Δικτύων

Στην εκπαίδευση ενός νευρωνικού δικτύου πρωταρχικό ρόλο έχει η αρχιτεκτονική που θα επιλέξουμε για αυτό. Ένας μόνος τεχνητός νευρώνας δεν θα είναι αποτελεσματικός στον βαθμό που χρειάζεται για να επιλυθεί ένα πρόβλημα, όπως ακριβώς γίνεται και με τους νευρώνες στον ανθρώπινο οργανισμό, ο κάθε ένας είναι υπεύθυνος για να δεχτεί και να επεξεργαστεί διαφορετικά ερεθίσματα. Όπως προαναφέρθηκε ένα νευρωνικό δίκτυο αποτελείται από το στρώμα εισόδου, το στρώμα εξόδου και ανάμεσά τους υπάρχουν κρυφά στρώματα νευρώνων, των οποίων ο αριθμός εξαρτάται από το πρόβλημα που έχουμε να λύσουμε. Το πλήθος των νευρώνων στο στρώμα εισόδου είναι όσες και οι επεξηγηματικές μεταβλητές του μοντέλου που έχουμε επιλέξει, ενώ όσον αφορά το στρώμα εξόδου παράγει νευρώνα ή νευρώνες ανάλογα με το είδος του νευρωνικού δικτύου. Εάν χρησιμοποιείται για



πρόβλεψη τιμής σε ένα πρόβλημα παλινδρόμησης, τότε το στρώμα εξόδου του νευρωνικού δικτύου θα παράγει την τιμή της πρόβλεψης για μία τιμή απόκρισης, ενώ αν χρησιμοποιείται σε ένα πρόβλημα ταξινόμησης, τότε το στρώμα εξόδου θα έχει τόσους νευρώνες όσες οι κλάσεις που θέλουμε να ταξινομήσουμε τις παρατηρήσεις.



Όπως βλέπουμε στην παραπάνω εικόνα, οι νευρώνες του κάθε στρώματος συνδέονται με ακμές και η κάθε ακμή έχει ένα βάρος. Σε γενικές γραμμές, σε ότι αφορά την εκπαίδευση του νευρωνικού δικτύου γίνεται με το να δώσουμε μία ομάδα από πρότυπα στο δίκτυο, αντιπροσωπευτικά ή παρόμοια με αυτά που θέλουμε να προβλέπει το δίκτυο. Στην ουσία δίνουμε στο νευρωνικό δίκτυο ως εισόδους κάποια πρότυπα δεδομένα για τα οποία ξέρουμε ποια πρέπει να είναι η έξοδος στο δίκτυο, ποιά δηλαδή είναι η επιθυμητή πρόβλεψη που περιμένουμε από το δίκτυο να δώσει. Ουσιαστικά του δίνουμε και την ερώτηση και την απάντηση που αντιστοιχεί σε αυτή. Αυτή η ομάδα από πρότυπα δεδομένα λέγονται σύνολο εκπαίδευσης και είναι της ίδιας μορφής με τα δεδομένα με τα οποία θα πρέπει στην συνέχεια το νευρωνικό δίκτυο να προβλέπει και με τα δεδομένα αυτά τροποποιεί την εσωτερική του δομή. Έτσι, αφού βρει την σωστή εσωτερική δομή, θα μπορεί ανάλογα να λύνει και άλλα προβλήματα της ίδιας μορφής στα οποία δεν έχει εκπαιδευτεί προηγουμένως, δηλ. δεν έχει εκπαιδευτεί στα πρότυπα τους.

Πώς ακριβώς εκπαιδεύεται το νευρωνικό δίκτυο με κρυφά στρώματα;

Θα επικεντρωθούμε στην εκπαίδευση των νευρωνικών δικτύων που σκοπό έχουν την επίλυση ενός προβλήματος παλινδρόμησης. Βασικός πυλώνας της εκπαίδευσης ενός νευρωνικού δικτύου είναι η επίτευξη της τέλει πρόβλεψης

της ζητούμενης μεταβλητής. Με άλλα λόγια η ελαχιστοποίηση μιας συνάρτησης κόστους που υπολογίζει το σφάλμα της πρόβλεψης δοθείσης της πραγματικής τιμής. Όσον αφορά τα προβλήματα παλινδρόμησης, μία από τις πιο συχνά χρησιμοποιούμενες συναρτήσεις κόστους είναι το μέσο τετραγωνικό σφάλμα,  $MSE (C = \frac{1}{n} \sum_1^n (\hat{y}_i - y_i)^2)$ .

Ξεκινώντας, το πρώτο βήμα είναι η είσοδος των παρατηρήσεων του συνόλου εκπαίδευσης στο νευρωνικό δίκτυο, από το στρώμα εισόδου και η επιβάρυνση τους με τα αρχικοποιημένα βάρη. Το πλήθος των παρατηρήσεων που δέχεται το νευρωνικό δίκτυο τις περισσότερες φορές δίνεται ως παράμετρος και μπορεί να καθοριστεί από το χρήστη. Στη συνέχεια, στους νευρώνες του πρώτου κρυφού στρώματος ή, αν το νευρωνικό δεν έχει κρυφά στρώματα, στο ενδιάμεσο στρώμα, υπολογίζονται το βεβαρημένο άθροισμα και η τιμή της συνάρτησης ενεργοποίησης που έχουμε επιλέξει να χρησιμοποιήσουμε και ανάλογα με την υπολογισμένη τιμή αυτή θα ενεργοποιηθεί ή όχι ο νευρώνας. Αυτή η διαδικασία επαναλαμβάνεται μέχρις ότου φτάσουμε στο τελευταίο κρυφό στρώμα του νευρωνικού δικτύου και πάντοτε το επόμενο κρυφό στρώμα δέχεται ως τιμές εισόδου, τις υπολογισμένες τιμές του προηγούμενου κρυφού στρώματος. Αφού φτάσουμε στο τελευταίο κρυφό στρώμα, οι υπολογισμένες τιμές αυτού, οδηγούνται στο στρώμα εξόδου και εν τέλει δίνεται η ζητούμενη πρόβλεψη της μεταβλητής  $y$ ,  $\hat{y}$ , για τις παρατηρήσεις. Όσο το πλήθος των παρατηρήσεων που εισήλθαν στο νευρωνικό δίκτυο, τόσες και προβλέψεις που θα εξέλθουν από αυτό. Τότε έρχεται η σειρά της συνάρτησης κόστους, η οποία υπολογίζεται για κάθε τιμή  $\hat{y}$  που παράχθηκε για τις παρατηρήσεις που εισήλθαν, και στη συνέχεια ανανεώνονται τα βάρη, έτσι ώστε η νέα τιμή της συνάρτησης κόστους να είναι μικρότερη. Στόχος είναι η τιμή της συνάρτησης κόστους να φτάσει ιδανικά στο μηδέν ή πιο ρεαλιστικά να μην βελτιώνεται άλλο. Για την βέλτιστη τιμή αυτή, έχουμε και τα βέλτιστα βάρη για κάθε ακμή του νευρωνικού δικτύου. Ελάχιστος χρόνος εκπαίδευσης ενός νευρωνικού δικτύου θεωρείται η μία εποχή (epoch), η οποία ολοκληρώνεται όταν όλες οι παρατηρήσεις του συνόλου εκπαίδευσης των δεδομένων, εισέλθουν στο νευρωνικό δίκτυο.

## Βήματα Λειτουργίας Νευρωνικού Δικτύου

Βήμα 1: Αρχικοποίηση των βαρών των ακμών με τυχαίο τρόπο χρησιμοποιώντας μικρές τιμές κοντά στο 0

Βήμα 2: Εισαγωγή του δοθέντος πλήθους των παρατηρήσεων στους νευρώνες του στρώματος εισόδου, με την κάθε επεξηγηματική μεταβλητή να τοποθετείται στον κατάλληλο νευρώνα

Βήμα 3: Από το στρώμα εισόδου προς το στρώμα εξόδου (Forward Propagation) ενεργοποιούνται οι νευρώνες και παράγεται η πρόβλεψη του νευρωνικού δικτύου από αυτές τις παρατηρήσεις

Βήμα 4: Σύγκριση της πρόβλεψη του νευρωνικού δικτύου  $\hat{y}$  με την αντίστοιχη πραγματική της τιμή  $y$ , για τις παρατηρήσεις αυτές και υπολογισμός του σφάλματος (τιμή της συνάρτησης κόστους)

Βήμα 5: Από το στρώμα εισόδου προς το στρώμα εξόδου (Back Propagation) ενημέρωση των βαρών, σύμφωνα με την τιμή της συνάρτησης κόστους

Βήμα 6: Επανάληψη βημάτων 1-5 για κάθε πλήθος παρατηρήσεων μέχρι ότου να περάσουν όλα τα δεδομένα εκπαίδευσης από το νευρωνικό δίκτυο και να ολοκληρωθεί μία εποχή

Βήμα 7: Επανάληψη διαδικασίας για όσες εποχές επιλέξουμε

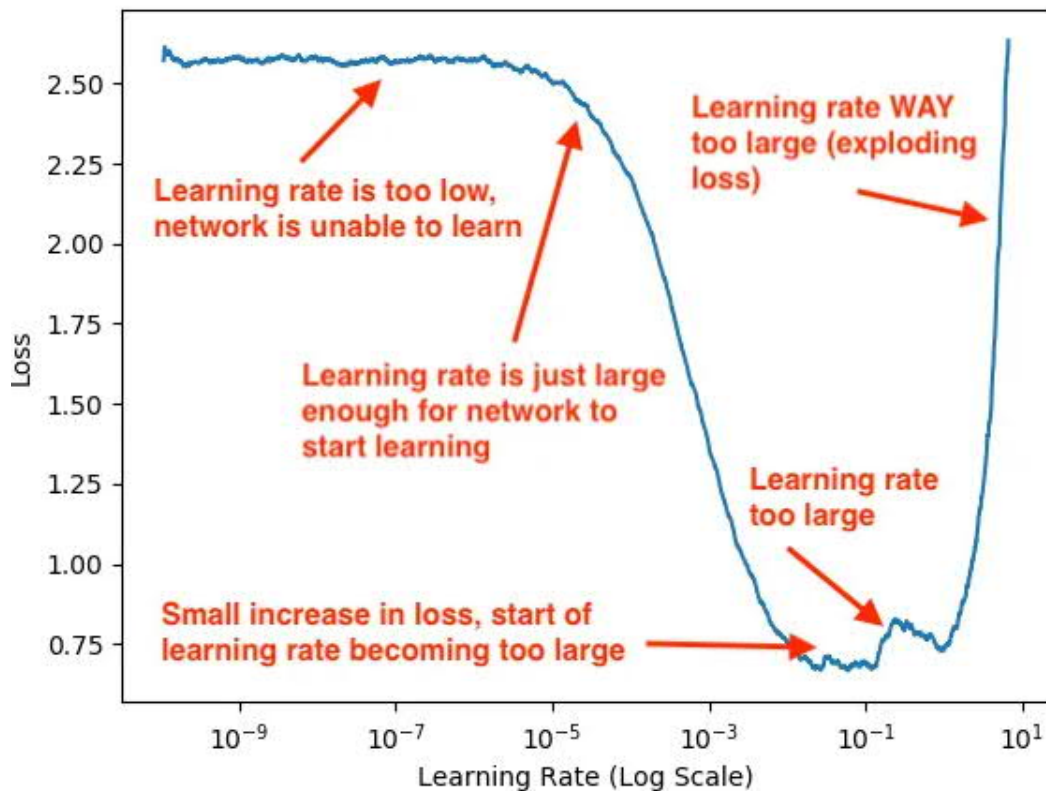
### Διαδικασία Ενημέρωσης Βαρών

Με την εύρεση των κατάλληλων βαρών για τις συνάψεις του νευρωνικού δικτύου, πετυχαίνεται η ελαχιστοποίηση της συνάρτησης κόστους και κατά συνέπεια η εύρεση της λύσης για τις αντίστοιχες τιμές εισαγωγής. Η πιο γνωστή και ευρέως χρησιμοποιούμενη μέθοδος για το σκοπό αυτό είναι η προαναφερθείσα μέθοδος της απότομης καθόδου (Gradient Descent). Έχουμε ήδη αναφερθεί στη λειτουργία της στο κεφάλαιο [2.3.1](#) και τώρα θα αναφερθούμε στο πως εφαρμόζεται στην εκμάθηση βαρών σε νευρωνικά δίκτυα.

Η μέθοδος απότομης καθόδου υπολογίζει πρώτα τη παράγωγο της συνάρτησης κόστους ως προς κάθε βάρος και το πολλαπλασιάζει με τον ρυθμό εκμάθησης, έναν σταθερό αριθμό που επιλέγεται από τον χρήστη του δικτύου.

Ο ρυθμός εκμάθησης καθορίζει το πόσο πολύ ή λίγο θα διορθωθούν τα βάρη, δηλαδή τον ρυθμό που αυτά διορθώνονται. Εάν επιλεγεί μία πολύ μικρή τιμή ως ρυθμός εκπαίδευσης, το νευρωνικό θα εκπαιδευτεί πολύ αργά - ή δεν θα μπορεί να εκπαιδευτεί καθόλου - και η σύγκλιση στην ελάχιστη τιμή της συνάρτησης κόστους θα παίρνει πολύ χρόνο και συνεπώς θα κοστίζει πολύ.

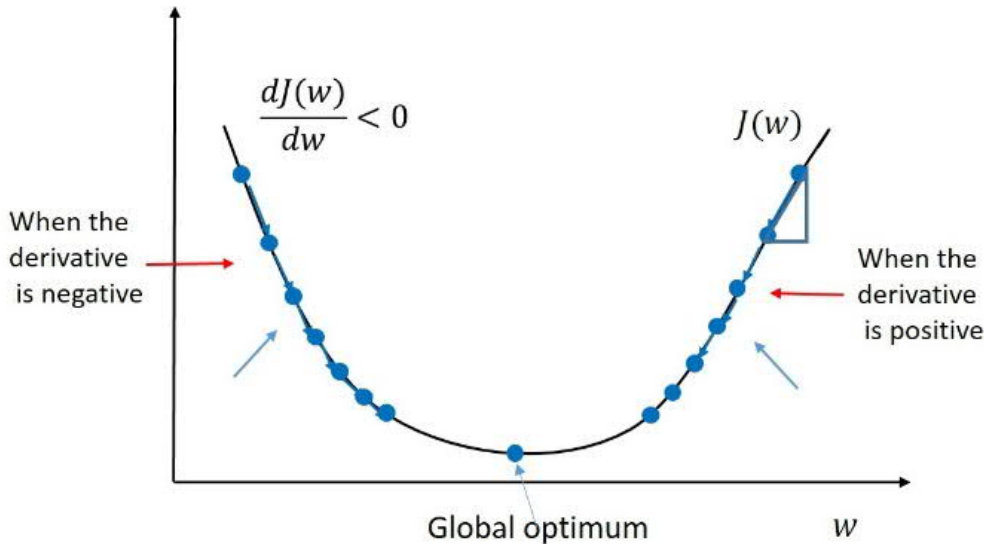
Αντίθετα, η επιλογή μιας μεγάλης τιμής ρυθμού εκμάθησης, μπορεί να οδηγήσει το νευρωνικό στο να προσπερνά την ελάχιστη τιμή και συνεπώς να αποκλίνει.



