



---

**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών  
*«Παραγωγή και Διαχείριση Ενέργειας»*

---

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

*«Πρόβλεψη της Απορριπτόμενης Ισχύος των  
Αιολικών Πάρκων της Ρόδου με Μεθόδους  
Μηχανικής Μάθησης»*

**Του Μεταπτυχιακού Φοιτητή**

Ιωάννη Χανή

**Επιβλέπων**

Νικόλαος Χατζηαργυρίου  
Καθηγητής Ε.Μ.Π. , Τομέας Ηλεκτρικής Ισχύος

---

Αθήνα, Σεπτέμβριος 2019





**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών  
«Παραγωγή και Διαχείριση Ενέργειας»

---

*Πρόβλεψη της Απορριπτόμενης Ισχύος των Αιολικών  
Πάρκων της Ρόδου με Μεθόδους Μηχανικής Μάθησης*

---

Μεταπτυχιακή Εργασία

του

**Ιωάννη Χανή**

**Επιβλέπων:** Νικόλαος Χατζηαργυρίου  
Καθηγητής Ε.Μ.Π. , Τομέας Ηλεκτρικής Ισχύος

---

Αθήνα, Σεπτέμβριος 2019

.....  
**ΧΑΝΗΣ ΙΩΑΝΝΗΣ**

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Α.Π.Θ.

Copyright © Χανής Ιωάννης, 2019

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

## ΕΥΧΑΡΙΣΤΙΕΣ

---

Θα ήθελα να ευχαριστήσω όλα τα άτομα τα οποία με βοήθησαν και με στήριξαν κατά τη διάρκεια εκπόνησης της μεταπτυχιακής μου εργασίας. Πρώτα από όλα, ευχαριστώ ιδιαίτερω τον Καθηγητή κ. Χατζηαργυρίου Νικόλαο, που με εμπιστεύτηκε με την παρούσα μεταπτυχιακή εργασία και μου έδωσε την ευκαιρία να ασχοληθώ με αυτό το θέμα. Ένα μεγάλο ευχαριστώ οφείλω και στον Θεόδωρο Κωνσταντίνου, Υποψήφιο Διδάκτορα στον τομέα Ηλεκτρικής Ισχύος του ΕΜΠ, για την πολύτιμη βοήθειά του, την καθοδήγηση που μου προσέφερε και τον χρόνο που αφιέρωσε. Θα ήθελα επίσης να ευχαριστήσω θερμά τους γονείς μου και τον αδερφό μου, που ήταν πάντα δίπλα μου όλα αυτά τα χρόνια και στήριξαν τις επιλογές μου. Τέλος, οφείλω ένα μεγάλο ευχαριστώ στους φίλους μου και τους ανθρώπους που στάθηκαν δίπλα μου κατά της διάρκεια εκπόνησης της εργασίας, για την αγάπη τους, την υποστήριξή τους και την κατανόηση που μου έδειξαν.



## ΣΥΝΟΨΗ

---

<b>ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ:</b>	<b>«Πρόβλεψη της Απορριπτόμενης Ισχύος των Αιολικών Πάρκων της Ρόδου με Μεθόδους Μηχανικής Μάθησης»</b>
<b>ΦΟΙΤΗΤΗΣ:</b>	<b>Χανής Ιωάννης</b>
<b>ΕΠΙΒΛΕΠΩΝ:</b>	<b>Χατζηαργυρίου Νικόλαος, Καθηγητής, Τομέας Ηλεκτρικής Ισχύος Ε.Μ.Π.</b>
<b>ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ:</b>	<b>2018 - 19</b>

### Σύνοψη

Στην παρούσα μεταπτυχιακή εργασία θα υπολογιστεί η ετήσια απορριπτόμενη ισχύς των αιολικών πάρκων της Ρόδου, δηλαδή η ισχύς που μπορεί να παραχθεί από τα πάρκα αλλά δε χρησιμοποιείται, γιατί ο διαχειριστής του συστήματος ζητάει λιγότερη ισχύ. Οι υπολογισμοί θα βασιστούν σε προβλέψεις της αιολικής ισχύος των πάρκων, οι οποίες θα προκύψουν με χρήση τεχνικών μηχανικής μάθησης (machine learning). Θα εκπαιδευτούν διάφορα μοντέλα παλινδρόμησης χρησιμοποιώντας δείγματα από μια γνωστή χρονοσειρά ανεμολογικών δεδομένων, και θα υπολογιστούν οι βέλτιστες τιμές των παραμέτρων τους. Το καλύτερο μοντέλο από αυτά θα χρησιμοποιηθεί στη συνέχεια για πραγματοποίηση προβλέψεων ισχύος: το μοντέλο θα κάνει μια εκτίμηση της ετήσιας ισχύος που παράγεται από κάθε πάρκο, βασιζόμενο στα δεδομένα με τα οποία εκπαιδεύτηκε. Αν από αυτή την τιμή αφαιρέσουμε την ετήσια ισχύ που χρησιμοποιείται από τον διαχειριστή του δικτύου, η οποία είναι γνωστή, θα προκύψει η τιμή της ετήσιας απορριπτόμενης ισχύος του κάθε πάρκου.