Πηγή : bitprime.co

Στο παραπάνω γράφημα βλέπουμε τη μεταβολή της συνάρτησης κόστους την οποία θέλουμε να ελαχιστοποιήσουμε σε σχέση με την τιμή του ρυθμού εκμάθησης που έχουμε επιλέξει. Βλέπουμε πως αν επιλεχθεί μία τιμή κοντά στο μηδέν για ρυθμός εκμάθησης, τότε το νευρωνικό δεν θα μπορεί να εκπαιδευτεί, ενώ μία μεγάλη τιμή δεν θα επέτρεπε στο δίκτυο να βρει ποτέ την ελάχιστη τιμή της συνάρτησης.

Η ενημέρωση βαρών από την μέθοδο απότομης καθόδου γίνεται μέσω της εξίσωσης 2.4.1.



Γνωρίζοντας ότι για τιμές μικρότερες της βέλτιστης τιμής της συνάρτησης, η τιμή της θα είναι θετική και η παράγωγος αρνητική, τότε η αύξηση του βάρους, θα μειώσει και τη συνάρτηση κόστους και θα μας οδηγήσει πιο κοντά στη βέλτιστη τιμή. Αντίστοιχα, για τιμές μεγαλύτερες της βέλτιστης τιμής, η συνάρτηση κόστους θα έχει πάλι θετική τιμή αλλά με θετική παράγωγο, συνεπώς αν αυξήσουμε τα βάρη θα αυξήσουμε και την συνάρτηση κόστους, σε αυτή τη περίπτωση μειώνουμε τα αντίστοιχα βάρη και οδηγούμαστε πιο κοντά στη βέλτιστη τιμή. Η διαδικασία ιδανικά ολοκληρώνεται όταν η τιμή της συνάρτησης κόστους είναι μηδέν, όπου έχουμε βρει τις βέλτιστες τιμές για τα βάρη του μοντέλου μας. Αξίζει να σημειωθεί εδώ ότι τα βάρη διορθώνονται ανάλογα με το πόσο υπεύθυνα είναι για το παραγόμενο σφάλμα και ότι κατά τη διαδικασία ενημέρωσης των βαρών, είναι πιθανό να μηδενιστούν κάποια. Συνεπώς στο τελικό μοντέλο δεν είναι απαραίτητο να υπάρχουν βάρη μεταξύ όλων των νευρώνων του ενός στρώματος, με το επόμενο στρώμα. Πολλές φορές, το μοντέλο αναγνωρίζει ποιες συνάψεις είναι σημαντικές για την εκπαίδευσή του αλλά και πιθανά μοτίβα που δημιουργούνται από τα δεδομένα και μηδενίζει συνάψεις μεταξύ νευρώνων που θεωρεί ότι δεν είναι απαραίτητες και δεν προσφέρουν κάτι στην εκπαίδευσή του.

## 2.4.4 | Υπερεκπαίδευση των Τεχνητών Νευρωνικών Δικτύων (Overfitting)

*Προκειμένου να γενικευτεί καλά, ένα σύστημα πρέπει να είναι αρκετά ισχυρό ώστε να προσεγγίζει το στόχο. Εάν είναι τόσο απλό που δεν μπορεί να προσαρμοστεί ούτε στα δεδομένα εκπαίδευσης του, τότε η γενίκευση σε νέα δεδομένα θα είναι πιθανώς και αυτή κακή. [...] Ένα υπερβολικά περίπλοκο σύστημα, ωστόσο, μπορεί να είναι σε θέση να προσεγγίσει τα δεδομένα με πολλούς διαφορετικούς τρόπους που δίνουν παρόμοια σφάλματα και είναι απίθανο να επιλέξει εκείνο που θα μπορεί να γενικευτεί...*

- Page 241, Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks, 1999

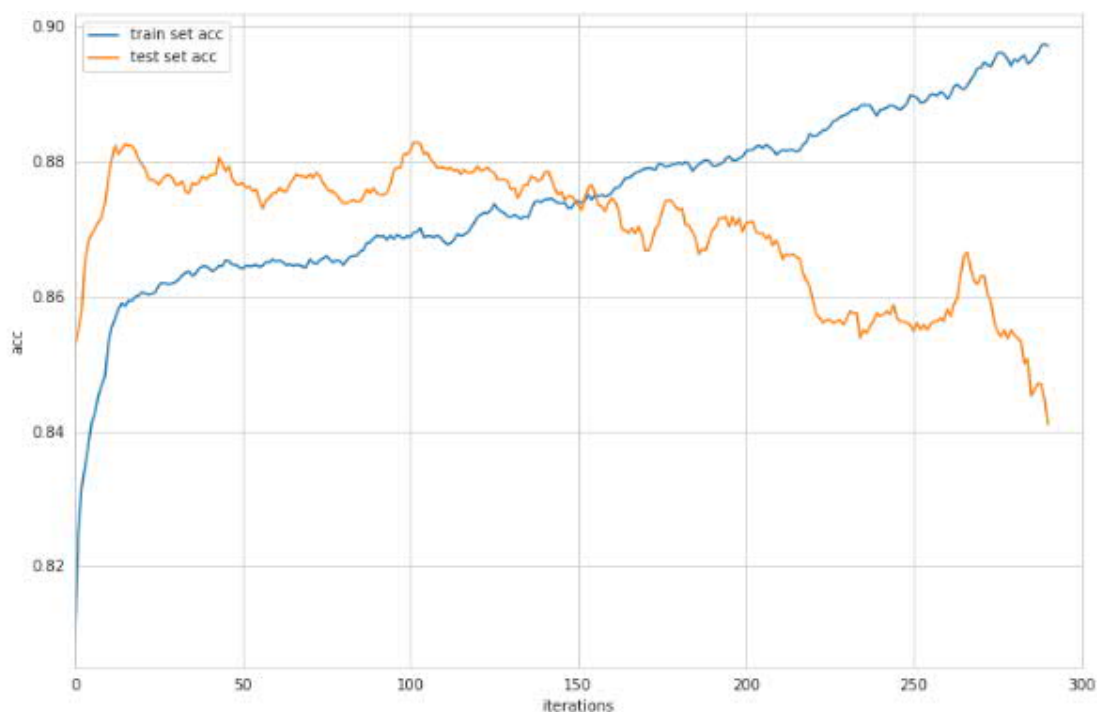
Ο στόχος των νευρωνικών δικτύων, είναι η αυτόματη πρόβλεψη μεταβλητών από ένα σύνολο εισαγόμενων δεδομένων, αφού πρώτα έχουν εκπαιδευτεί μέσω ενός υποσυνόλου αυτών. Η αληθινή πρόκληση των νευρωνικών δικτύων, και το σημείο από όπου αξιολογούνται στην πραγματικότητα, είναι η δυνατότητα λειτουργίας τους σε διαφορετικό σύνολο δεδομένων ίδιας μορφής με αυτά που εκπαιδεύτηκαν. Με αυτό τον τρόπο, ελέγχεται η εκπαίδευση του νευρωνικού και το αν έχει αποφύγει ένα από τα πιο επικίνδυνα τεχνικά προβλήματα στον τομέα της τεχνητής νοημοσύνης, την υπερεκπαίδευση.

Ως υπερεκπαίδευση ορίζεται η κατάσταση όπου η απόδοση του νευρωνικού δικτύου είναι με μεγάλη διαφορά καλύτερη όταν εφαρμόζεται στα δεδομένα με τα οποία εκπαιδεύτηκε (training set) από ότι όταν εφαρμόζεται στα δεδομένα επαλήθευσης (test set). Δεν υπάρχει συγκεκριμένος τρόπος με τον οποίο ένα νευρωνικό δίκτυο υπερεκπαίδευεται. Είναι μία κατάσταση στην οποία οδηγείται μετά από μη στοχευμένη και μελετημένη επιλογή υπερπαραμέτρων, από τον χρήστη. Οι υπερπαραμέτροι είναι απαραίτητες για την κατασκευή των νευρωνικών δικτύων, ποικίλουν αναλόγως με τη μέθοδο και το είδος του νευρωνικού που θέλουμε να κατασκευάσουμε και είναι στην ευχέρεια του χρήστη να τις προσδιορίσει με όσο το δυνατόν περισσότερη σύνεση και μελέτη. Η πιο απλή μέθοδος για την αποφυγή της υπερεκπαίδευσης είναι η σάρωση του φασικού χώρου των παραμέτρων και η σύγκριση κάποιων χαρακτηριστικών μετρικών για κάθε συνδυασμό αυτών, με σκοπό την εξασφάλιση της γενίκευσης τους. Πρέπει στο μυαλό του σχεδιαστή ενός νευρωνικού δικτύου να υπάρχει πάντα η σκέψη ότι η εξασφάλιση της γενίκευσης του μοντέλου, είναι πολύ πιο σημαντική από μία φαινομενική τέλεια απόδοση.

Θα αναφερθούμε σε μερικές από τις πιο βασικές υπερπαραμέτρους που οδηγούν στην υπερεκπαίδευση ενός νευρωνικού δικτύου.

Πρωταρχικό ρόλο έχει ο ρυθμός εκμάθησης που έχει επιλέξει ο σχεδιαστής του μοντέλου καθώς, μία μικρή τιμή ρυθμού εκμάθησης μπορεί να οδηγήσει το μοντέλο σε ένα τοπικό ελάχιστο του υποσυνόλου δεδομένων εκπαίδευσης, από το οποίο δεν μπορεί να ξεφύγει. Ιδανικά, πρέπει να επιλεχθεί μία τιμή αρκετά μεγάλη έτσι ώστε να αποφεύγονται τα τοπικά ελάχιστα αλλά και αρκετά μικρή ώστε να μπορεί να βρεθεί το ελάχιστο χωρίς να το προσπερνά το μοντέλο λόγω μεγάλων βημάτων.

Εξίσου σημαντική υπερπαραμέτρος, καθοριστική για την υπερεκπαίδευση του μοντέλου είναι το πλήθος των παρατηρήσεων που δέχεται το μοντέλο στο στρώμα εισόδου (batch). Εάν επιλεχθεί μεγάλο batch τότε είναι πολύ δύσκολο να πετύχουμε τη γενίκευση του μοντέλου. Από την άλλη, μικρό μέγεθος batch συνήθως οδηγεί σε γρηγορότερη σύγκλιση σε σωστές - καλές λύσεις, καθώς τότε το μοντέλο ξεκινά να μαθαίνει, χωρίς να έχει δει όλα τα δεδομένα, αλλά το μικρό μέρος από αυτά που έχουν δοθεί από τα πρώτα batch δεδομένων. Παρόλα αυτά το μειονέκτημα της επιλογής μικρού μεγέθους batch είναι ότι το μοντέλο δεν είναι σίγουρο ότι θα καταφέρει να συγκλίνει στο ολικό βέλτιστο σημείο. Η επιλογή του μεγέθους batch πρέπει να γίνεται μέσω δοκιμών σε συνδυασμό με το ρυθμό εκπαίδευσης και των μετρικών που έχουμε επιλέξει για την αξιολόγηση του μοντέλου μας.



Οι εποχές που θα επιλέξουμε για την εκπαίδευση του νευρωνικού δικτύου, δηλαδή οι φορές που όλα τα δεδομένα εκπαίδευσης θα εισέλθουν στους νευρώνες, είναι στενά συνδεδεμένες με το φαινόμενο της υπερεκπαίδευσης

ενός μοντέλου. Η επιλογή πολλών εποχών οδηγεί στο νευρωνικό δίκτυο να συνεχίζει να εκπαιδεύεται ακόμα και όταν έχει φτάσει σε μία ιδανική λύση που πετυχαίνει τη ζητούμενη γενίκευση που θέλουμε και έτσι υπερεκπαιδεύεται και η γενίκευση του μοντέλου παύει να ικανοποιείται. Στο επόμενο παράδειγμα, βλέπουμε τη γραφική παράσταση της απόδοσης του μοντέλου, σύμφωνα με τις μετρικές που έχουν επιλεγεί, συναρτήσει των εποχών εκπαίδευσης. Ζητάμε τον αριθμό εποχών όπου η απόδοση του μοντέλου είναι η μέγιστη και ταυτόχρονα το μοντέλο αποδίδει με σχεδόν τον ίδιο τρόπο στα δεδομένα εκπαίδευσης και επαλήθευσης. Στο συγκεκριμένο παράδειγμα οι βέλτιστες εποχές φαίνεται να είναι της τάξης των 150.

Για την επίτευξη αυτού του αποτελέσματος χρησιμοποιείται η τεχνική του πρόωρου σταματημού της εκπαίδευσης (early stopping), όπου βρίσκεται ο βέλτιστος αριθμός εποχών προτού το μοντέλο αρχίσει να υπερεκπαιδεύεται.

#### 2.4.5 | **Σχεδιασμός Νευρωνικού Δικτύου και Αποτελέσματα**

Ο σχεδιασμός ενός νευρωνικού δικτύου και η επιλογή της κατάλληλης αρχιτεκτονικής δεν έχει κάποιο συγκεκριμένο τρόπο που γίνεται ή κάποιους κανόνες που πρέπει να ακολουθούνται, παρόλα αυτά έχουν άμεση σχέση με το πρόβλημα για το οποίο σχεδιάζεται. Πρωταρχικό ρόλο στον σχεδιασμό, έχει το πλήθος των δεδομένων που έχουμε στη διάθεσή μας για την εκπαίδευση του νευρωνικού. Εάν τα δεδομένα που έχουμε στη διάθεσή μας για την επίλυση ενός προβλήματος είναι λίγα, τότε ο σχεδιασμός του δικτύου θα πρέπει να είναι όσο πιο απλός γίνεται, δηλαδή με λίγους νευρώνες και λίγα κρυφά στρώματα, το αντίθετο θα οδηγούσε σε υπερεκπαίδευση του μοντέλου. Για το σωστό χτίσιμο της αρχιτεκτονικής του μοντέλου, καλό είναι να ξεκινάμε πάντα με την πιο απλή, χωρίς κρυφά στρώματα και σιγά σιγά να προσθέτουμε για να έχουμε πάντα τον έλεγχο των μετρικών μας και να μην ξεφεύγουμε από τη διατήρηση της γενίκευσης του μοντέλου. Ως απλό νευρωνικό δίκτυο θεωρούμε αυτό το οποίο έχει μέχρι τρία κρυφά στρώματα, αυξάνοντας τον αριθμό των κρυφών στρωμάτων αναφερόμαστε σε βαθιά νευρωνικά δίκτυα (deep learning) με τα οποία δεν θα ασχοληθούμε στη συγκεκριμένη διπλωματική.

Αφού έχει σχεδιαστεί το μοντέλο, έχουμε καταλήξει στην κατάλληλη αρχιτεκτονική και έχουν ολοκληρωθεί οι επιλεγμένες εποχές εκπαίδευσης, έχουμε τα αποτελέσματα της πρόβλεψης που έκανε το νευρωνικό δίκτυο. Για την αξιολόγηση των αποτελεσμάτων, εκτός από τις μετρικές που έχουμε επιλέξει να ακολουθούμε και τη διατήρηση της γενίκευσης του, καλό είναι να κατανοούμε και να μεταφράζουμε τον τρόπο με τον οποίο το νευρωνικό δίκτυο κατέληξε σε αυτά τα αποτελέσματα. Σε αυτό, βοηθάει η οπτικοποίηση των



αποτελεσμάτων, έτσι ώστε να δούμε στο τελικό στάδιο ποιες επεξηγηματικές μεταβλητές συνδέονται μεταξύ τους, ποιες έπαιξαν σημαντικό ρόλο στο αποτέλεσμα και τον τρόπο με τον οποίο έχουν διαμορφωθεί τα βάρη στο μοντέλο μας.

Είναι πολύ πιθανό να παρατηρήσουμε νευρώνες του στρώματος εισόδου να μην συνδέονται με όλους τους νευρώνες του επόμενου στρώματος, δηλαδή μετά την διαδικασία ανανέωσης βαρών το νευρωνικό να αποφάσισε να μηδενίσει κάποια από αυτά τα βάρη. Έτσι, είναι πιθανό να μην συνδέονται όλες οι επεξηγηματικές μεταβλητές μεταξύ τους και με τους νευρώνες του επόμενου στρώματος. Εάν για παράδειγμα ένας νευρώνας δέχεται συνάψεις από δύο επεξηγηματικές μεταβλητές, τότε έχει βρει κάποια σχέση μεταξύ αυτών που έχει σημασία και βαρύτητα για το αποτέλεσμα της μεταβλητής πρόβλεψης και έτσι τις επεξεργάζεται ξεχωριστά σε κάθε δεδομένο εισόδου. Στη συνέχεια σύμφωνα με τις τιμές αυτών και τα βάρη τους στη συνάρτηση ενεργοποίησης, θα παράγει τη πρόβλεψη του ο συγκεκριμένος νευρώνας. Έτσι, η τελική πρόβλεψη για το batch εισόδου του νευρωνικού είναι αποτέλεσμα συνδυασμού της πρόβλεψης του κάθε νευρώνα, όπως γίνεται και με την επεξεργασία της πληροφορίας στον ανθρώπινο εγκέφαλο, συνδυάζοντας τις αισθήσεις και καταλήγοντας στο αποτέλεσμα σε επεξεργασία όλων αυτών. Αυτή είναι και η δύναμη των νευρωνικών δικτύων.

### 3 | **Επιλογή Μεταβλητών Για Την Κατασκευή Μοντέλου Πρόβλεψης**

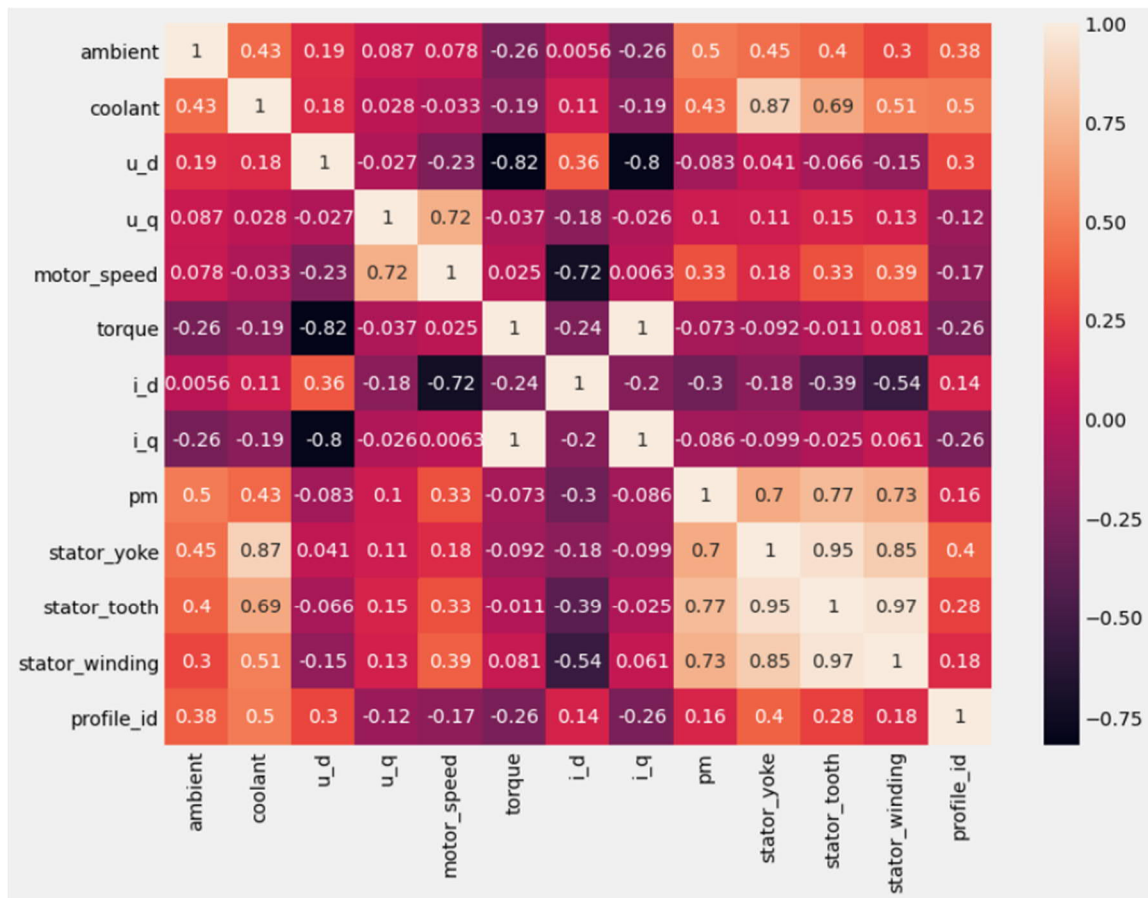
Το σύνολο δεδομένων που επεξεργαστήκαμε για τη προσπάθεια πρόβλεψης της θερμοκρασίας του ρότορα μιας Σύγχρονης Μηχανής Μόνιμου Μαγνήτη (ΣΜΜΜ), περιέχει 12 μεταβλητές (εξαιρουμένης της μεταβλητής απόκρισης) και καλούμαστε να επιλέξουμε ποιες μεταβλητές θα χρησιμοποιήσουμε για την κατασκευή του μοντέλου παλινδρόμησης. Η επιλογή μεταβλητών είναι το πρώτο και σημαντικότερο βήμα στην κατασκευή ενός μοντέλου, καθώς καθορίζει την ποιότητα πρόβλεψης ανεξαρτήτου της μεθόδου που θα επιλεγεί την για εκπαίδευση του. Στον πραγματικό κόσμο η πληροφορία βρίσκεται παντού αλλά το νόημα είναι στη σωστή επεξεργασία της. Έτσι και στην κατασκευή ενός μοντέλου είναι βασικό να επιλέξουμε τόσες μεταβλητές όσες μας δίνουν την απαραίτητη πληροφορία για να προβλέψουμε τη μεταβλητή απόκρισης. Πριν χρησιμοποιήσουμε όλες τις διαθέσιμες μεταβλητές πρέπει αρχικά να αξιολογήσουμε τη πληροφορία που θα μας προσφέρουν σε σχέση με αυτό που θέλουμε να πετύχουμε και οπωσδήποτε να έχουμε στο μυαλό μας ότι στον πραγματικό κόσμο η χρήση πληροφορίας μεταφράζεται σε χρήματα,

καθώς για να συλλέξουμε πληροφορία αλλά και να εκπαιδεύσουμε ένα μοντέλο με μεγάλο όγκο δεδομένων, χρειάζεται χρόνο και χρήματα.

Ένα πρώτο βήμα που γίνεται είναι η αφαίρεση επεξηγηματικών μεταβλητών που παρέχουν παρόμοια πληροφορία και δεν προσφέρουν κάτι ουσιαστικό στο μοντέλο μας. Από αυτές τις μεταβλητές κρατάμε εκείνη με τη μεγαλύτερη συσχέτιση με την μεταβλητή απόκρισης, έτσι απαλείφεται η διπλή πληροφορία και μειώνονται οι απαιτούμενες μεταβλητές για την κατασκευή του μοντέλου. Εκτός από μείωση μεγέθους δεδομένων και συνεπώς κόστους, η σωστή επιλογή μεταβλητών βελτιώνει γενικώς το μοντέλο, μειώνει το φαινόμενο την υπερπροσαρμογής, αφού η εκπαίδευση γίνεται πιο απλή χωρίς θόρυβο και περιττή πληροφορία και μειώνεται ο χρόνος εκπαίδευσης αφού ο αλγόριθμος επεξεργάζεται λιγότερα δεδομένα για την εκπαίδευσή του.

Για την επιλογή μεταβλητών συμβουλευόμαστε τον πίνακα συσχέτισης (correlation matrix) που απεικονίζει τις συσχετίσεις όλων των μεταβλητών του συνόλου δεδομένων και τα διαγράμματα διασποράς (scatterplots) μεταξύ της μεταβλητής απόκρισης και των επεξηγηματικών μεταβλητών. Από εκεί εντοπίζουμε τις μεταβλητές που έχουν μεγαλύτερη συσχέτιση με τη μεταβλητή απόκρισης και μικρότερη συσχέτιση με τις υπόλοιπες επεξηγηματικές μεταβλητές και καταλήγουμε στις κατάλληλες μεταβλητές για το μοντέλο μας. Μία άλλη τεχνική είναι η φασική εξερεύνηση των μεταβλητών, όπου αφαιρούμε μεταβλητές και συγκρίνουμε την απόδοση του μοντέλου και καταλήγουμε στον πιο πετυχημένο συνδυασμό. Αυτό δεν είναι υλοποιήσιμο όταν υπάρχουν πολλές διαθέσιμες μεταβλητές και όταν υπάρχουν μεγάλες συσχετίσεις μεταξύ τους, οι οποίες δεν θα βρεθούν με αυτή τη τεχνική.

### 3.1 | Σχέση Μεταβλητών

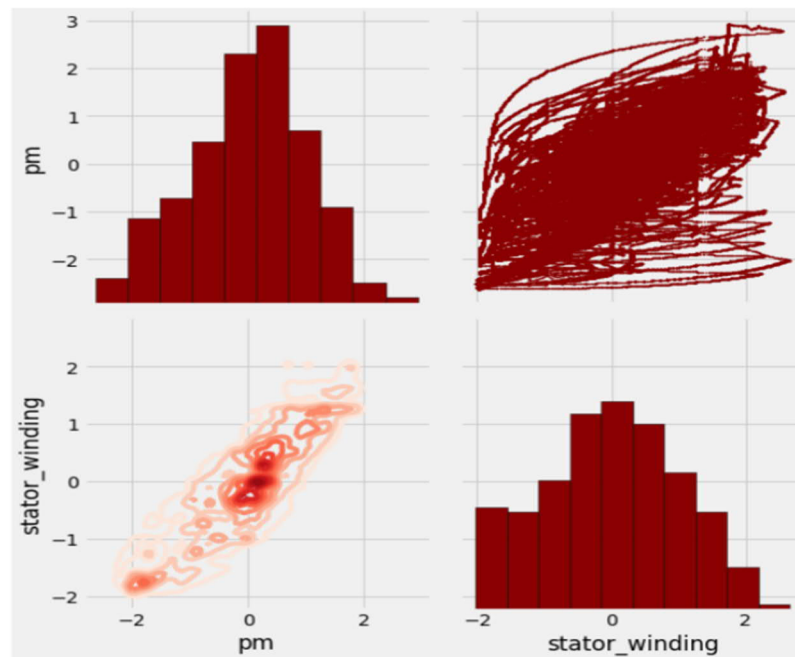


Για το πρόβλημά μας συμβουλευόμαστε τον πίνακα συσχετίσεων των μεταβλητών, έτσι επιβεβαιώνουμε το αναμενόμενο από τον ορισμό των μεταβλητών, ότι οι 3 μεταβλητές που αφορούν τις θερμοκρασίες του στάτη έχουν απόλυτη συσχέτιση μεταξύ τους αλλά και με τη θερμοκρασία του ρότορα, κάτι απόλυτα λογικό, εφόσον βρίσκονται σχεδόν δίπλα σε μία ΣΜΜΜ. Για να επιβεβαιωθεί αυτή η σχέση, παραθέτουμε και το διάγραμμα διασποράς μεταξύ της μεταβλητής απόκρισης και μία εκ των τριών θερμοκρασιών του στάτη και η γραμμική συσχέτιση μεταξύ τους φαίνεται ξεκάθαρα. Όπως είναι προφανές δεν θα χρησιμοποιηθούν στο μοντέλο καθώς αν κάποιος έχει στη διάθεση του αυτή την πληροφορία δεν θα ενδιαφέρεται για την κατασκευή μοντέλου που προβλέπει τη θερμοκρασία του ρότορα γιατί μπορεί να βρεθεί και με απλή γραμμική παλινδρόμηση. Η μεταβλητή torque δεν θα χρησιμοποιηθεί στο μοντέλο καθώς όπως αναφέρεται στο σύνολο δεδομένων, δεν είναι εύκολο να μετρηθεί και στην πραγματικότητα δεν είναι μια πληροφορία που θεωρείται δεδομένη. Τέλος, δεν θα χρησιμοποιηθεί η μεταβλητή profile\_id γιατί είναι μια κατηγορική μεταβλητή που για να χρησιμοποιηθεί σε ένα μοντέλο παλινδρόμησης πρέπει να μετατραπεί σε όσες ψευδομεταβλητές είναι και οι