# ABSTRACT

---

***POST-GRADUATE THESIS:*** «Calculation of the Power Rejected from the Rhodes Wind Farms by means of Machine Learning Methods»

***STUDENT:*** Ioannis Hanis

***SUPERVISOR:*** Hatziargyriou Nikolaos, Professor, National Technical University of Athens

***ACADEMIC YEAR:*** 2018 - 19

## **Abstract**

Within the context of this postgraduate thesis we will calculate the power that is rejected annually from the Rhodes wind farms. This is the power that can be produced by the farms but is not harnessed, because the grid administrator requests less power. The calculations will be based on wind power predictions carried out by machine learning models. A variety of regression models will be trained on samples taken from a known time series of wind data, and their hyper-parameters will be tuned to their optimal values. The best model trained will then be used to conduct power predictions: it will estimate the power produced annually by each park, based on the data on which it was trained. From this value we can then subtract the value of the power that is annually requested from the grid administrator, in order to get an estimation of the power which is annually rejected from each wind farm.



## ΠΕΡΙΕΧΟΜΕΝΑ

<b>ΕΥΧΑΡΙΣΤΙΕΣ</b> .....	<b>iii</b>
<b>ΣΥΝΟΨΗ</b> .....	<b>v</b>
<b>ABSTRACT</b> .....	<b>vi</b>
<b>ΠΕΡΙΕΧΟΜΕΝΑ</b> .....	<b>vii</b>
<b>ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ</b> .....	<b>x</b>
<b>ΛΙΣΤΑ ΠΙΝΑΚΩΝ</b> .....	<b>xiii</b>
<b>ΚΕΦΑΛΑΙΟ 1 – ΕΙΣΑΓΩΓΗ</b> .....	<b>1</b>
1.1 – Αντικείμενο της Εργασίας.....	1
1.2 – Δομή της Εργασίας.....	2
<b>ΚΕΦΑΛΑΙΟ 2 – ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ</b> .....	<b>5</b>
2.1 – Γενικά.....	5
2.2 – Επιβλεπόμενη Μάθηση .....	7
2.2.1 – Δημιουργία Συνόλου Εκπαίδευσης.....	7
2.2.2 – Ταξινόμηση (Classification) .....	8
2.2.3 – Παλινδρόμηση (Regression).....	9
2.3 – Εφαρμογή στην Πρόβλεψη Αιολικής Ισχύος.....	10
<b>ΚΕΦΑΛΑΙΟ 3 – ΜΕΘΟΔΟΣ</b> .....	<b>13</b>
3.1 – Δεδομένα.....	13
3.1.1 – Διαθέσιμα Ανεμολογικά Δεδομένα .....	13
3.1.2 – Προεπεξεργασία Δεδομένων .....	14
3.1.3 – Πλαίσιο Υλοποίησης .....	15
3.2 – Μέθοδος Εκπαίδευσης των Μοντέλων Παλινδρόμησης .....	16
3.2.1 – Διαχωρισμός σε Σύνολο Εκπαίδευσης και Σύνολο Δοκιμής.....	16
3.2.2 – Βέλτιστη Ρύθμιση των Υπερ-παραμέτρων (Hyperparameter Tuning).....	17
3.3 - Κριτήρια Αξιολόγησης των Μοντέλων .....	21
<b>ΚΕΦΑΛΑΙΟ 4 – ΔΟΚΙΜΕΣ ΜΟΝΤΕΛΩΝ ΠΑΛΙΝΔΡΟΜΗΣΗΣ</b> .....	<b>23</b>
4.1 – Εισαγωγή.....	23
4.2 – Μοντέλα Γραμμικής Παλινδρόμησης .....	24
4.2.1 – Μέθοδος Κανονικών Ελαχίστων Τετραγώνων (Ordinary Least Squares) .....	24
4.2.2 – Ridge Regression .....	25
4.2.3 – Lasso .....	26
4.2.4 – Elastic Net.....	27
4.2.5 – Bayesian Ridge Regression .....	29
4.2.6 – Stochastic Gradient Descent (SGD) .....	30