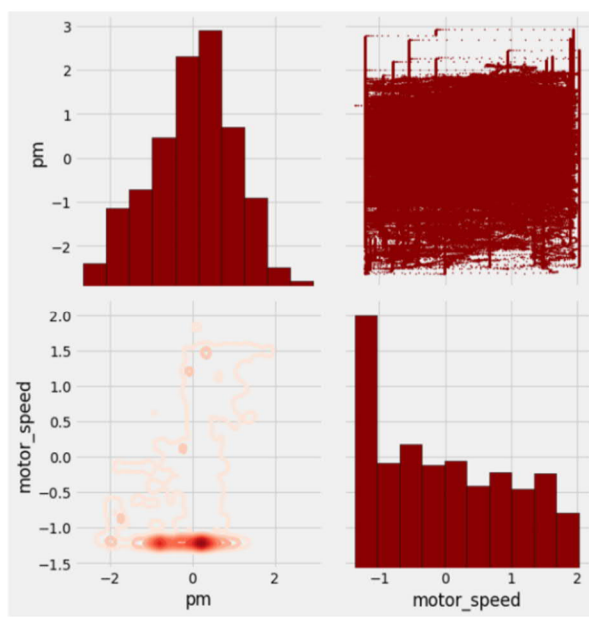
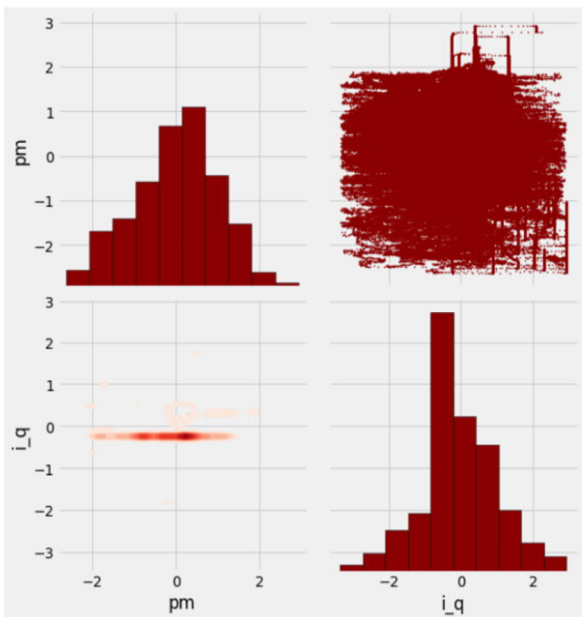
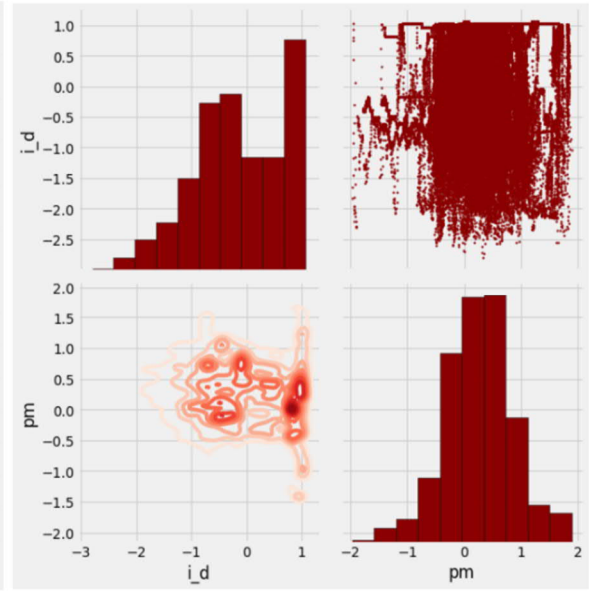
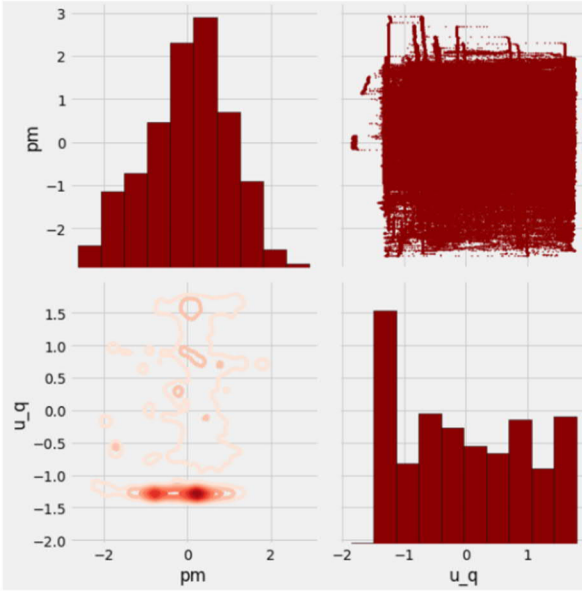
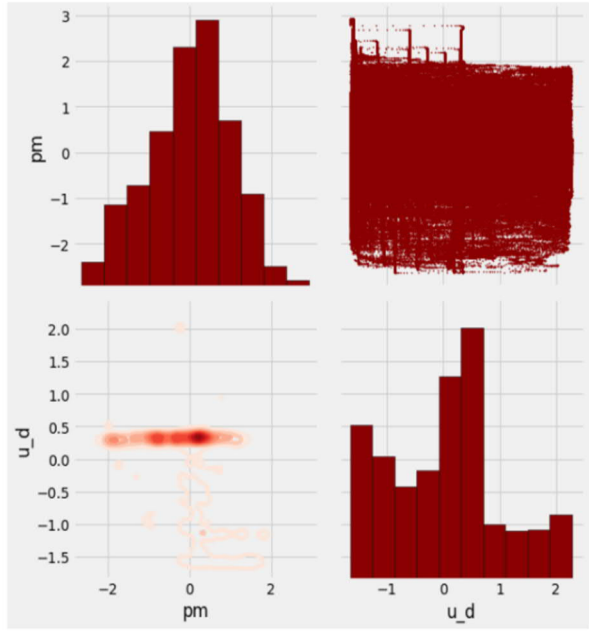
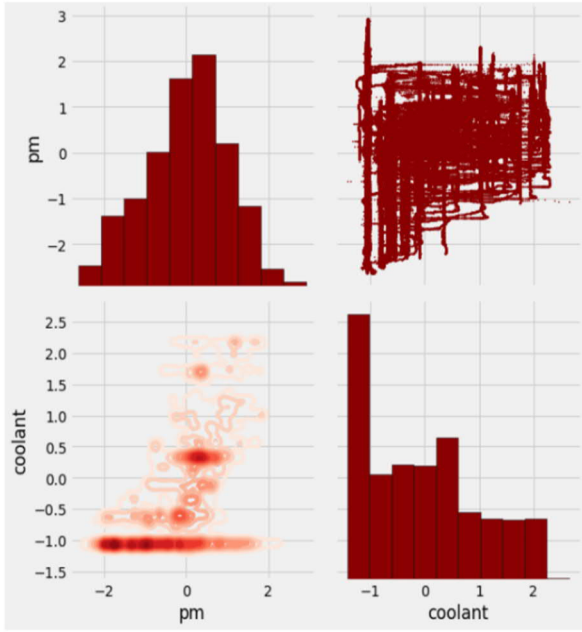
κατηγορίες, με άσσο στα δεδομένα που αντιστοιχεί η κάθε ομάδα και μηδέν στις υπόλοιπες.

Συνεπώς το τελικό μοντέλο με επεξηγηματική μεταβλητή την θερμοκρασία ρότορα 'pm', θα έχει τις εξής επεξηγηματικές μεταβλητές:

- ambient, coolant
- u\_d, u\_q
- i\_d, i\_q
- motor\_speed



Παραθέτουμε τα διαγράμματα διασποράς μεταξύ της μεταβλητής απόκρισης και των επιλεγμένων επεξηγηματικών μεταβλητών για να επιβεβαιώσουμε ότι δεν υπάρχει γραμμική συσχέτιση που θα μπερδέψει το μοντέλο.



## 4 | Κατασκευή Μοντέλου Πρόβλεψης κ' Αποτελέσματα

Για την επίλυση ενός προβλήματος οποιασδήποτε φύσης, ξεκινάμε την προσπάθεια επίλυσής του πάντα από την απλούστερη λύση, η οποία θα είναι πιο κατανοητή, πιο γρήγορη και πιο συμφέρουσα υπολογιστικά, καθώς θα χρειάζεται λιγότερη μνήμη για την υλοποίησή της. Έτσι και στην προσπάθεια μας να προβλέψουμε τη θερμοκρασία του ρότορα ενός ηλεκτρικού κινητήρα, ξεκινήσαμε από το πιο απλό μοντέλο, το γραμμικό.

### 4.1 | Γραμμική Παλινδρόμηση

Με χρήση της απλούστερης μεθόδου παλινδρόμησης, της γραμμικής παλινδρόμησης κ με τη βοήθεια της μεθόδου ελαχίστων τετραγώνων λαμβάνουμε τους συντελεστές του γραμμικού μοντέλου (εκτιμήτριες ελαχίστων τετραγώνων,  $a$ ,  $b_1, \dots, b_k$ ) καθώς και διάφορα αποτελέσματα στατιστικής συμπερασματολογίας που μας δίνονται με την εντολή 'summary' του μοντέλου.

```

Intercept:
[-0.00254332]
Coefficients:
[[ 0.36802172  0.30818506 -0.15313956 -0.32326906  0.53725465  0.03920358
 -0.05919307]]

                OLS Regression Results
=====
Dep. Variable:          pm      R-squared:                0.470
Model:                  OLS      Adj. R-squared:           0.470
Method:                 Least Squares      F-statistic:              1.266e+05
Date:                   Mon, 14 Sep 2020    Prob (F-statistic):       0.00
Time:                   21:32:26           Log-Likelihood:          -1.0947e+06
No. Observations:      998070            AIC:                     2.189e+06
Df Residuals:          998062            BIC:                     2.190e+06
Df Model:               7
Covariance Type:       nonrobust
=====
                coef      std err      t      P>|t|      [0.025      0.975]
-----
const           -0.0025      0.001      -3.506      0.000      -0.004      -0.001
ambient         0.3680      0.001     441.656      0.000      0.366      0.370
coolant         0.3082      0.001     380.001      0.000      0.307      0.310
u_d            -0.1531      0.001    -110.361      0.000      -0.156      -0.150
u_q            -0.3233      0.002    -203.817      0.000      -0.326      -0.320
motor_speed     0.5373      0.002     227.915      0.000      0.533      0.542
i_d             0.0392      0.002      24.394      0.000      0.036      0.042
i_q            -0.0592      0.001     -42.189      0.000      -0.062      -0.056
=====
Omnibus:                14790.863      Durbin-Watson:           0.001
Prob(Omnibus):          0.000      Jarque-Bera (JB):       24707.716
Skew:                   0.126      Prob(JB):                0.00
Kurtosis:               3.728      Cond. No.                6.94
=====

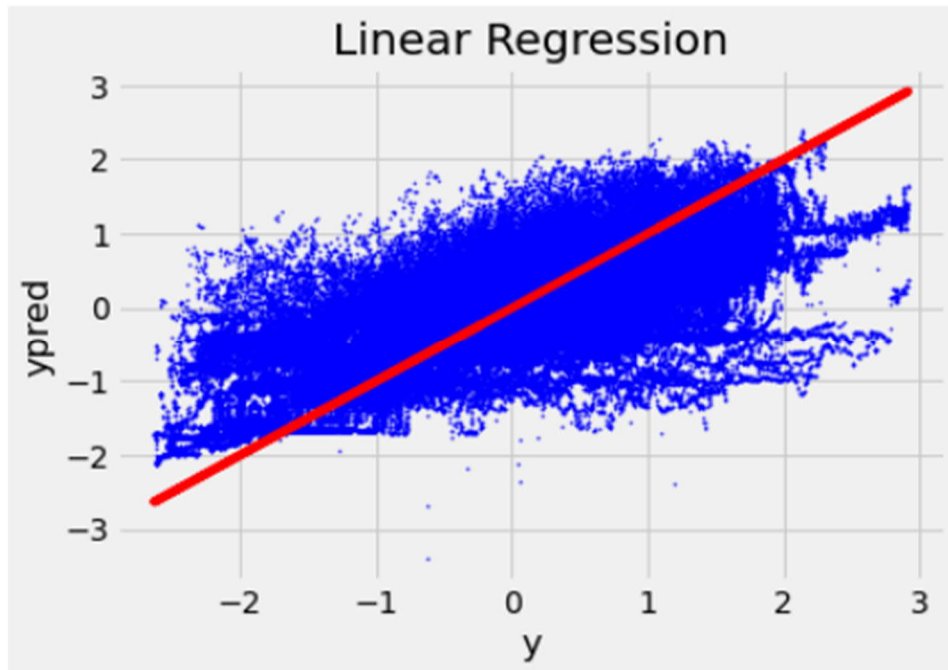
```

Η εντολή 'summary' μας δείχνει με μία ματιά το μοντέλο γραμμικής παλινδρόμησης που προσαρμόστηκε στα δεδομένα μας. Βλέπουμε αρχικά τους συντελεστές που έδωσε σε κάθε επεξηγηματική μεταβλητή και κατά συνέπεια τη σημαντικότητα της κάθε μία στη πρόβλεψη της θερμοκρασίας του ρότορα. Παρατηρώντας τον μεγαλύτερο συντελεστή να είναι το 0.54 της 4ης μεταβλητής 'motor\_speed' βλέπουμε ότι είναι και η μεταβλητή που δίνει πολύτιμη πληροφορία στο μοντέλο για να προβλέψει τη μεταβλητή απόκρισης. Παρόλα αυτά η τιμή του συντελεστή προσδιορισμού και του προσαρμοσμένου συντελεστή προσδιορισμού είναι πολύ μακριά από τη μονάδα (0.47), το οποίο απλώς δείχνει ότι το γραμμικό μοντέλο δεν προσαρμόζεται καλά στα δεδομένα. Επίσης, παρατηρούμε ότι οι p-values των επεξηγηματικών μεταβλητών για το στατιστικό έλεγχο με μηδενική υπόθεση  $H_0: \beta_k = 0$ , είναι όλες μικρότερες του 0.05 όπως και τα διαστήματα εμπιστοσύνης δεν περιέχουν το μηδέν. Συνεπώς μπορούμε να απορρίψουμε τη μηδενική υπόθεση  $H_0$ , ότι όλες οι επεξηγηματικές μεταβλητές είναι στατιστικά σημαντικές για το μοντέλο, όπως και ο σταθερός όρος.

Δίνοντας ως είσοδο στο μοντέλο γραμμικής παλινδρόμησης τις τιμές των επεξηγηματικών μεταβλητών για κάθε μέτρηση που έχουμε στο σύνολο



δεδομένων, θέλουμε να δούμε το ποσοστό που θα καταφέρει να προβλέψει τη θερμοκρασία του ρότορα. Το μοντέλο παράγει τις προβλέψεις  $\hat{y}$  (*ypred*) τις οποίες συγκρίνουμε με τις πραγματικές τους τιμές  $y$ . Στο παρακάτω γράφημα απεικονίζονται τα σημεία  $(y, \hat{y})$  καθώς και η ευθεία  $y=\hat{y}$ .



Η απόλυτα επιτυχημένη πρόβλεψη απεικονίζεται με όλα τα σημεία να συμπίπτουν με την κόκκινη ευθεία  $y=\hat{y}$  (*ypred*) ή να απέχουν πολύ λίγο από αυτή. Από το παραπάνω γράφημα των αποτελεσμάτων του γραμμικού μοντέλου βλέπουμε πως αυτό δεν συμβαίνει. Η χαμηλή ακρίβεια του μοντέλου, φαίνεται και από τις υψηλές τιμές που παρουσίασαν οι μετρικές (μέσο τετραγωνικό σφάλμα) MSE, (μέσο απόλυτο σφάλμα) MAE, RMSE, μέση τιμή των σχετικών σφαλμάτων (mean of Relative Errors), διασπορά των σχετικών σφαλμάτων (variance of Relative Errors). Συγκεκριμένα για το γραμμικό μοντέλο παλινδρόμησης οι τιμές των μετρικών με τις οποίες αξιολογήσαμε το μοντέλο μας υπολογίστηκαν ως:

RMSE : 0.7246800266586504

MSE : 0.5251611410379822

MAE : 0.5656190572659703

$R^2$  : -0.1233974278265646

Mean of relative errors : -0.5851175310254135

Variance score: 0.47134785825699277

Το γεγονός ότι το  $R^2$  είναι αρνητικό δείχνει ότι το μοντέλο δεν ακολουθεί την τάση των δεδομένων, οπότε όπως βλέπουμε και από το παραπάνω γράφημα



δεν προσαρμόζονται σε σχέση με την κόκκινη γραμμή, που δηλώνει τα σημεία στα οποία οι τιμές πρόβλεψης ισούνται με τις πραγματικές τιμές.

Η μη καλή προσαρμογή του γραμμικού μοντέλου ήταν αναμενόμενη, καθώς έχοντας ήδη παρατηρήσει τις σχέσεις μεταξύ των επεξηγηματικών μεταβλητών με τη μεταβλητή απόκρισης, η οποία δεν είναι γραμμική, γνωρίζαμε ότι η προσπάθεια πρόβλεψης της μέσω μιας γραμμικής σχέσης δεν θα είναι επιτυχημένη.

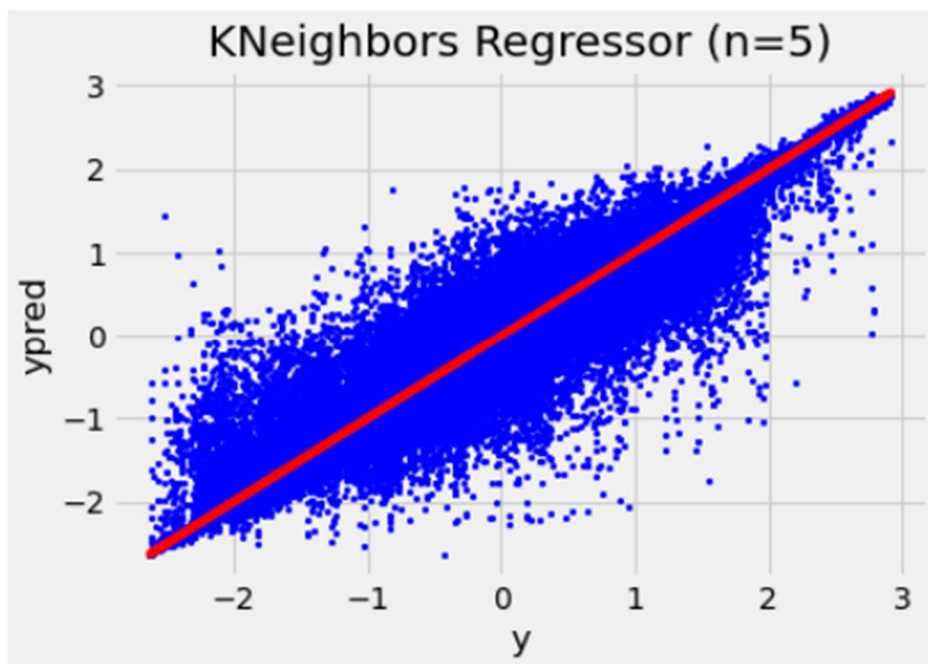
## 4.2 | Μη Γραμμική Παλινδρόμηση

Μετά την αδυναμία της γραμμικής παλινδρόμησης, να επιλύσει το πρόβλημα και να δημιουργήσει ένα δυνατό μοντέλο πρόβλεψης της θερμοκρασίας του ρότορα, συνεχίσαμε στην εφαρμογή μοντέλων μη γραμμικής παλινδρόμησης, με σκοπό να αναγνωριστούν και να εντοπιστούν πιθανές σχέσεις και μοτίβα μεταξύ των μεταβλητών απόκρισης και της επεξηγηματικής μεταβλητής. Στην ενότητα αυτή θα παρουσιαστούν τα αποτελέσματα όλων των μεθόδων που αναλύθηκαν παραπάνω, στο Κεφάλαιο 2.2.

Για την εφαρμογή των μοντέλων χρησιμοποιήθηκε κυρίως η βιβλιοθήκη `sklearn` της Python.

### 4.2.1 | Κ-Κοντινότεροι Γείτονες

Κάνοντας δοκιμές χρησιμοποιώντας διαφορετικό αριθμό κοντινότερων γειτόνων και τύπο αποστάσεων, ελέγχοντας πάντα για τη γενίκευση του μοντέλου καταλήγουμε στους 5 κοντινότερους γείτονες.



RMSE : 0.21692372117518013

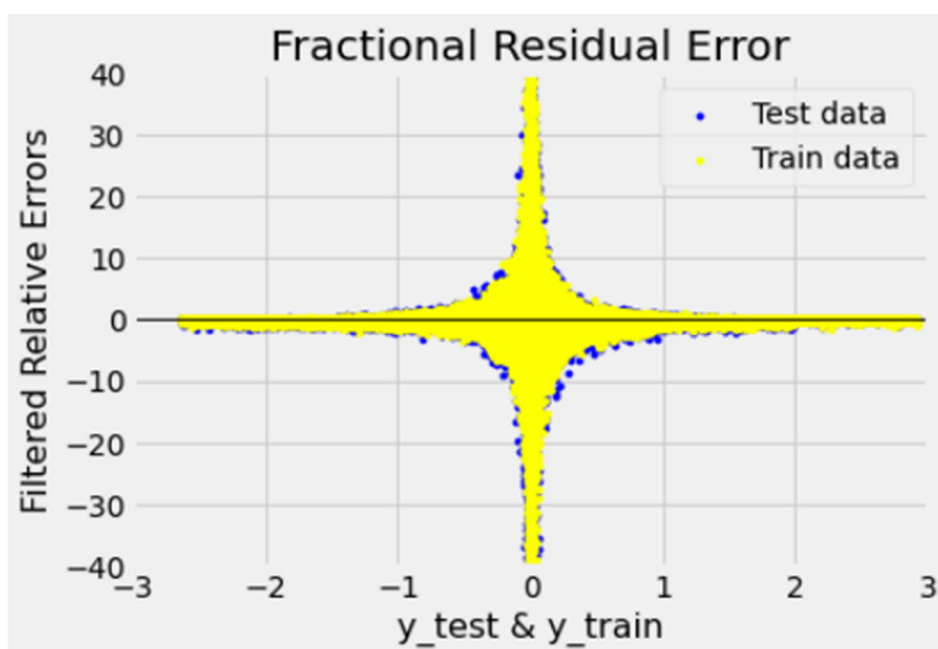
MSE : 0.04705590080848729

MAE : 0.10227216075574755

$R^2$ : 0.9508326354077614

Mean of relative errors : -0.05851130525031232

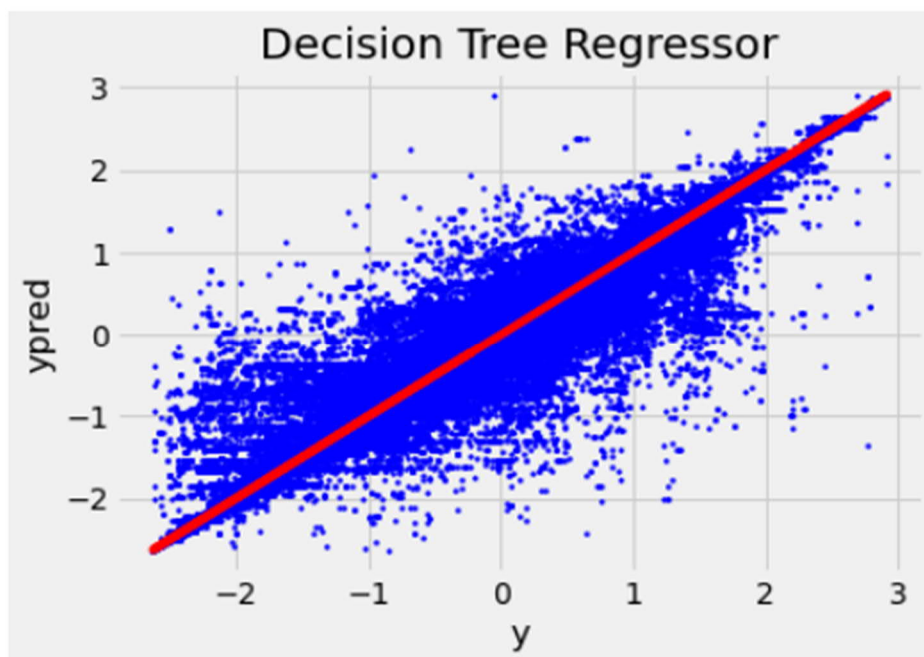
Variance score: 0.9526312958059208



Η απόδοση του μοντέλου φαίνεται να είναι αρκετά καλή αν αναλογιστούμε την απλότητα του μοντέλου.

#### 4.2.2 | Δέντρα Απόφασης

Πριν προχωρήσουμε σε ενισχυμένες μεθόδους, ξεκινάμε με δοκιμή ενός δέντρου απόφασης χωρίς να το ενισχύσουμε. Βρίσκοντας τις κατάλληλες παραμέτρους για να πετύχουμε τη μεγαλύτερη δυνατή ακρίβεια έχοντας παράλληλα ένα γενικευμένο μοντέλο, αποφεύγοντας δηλαδή την υπερεκπαίδευση του μοντέλου, έχουμε τα εξής αποτελέσματα.



RMSE : 0.26955469832241785

MSE : 0.07265973538768969

MAE : 0.14218607927112653

$R^2$ : 0.9221155391931932

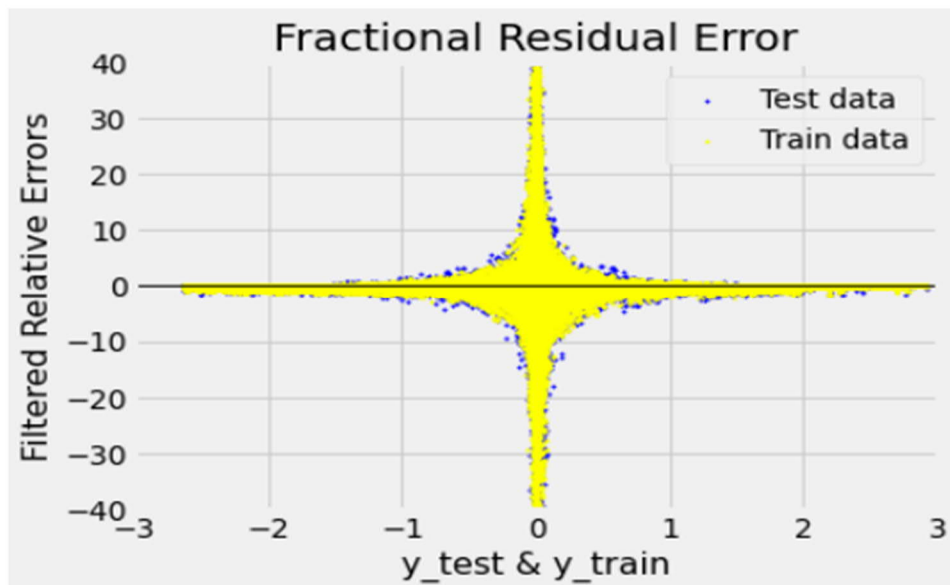
Mean of relative errors : -0.04062954548177538

Variance score: 0.47134785825699277

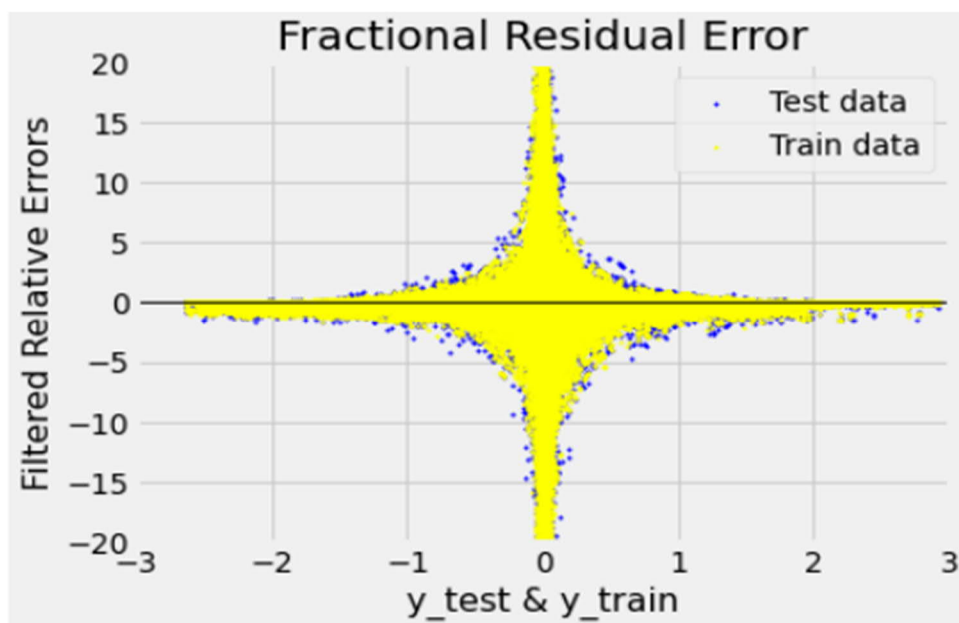
Τα αποτελέσματα είναι σαφώς καλύτερα από το γραμμικό μοντέλο και βλέπουμε ότι τα σημεία προσαρμόζονται καλύτερα στην κόκκινη γραμμή  $y = \hat{y}(ypred)$ . Οι τιμές των μετρικών MSE, MAE επιβεβαιώνουν την καλύτερη προσαρμογή σε σχέση με το γραμμικό μοντέλο καθώς είναι πιο κοντά στο μηδέν.

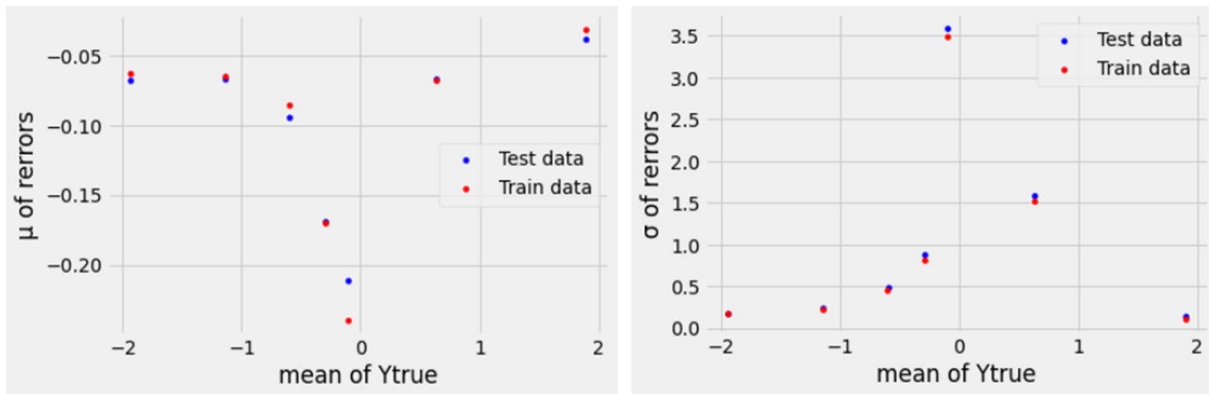
Παρατηρούμε ότι τα μεγαλύτερα σχετικά σφάλματα του μοντέλου, παρουσιάζονται για τιμές της κανονικοποιημένης θερμοκρασίας του ρότορα στο διάστημα  $(-1, 1)$ . Από την παρόμοια συμπεριφορά των σημείων εκπαίδευσης και

επαλήθευσης, βλέπουμε ότι έχουμε αποφύγει το φαινόμενο υπερπροσαρμογής και τα σημεία των σχετικών σφαλμάτων και στα δύο σύνολα σχεδόν συμπίπτουν.



Για πιο λεπτομερή αξιολόγηση του μοντέλου, αφαιρούμε τις ακραίες τιμές των σχετικών σφαλμάτων, δηλαδή αυτές που είναι  $>20$  ή  $<-20$  και παράλληλα χωρίζουμε το διάγραμμα σε ενδιάμεσα διαστήματα της θερμοκρασίας, και για κάθε ένα από αυτά βλέπουμε στα παρακάτω διαγράμματα τη μέση τιμή και τη διασπορά των σχετικών σφαλμάτων αντίστοιχα για τα δεδομένα επαλήθευσης και εκπαίδευσης ξεχωριστά.





Στα διαγράμματα της μέσης τιμής και της διασποράς η ιδανική συμπεριφορά είναι οι μέσες τιμές των σφαλμάτων των διαστημάτων να είναι όσο το δυνατόν πιο κοντά στο μηδέν και η διασπορά να είναι ομαλή μεταξύ των διαστημάτων. Η συμπεριφορά των μέσων τιμών και των τιμών της διασποράς των διαστημάτων είναι αποδεκτή, καθώς οι μέσες τιμές είναι μικρότερες από 0.5 αλλά και η διασπορά είναι ομαλή μεταξύ των διαστημάτων. Άλλη μία συμπεριφορά που πρέπει να προσέξουμε είναι οι τιμές των δεδομένων εκπαίδευσης και επαλήθευσης, οι οποίες είναι πολύ κοντά μεταξύ τους, κάτι το οποίο επαληθεύει ότι δεν έχουμε πρόβλημα υπερπροσαρμογής και ότι το μοντέλο μας είναι αρκετά γενικευμένο. Αυτή η συμπεριφορά αποτελεί κριτήριο και για το κατάλληλο βάθος δέντρων που επιλέχθηκε καθώς για μεγαλύτερες τιμές βάθους, τα σημεία απείχαν μεταξύ τους.

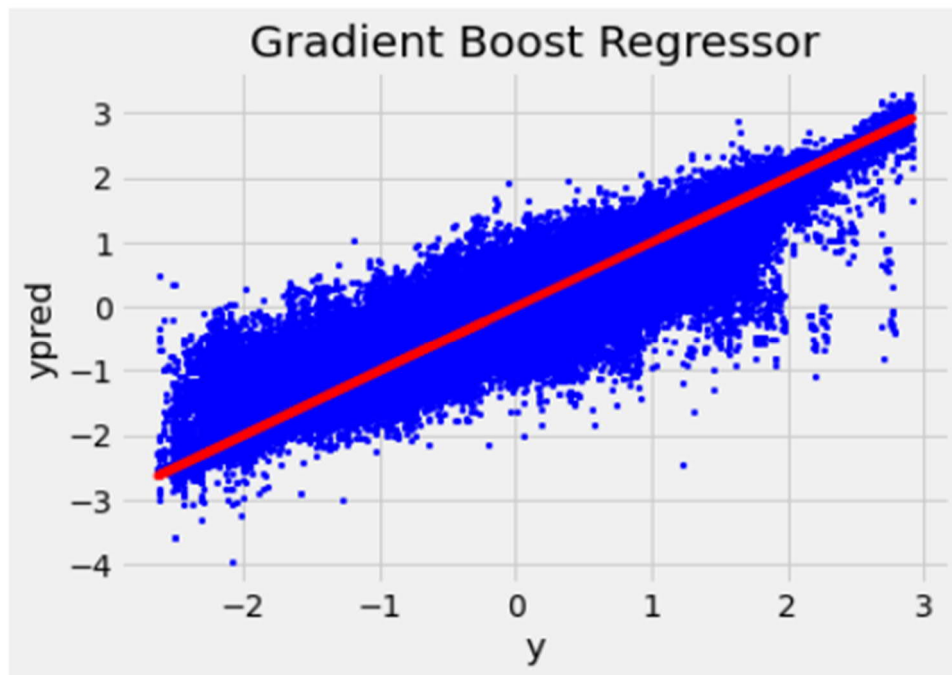
#### 4.2.3 | **Ενισχυμένα Δέντρα Απόφασης**

Έχοντας δοκιμάσει τη μέθοδο των απλών δέντρων απόφασης, θέλουμε χρησιμοποιήσουμε ενισχυτικές μεθόδους, να συγκρίνουμε τα απλά αποτελέσματα και να καταλήξουμε στην καλύτερη μέθοδο για την πρόβλεψη της θερμοκρασίας του ρότορα μίας ηλεκτρομηχανής.

- Gradient Boost **Μέθοδος**

Η Gradient Boost μέθοδος, όπως ορίζεται από τη βιβλιοθήκη sklearn της python έχει αρκετές παραμέτρους τις οποίες μπορεί να επιλέξει ο χρήστης για να πετύχει το βέλτιστο αποτέλεσμα πρόβλεψης για τα δεδομένα του. Το μόνο σίγουρο είναι ότι η μέθοδος για να λειτουργήσει, πρέπει να δέχεται αδύναμους εκτιμητές και επειδή στην περίπτωση μας θα χρησιμοποιήσουμε δέντρα απόφασης, τα οποία θέλουμε να είναι αδύναμοι εκτιμητές, συνεπώς να μην έχουν μεγάλο βάθος, θα περιορίσουμε το μέγιστο βάθος του κάθε δέντρου στα 5 φύλλα. Για την εύρεση των βέλτιστων υπόλοιπων παραμέτρων χρησιμοποιήθηκε διερεύνηση του φασικού χώρου, σταθεροποιώντας διαδοχικά τις μεταβλητές και συγκρίνοντας κάποιες μετρικές που θεωρούνται χρήσιμες για την

αξιολόγηση ενός μοντέλου πρόβλεψης. Πάντα με γνώμονα τη διατήρηση γενίκευση του μοντέλου, καταλήξαμε στα 110 δέντρα, με μέγιστο βάθος 5 και ρυθμό εκμάθησης 0,7.



RMSE : 0.30653125640911955

MSE : 0.0939614111557534

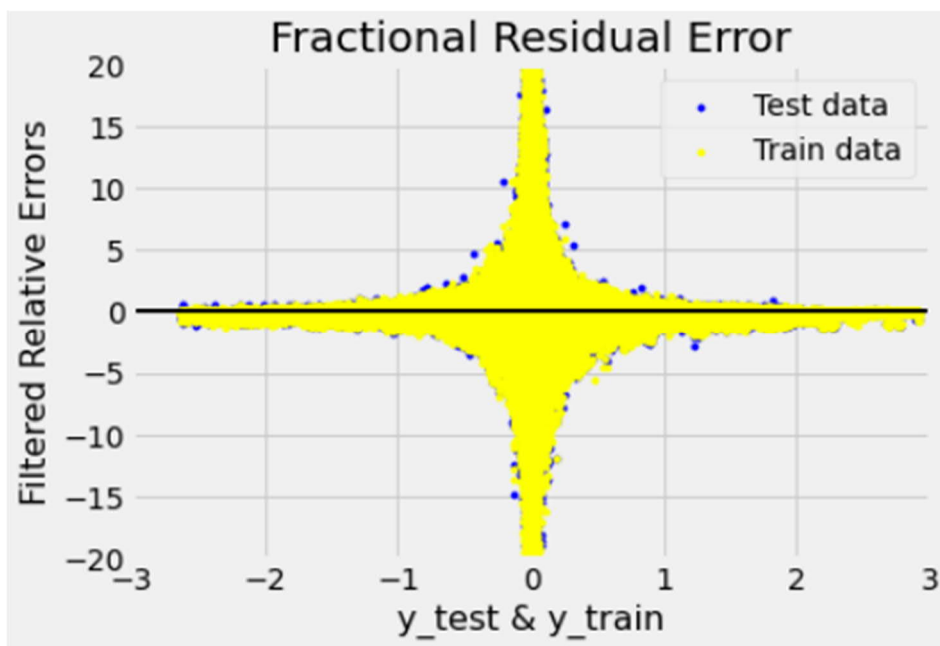
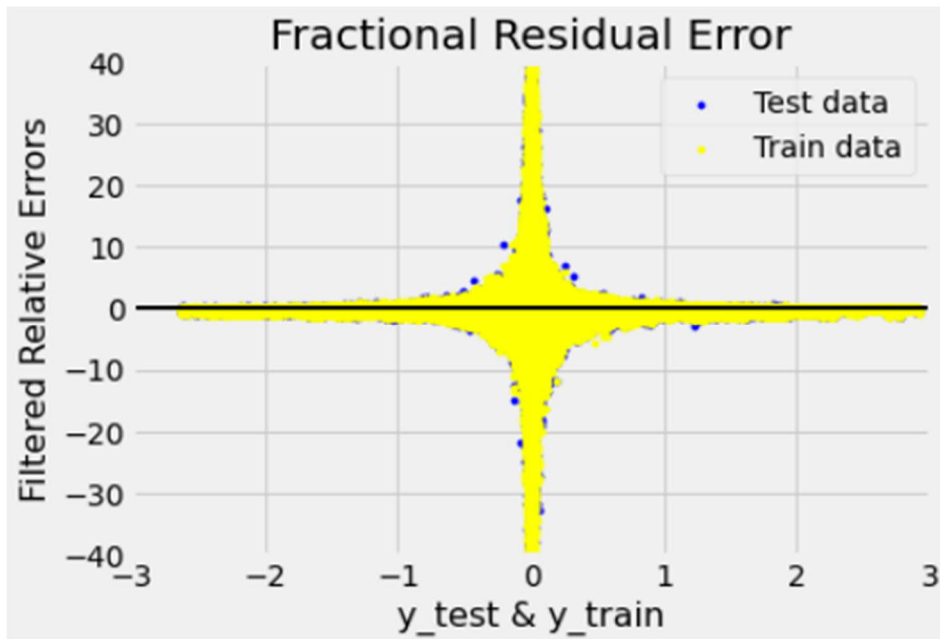
MAE : 0.210840978140686

$R^2$ : 0.8955372247868254

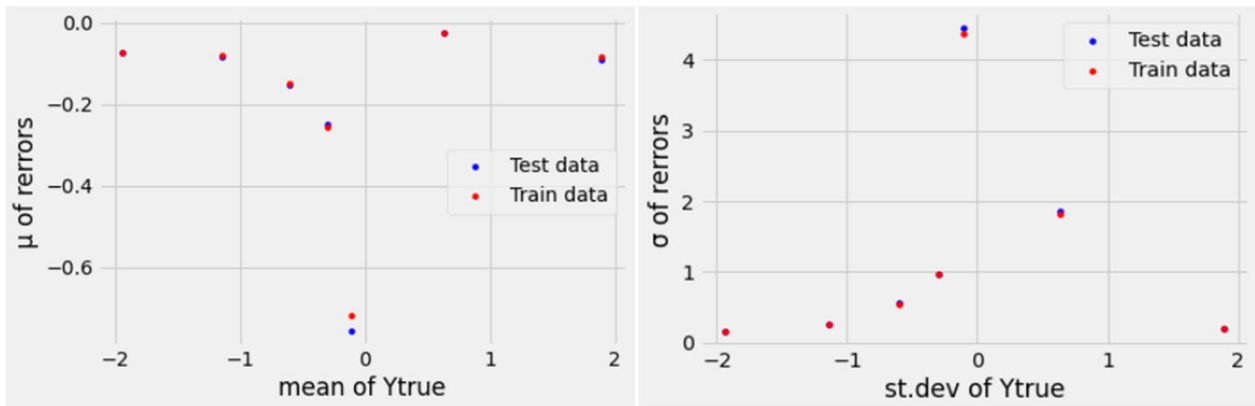
Mean of relative errors : -0.3961830446285185

Variance score: 0.9054139817913683

Οι τιμές των μετρικών φαίνεται να είναι σχεδόν ίδιες με του απλού δέντρου απόφασης. Επαναλαμβάνουμε την ανωτέρω διαδικασία με τα σχετικά σφάλματα των υπολοίπων



Και πάλι βλέπουμε ότι έχουμε αποφύγει το φαινόμενο της υπερπροσαρμογής καθώς τα σημεία του συνόλου δεδομένων που έχουμε χρησιμοποιήσει για εκπαίδευση και επαλήθευση συμπεριφέρονται με τον ίδιο τρόπο.



Η συμπεριφορά των μέσων τιμών και των τιμών της διασποράς των διαστημάτων είναι και πάλι αποδεκτή, καθώς οι μέσες τιμές είναι μικρότερες από 0.8 αλλά και η διασπορά είναι ομαλή μεταξύ των διαστημάτων. Οι τιμές των δεδομένων εκπαίδευσης και επαλήθευσης, είναι πολύ κοντά μεταξύ τους, κάτι το οποίο επαληθεύει ότι δεν έχουμε πρόβλημα υπερπροσαρμογής και ότι το μοντέλο μας είναι και πάλι αρκετά γενικευμένο.

Σε γενικές γραμμές, οι μετρικές που χρησιμοποιούμε για την αξιολόγηση των μοντέλων αλλά και τα διαγράμματα των μέσων τιμών και των διασπορών των διαστημάτων των υπολοίπων δείχνουν ότι τα αποτελέσματα του μοντέλου από το απλό δέντρο απόφασης φαίνονται να είναι καλύτερα από αυτά του ενισχυμένου δέντρου απόφασης.

### 4.3 | Νευρωνικά Δίκτυα

Για τις δοκιμές στα Νευρωνικά Δίκτυα χρησιμοποιούμε τη βιβλιοθήκη Keras της Python, με τη μέθοδο που εφαρμόζεται για τα προβλήματα παλινδρόμησης, KerasRegressor.

#### 4.3.1 | Απλό Νευρωνικό Δίκτυο

Χρησιμοποιώντας απλό νευρωνικό δίκτυο με ένα κρυφό στρώμα και δοκιμάζοντας πολλές αρχιτεκτονικές με διαφορετικά πλήθη νευρώνων καταλήγουμε στην εξής αρχιτεκτονική, που εξασφαλίζει τη γενίκευση του μοντέλου με καλό ποσοστό επιτυχίας πρόβλεψης. Επιλέγουμε τριπλάσιο αριθμό κόμβων, από το πλήθος μεταβλητών απόκρισης (δηλαδή 27 νευρώνες), για το κρυφό στρώμα με συνάρτηση ενεργοποίησης του να είναι η ανορθωτική, ReLU και συνάρτηση κόστους το MSE και το MAE. Τα αποτελέσματα του μοντέλου είναι:



RMSE : 0.47

MSE : 0.22

MAE : 0.34

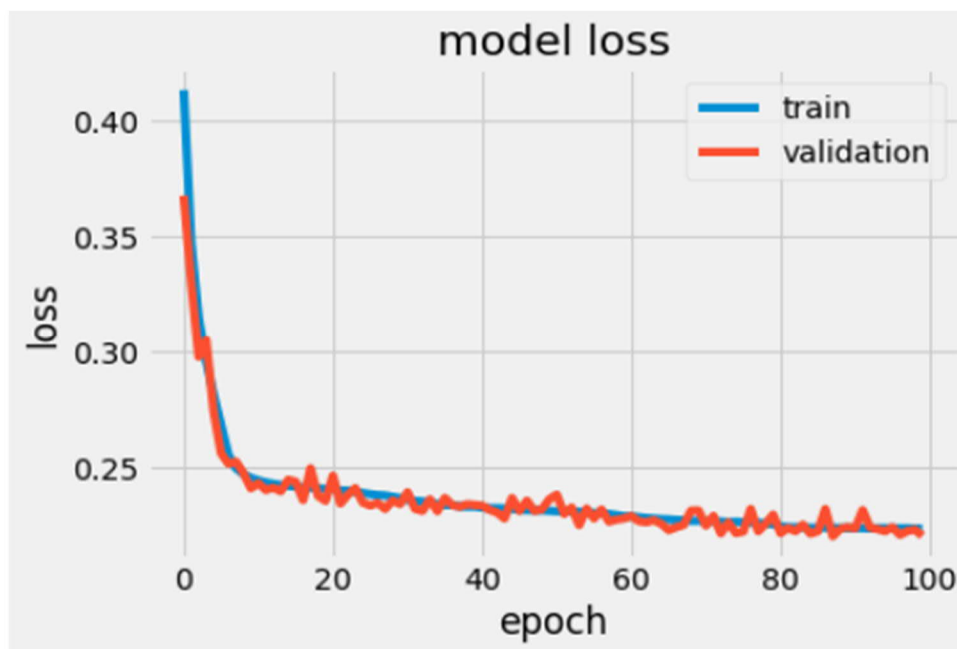
$R^2$ : 0.7

Mean of relative errors : -0.13

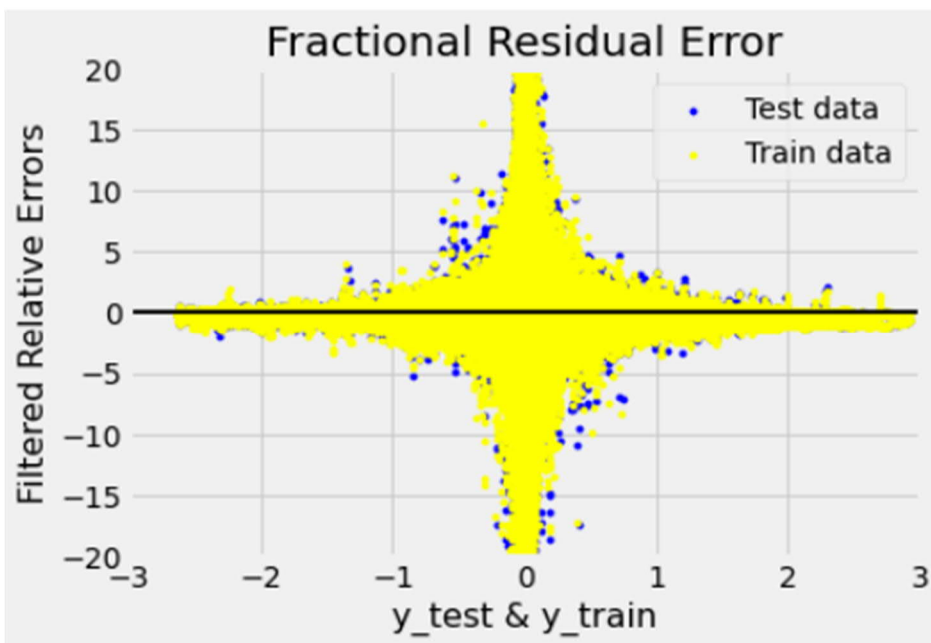
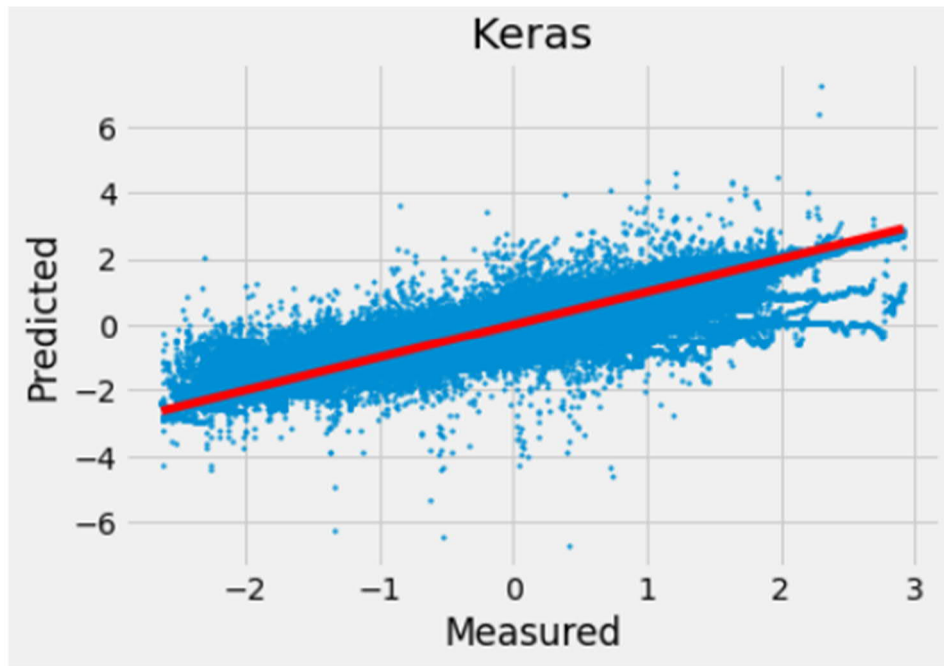
Variance score: -0.22

Standard Deviation of values : 1.0012510395392786

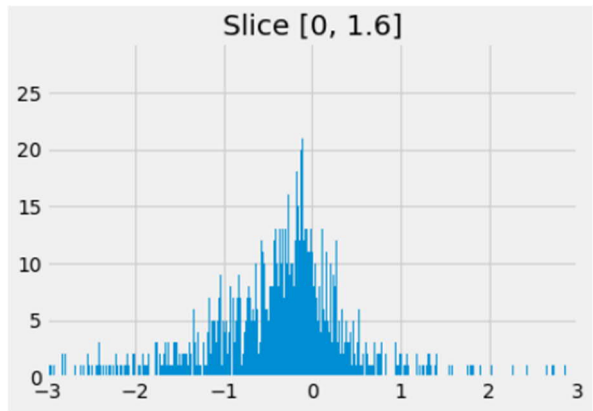
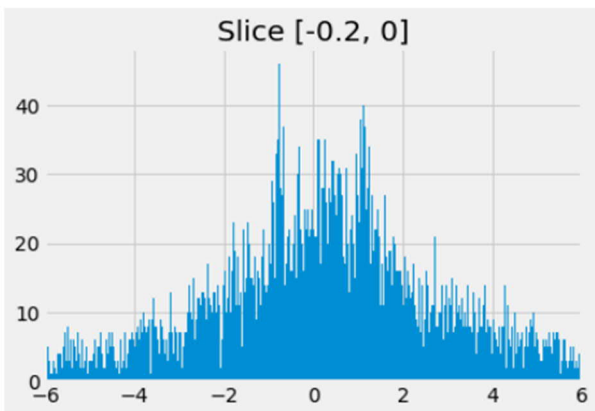
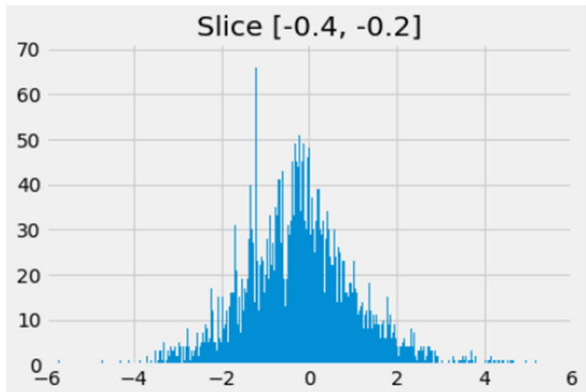
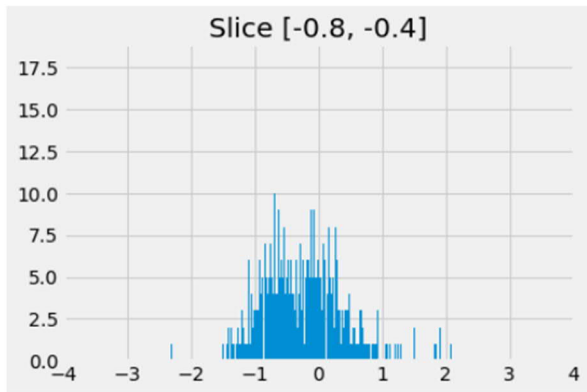
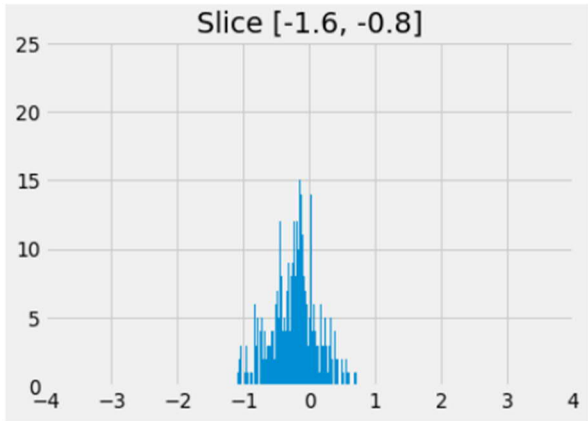
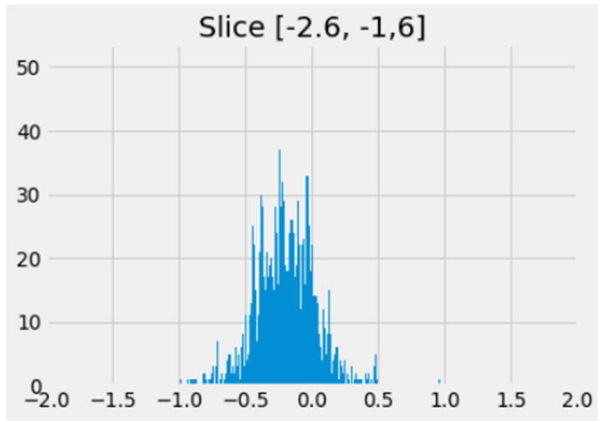
Ελέγχουμε πρώτα τη συμπεριφορά της συνάρτησης κόστους που επιλέξαμε, δηλαδή το MSE καθώς αυξάνονται οι εποχές εκπαίδευσης του νευρωνικού δικτύου, από το παρακάτω γράφημα. Παρατηρούμε ότι μέχρι τις 250 εποχές



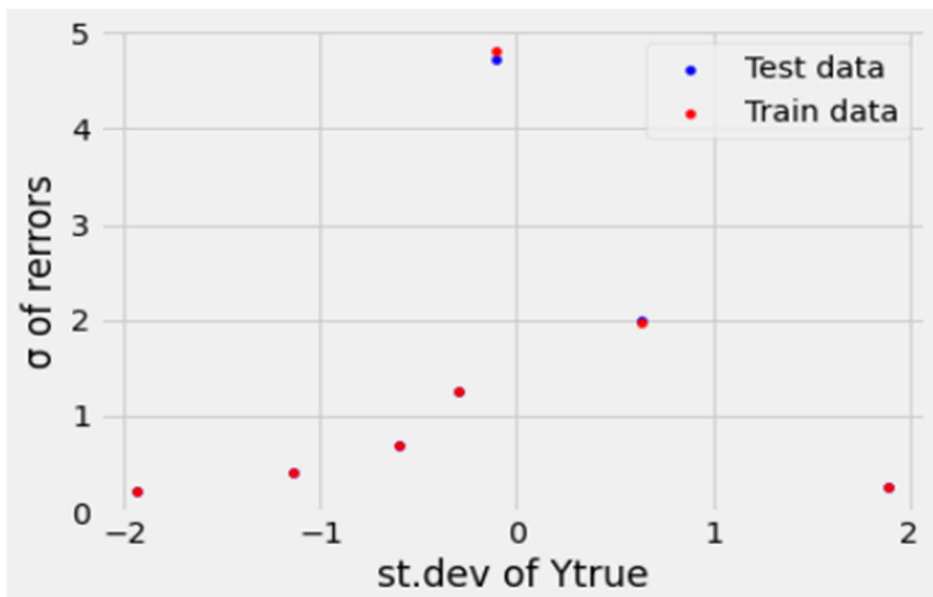
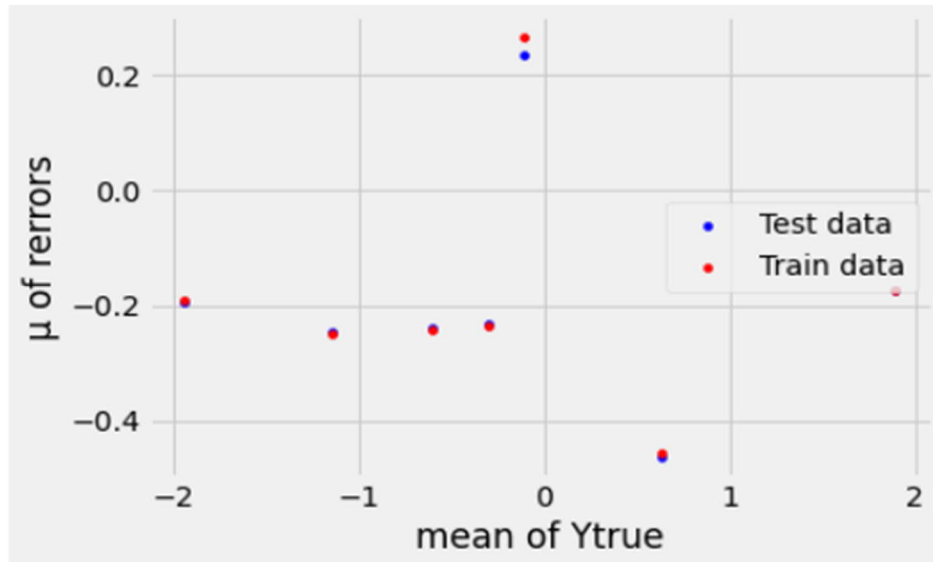
που επιλέξαμε για την εκπαίδευση, δεν υπάρχει υπερπροσαρμογή δεδομένων καθώς η απόδοση του μοντέλου είναι παρόμοια στα δεδομένα εκπαίδευσης και επαλήθευσης. Το ίδιο επιβεβαιώνουν τα παρακάτω γραφήματα των φιλτραρισμένων υπολοίπων σφαλμάτων και των μέσων τιμών και διασποράς για τα διαστήματα υπολοίπων, τα οποία συμπεριφέρονται παρόμοια για τα δύο σύνολα δεδομένων.



Χωρίζοντας σε διανύσματα τα υπόλοιπα, σιγουρευόμαστε ότι η συμπεριφορά τους είναι η αναμενόμενη σχεδιάζοντας τα ιστογράμματα κάθε διαστήματος.



Από τα παραπάνω ιστογράμματα παρατηρούμε ότι όλες οι τιμές των ιστογραμμάτων είναι κοντά στο μηδέν και κάποιες απομακρυσμένες από το μηδέν τιμές των σχετικών σφαλμάτων προκαλούν την αύξηση στην τελική μέση τιμή και διασπορά, όπως φαίνεται στα παρακάτω γραφήματα.



### 4.3.2 | Σύνθετα Νευρωνικά Δίκτυα

Όσον αφορά τα σύνθετα νευρωνικά, δεν θα επεκταθούμε σε περίπλοκες αρχιτεκτονικές, απλά προσθέτοντας και άλλα κρυφά στρώματα στην αρχιτεκτονική, θα προσπαθήσουμε να πετύχουμε τη βέλτιστη πρόβλεψη για τη θερμοκρασία του ρότορα. Έπειτα από δοκιμές καταλήγουμε στο μοντέλο με την εξής αρχιτεκτονική (χρησιμοποιούμε πολλαπλάσια του πλήθους των επεξηγηματικών μεταβλητών) :

- 1ο κρυφό στρώμα με 56 νευρώνες
- 2ο κρυφό στρώμα με 28 νευρώνες
- 3ο κρυφό στρώμα με 14 νευρώνες

Εκπαιδεύουμε σε 100 εποχές και χρησιμοποιούμε για συναρτήσεις κόστους τις μετρικές MAE και MSE και έχουμε τα εξής αποτελέσματα:

RMSE : 0.27

MSE : 0.07

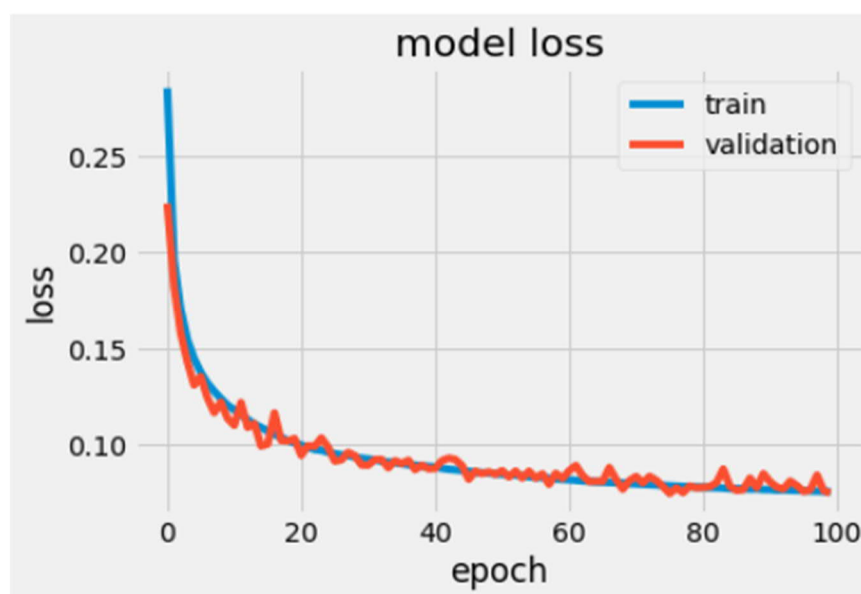
MAE : 0.18

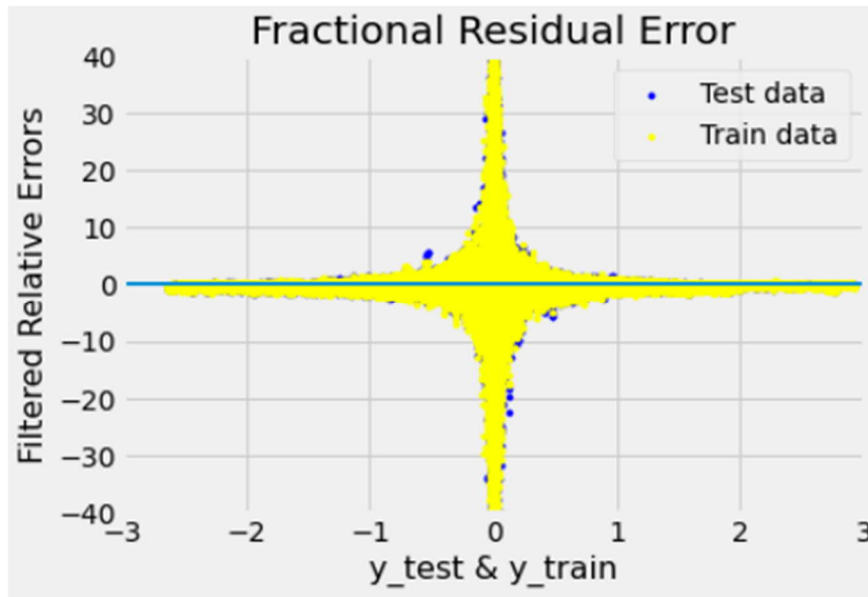
$R^2$ : 0.92

Mean of relative errors : -0.06

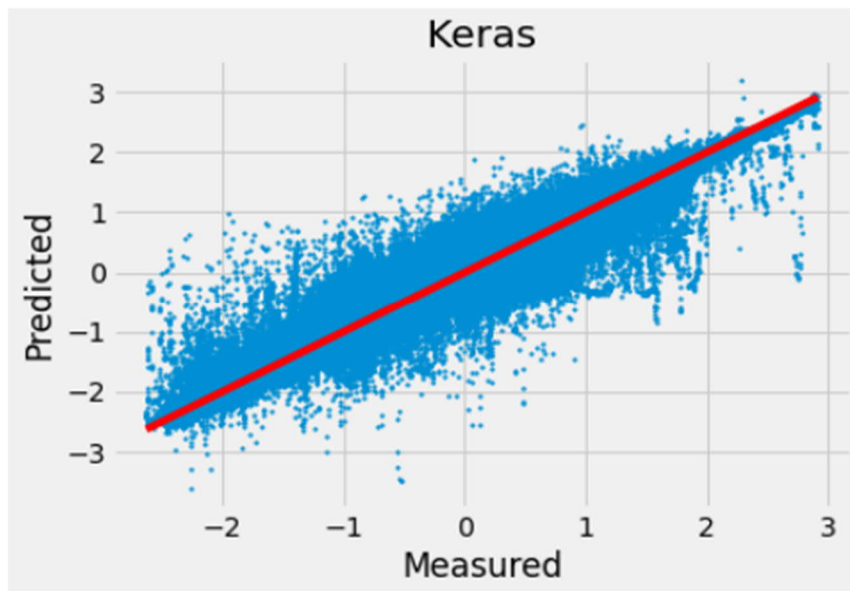
Variance score: -0.08

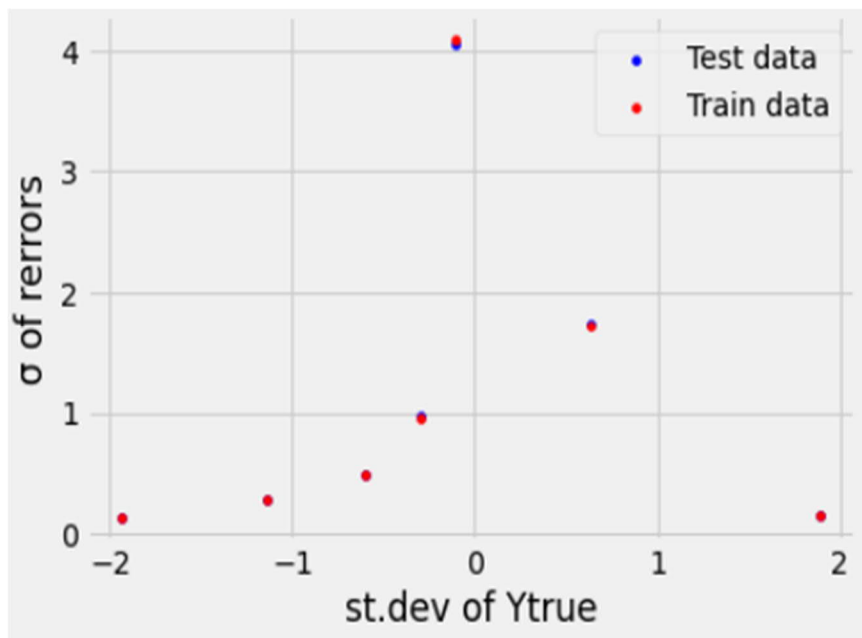
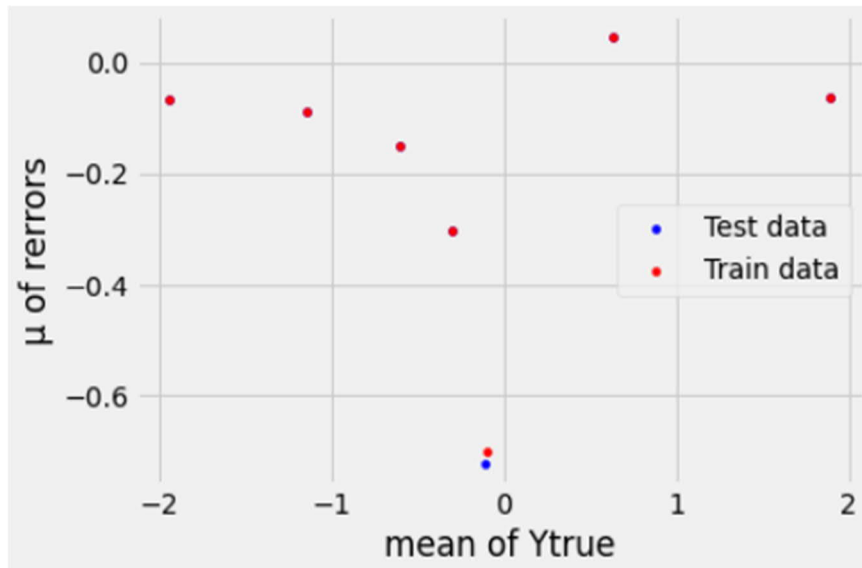
Επιβεβαιώνουμε από τα παρακάτω γραφήματα της συνάρτησης κόστους και των υπολοίπων ότι το μοντέλο δεν έχει υπερπροσαρμοστεί.





Η μορφή των προβλέψεων σε σχέση με την ευθεια  $y=y(\text{pred})$  φαίνεται στο παρακάτω γράφημα, όπως και οι μέσες τιμές και διασπορές κάθε διαστήματος που έχουμε χωρίσει όπως και στις υπόλοιπες μεθόδους.

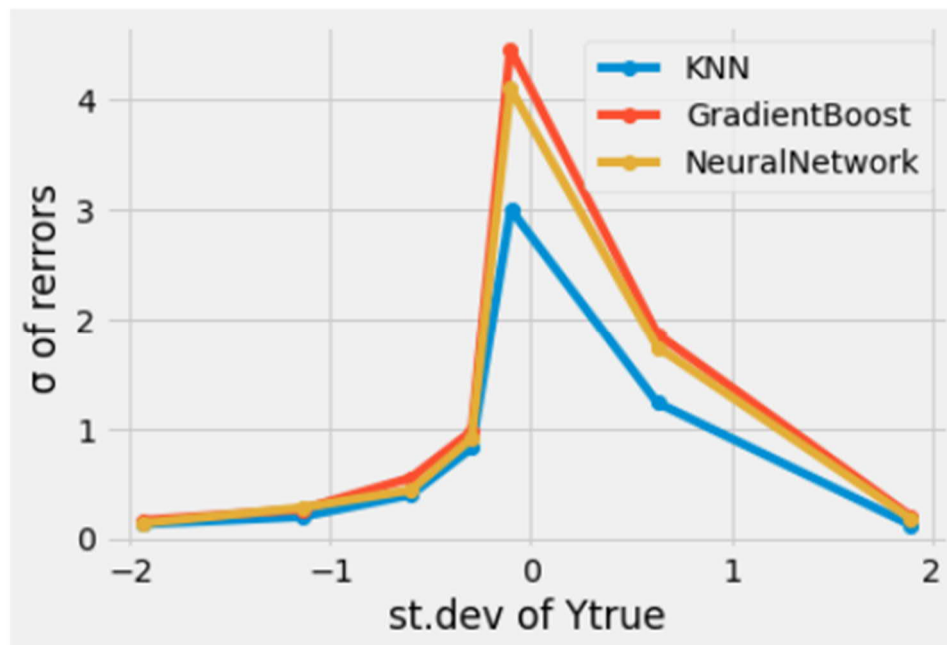
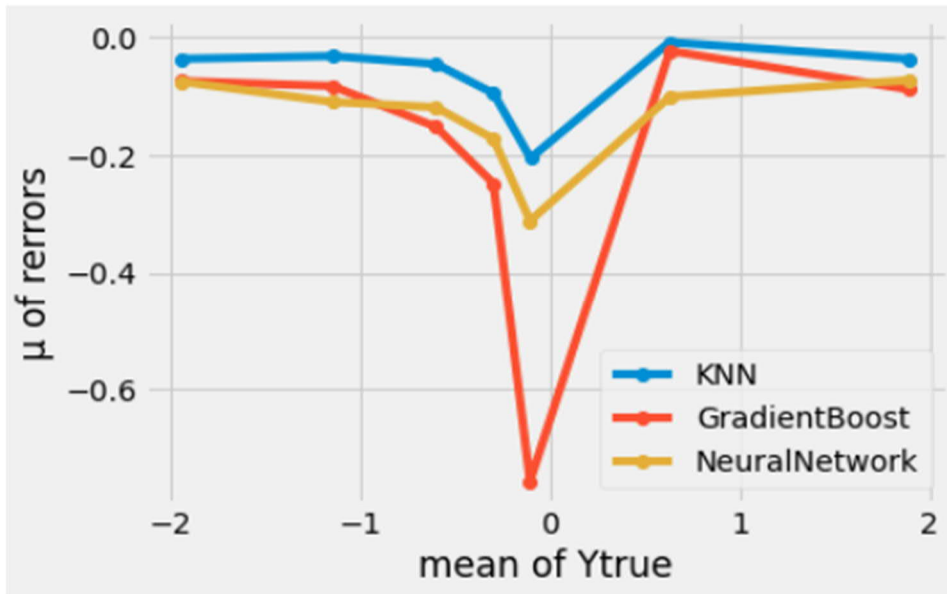




Το μοντέλο του σύνθετου νευρωνικού δικτύου φαίνεται να έχει τα καλύτερα αποτελέσματα, αλλά θα φανεί στη σύνοψη των αποτελεσμάτων.

#### 4.4 | Σύνοψη Αποτελεσμάτων

Έχοντας κάνει όλες τις απαραίτητες δοκιμές σε όλα τα μοντέλα που θέλουμε να χρησιμοποιήσουμε, συνοψίζουμε τα αποτελέσματα των τριών καλύτερων μοντέλων, σύμφωνα με τις μετρικές που έχουμε επιλέξει. Αυτά είναι το μοντέλο της μεθόδου KNN, της Gradient boost και του σύνθετου νευρωνικού δικτύου.



Παρατηρώντας τα γραφήματα βλέπουμε ότι για τα συγκεκριμένα δεδομένα η μέθοδος που προβλέπει με μικρότερη μέση τιμή σφαλμάτων και μικρότερη διασπορά σφαλμάτων τη θερμοκρασία του ρότορα μιας ΣΜΜΜ είναι η μέθοδος ΚΝΝ. Αμέσως καλύτερο είναι το νευρωνικό και ακολουθεί η Gradient Boost μέθοδος.

Άρα θα έλεγε κανείς ότι δεν υπάρχει σημαντική βελτίωση αν χρησιμοποιηθούν πιο πολύπλοκες μέθοδοι από την ΚΝΝ.



## Βιβλιογραφία

- [1] Christopher Bishop, (2006), Pattern Recognition and Machine Learning
- [2] Sergios Theodoridis, Konstantinos Koutroumbas, (1998), Pattern Recognition
- [3] Κοκολάκης Γιώργος, Φουσκάκης Δημήτρης, (2009), Στατιστική θεωρία και εφαρμογές, Συμεών
- [4] Φουσκάκης Δημήτρης, (2013), Ανάλυση Δεδομένων με χρήση της R, Τσότρας
- [5] Οικονόμου Π. Καρώνη Χ., (2010), Στατιστικά μοντέλα παλινδρόμησης, Συμεών
- [6] [Gradient Boosting from scratch. Simplifying a complex algorithm | by Prince Grover | ML Review](#)
- [7] "Multidimensional binary search trees used for associative searching", Bentley, J.L., Communications of the ACM (1975)
- [8] [Adaboost vs Gradient Boosting](#)
- [9] A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting ( Yoav Freund, Robert E Schapire - 1996)
- [10] [Permanent Magnet Synchronous Motor](#)
- [11] [PMSM optimal vector control](#)
- [12] [Neural Networks Part 1: Inside the black box](#)