4.3 – Παλινδρόμηση Kernel Ridge .....	32
4.3.1 – Συναρτήσεις Πυρήνα .....	32
4.3.2 – Kernel Ridge Regression .....	33
4.4 – Μηχανές Διαनुσμάτων Υποστήριξης (Support Vector Machines – SVM).....	35
4.4.1 – Γενικά .....	35
4.4.2 – Support Vector Regression (SVR).....	37
4.5 – Μοντέλα Κοντινότερων Γειτόνων .....	39
4.5.1 – Γενικά .....	39
4.5.2 – KNeighbors Regressor .....	40
4.5.3 – RadiusNeighbors Regressor.....	41
4.6 – Δέντρα Αποφάσεων (Decision Trees).....	43
4.7 – Αλγόριθμοι Μάθησης Συνόλων (Ensemble Methods).....	46
4.7.1 – Random Forest Regressor .....	46
4.7.2 – Extra Trees Regressor .....	48
4.7.3 – AdaBoost Regressor .....	50
4.7.4 – Gradient Boosting Regressor .....	52
4.8 – Τεχνητά Νευρωνικά Δίκτυα.....	55
4.8.1 – Γενικά .....	55
4.8.2 – Multi-Layer Perceptron (MLP).....	56
4.9 – Σύγκριση Μοντέλων .....	59
<b>ΚΕΦΑΛΑΙΟ 5 – ΚΑΜΠΥΛΕΣ ΕΚΠΑΙΔΕΥΣΗΣ ΒΕΛΤΙΣΤΩΝ ΜΟΝΤΕΛΩΝ.....</b>	<b>61</b>
5.1 – Εισαγωγή.....	61
5.2 – Καμπύλες Εκπαίδευσης Βέλτιστων Μοντέλων .....	63
5.2.1 – Support Vector Regressor (SVR) .....	63
5.2.2 – KNeighbors Regressor .....	68
5.2.3 – Random Forest Regressor .....	73
5.3 – Επιλογή Τελικού Βέλτιστου Μοντέλου .....	80
<b>ΚΕΦΑΛΑΙΟ 6 – ΠΡΟΒΛΕΨΗ ΙΣΧΥΟΣ ΑΙΟΛΙΚΩΝ ΠΑΡΚΩΝ.....</b>	<b>81</b>
6.1 – Εισαγωγή.....	81
6.2 – Αποτελέσματα για τα Πάρκα .....	82
6.2.1 – Αιολικό Πάρκο 1 (Aeiforiki) .....	82
6.2.2 – Αιολικό Πάρκο 2 (Dei) .....	85
6.2.3 – Αιολικό Πάρκο 3 (Diethnis) .....	88
6.2.4 – Αιολικό Πάρκο 4 (EuroWind) .....	91
6.2.5 – Αιολικό Πάρκο 5 (Atabyrou).....	94
6.3 – Συμπεράσματα.....	97
<b>ΚΕΦΑΛΑΙΟ 7 – ΕΠΙΛΟΓΟΣ .....</b>	<b>99</b>
7.1 – Γενικά Συμπεράσματα.....	99
7.2 – Ανοιχτά Θέματα .....	100

*Περιεχόμενα*

---

<b>ΠΑΡΑΡΤΗΜΑ Α - ΚΩΔΙΚΑΣ .....</b>	<b>103</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>111</b>

## ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ

---

<b>Σχήμα 2.1</b> – Ένα απλό πρόβλημα Ταξινόμησης. Ο αλγόριθμος ταξινόμησης μπορεί να προβλέψει το χρώμα ενός νέου δείγματος, ανάλογα με τη θέση του ως προς τη διαχωριστική γραμμή. [7].....	8
<b>Σχήμα 2.2</b> – Ένα απλό πρόβλημα Παλινδρόμησης. [7] .....	9
<b>Σχήμα 3.1</b> – Τα διαδοχικά στάδια της τεχνικής k-fold Cross-Validation, για $k = 5$ . [22] .....	19
<b>Σχήμα 3.2</b> – Συνοπτικό σχεδιάγραμμα της διαδικασίας που περιγράφηκε σε αυτή την ενότητα, για την εκπαίδευση και την αξιολόγηση των μοντέλων παλινδρόμησης. ....	20
<b>Σχήμα 4.1</b> – Τεχνική μεταφοράς σε επίπεδο Hilbert υψηλότερων διαστάσεων για ένα απλό πρόβλημα ταξινόμησης: οι κουκκίδες (κόκκινες για τη μία κλάση και μαύρες για την άλλη) δε μπορούν να διαχωριστούν γραμμικά στο αρχικό δισδιάστατο επίπεδο. Μετά τη μεταφορά τους σε ένα τρισδιάστατο επίπεδο, μπορεί εύκολα να σχεδιαστεί ένα επίπεδο που να χωρίζει τις κουκκίδες των δύο κλάσεων. [4].....	33
<b>Σχήμα 4.2</b> – Εύρεση του βέλτιστου υπερ-επιπέδου για διαχωρισμό των κλάσεων από μια Μηχανή Διανυσμάτων Υποστήριξης, σε ένα απλό πρόβλημα ταξινόμησης .....	35
<b>Σχήμα 4.3</b> – Ένα απλό παράδειγμα γραμμικής παλινδρόμησης με χρήση Μηχανών Διανυσμάτων Υποστήριξης. [44] .....	36
<b>Σχήμα 4.4</b> – Η δομή ενός δέντρου αποφάσεων (decision tree).....	43
<b>Σχήμα 4.5</b> – Η δομή ενός τεχνητού νευρωνικού δικτύου (artificial neural network). ....	55
<b>Σχήμα 5.1</b> – α) Φαινόμενο Underfitting: για χαμηλές τιμές της προς ρύθμιση υπερ-παραμέτρου, το σφάλμα είναι πολύ υψηλό και το μοντέλο δεν μπορεί να «μάθει» το σύνολο εκπαίδευσης [65]. β) Φαινόμενο Overfitting: το σημείο στο οποίο οι δύο καμπύλες αρχίζουν να απομακρύνονται μεταξύ τους σηματοδοτεί την έναρξη του Overfitting [65]. ....	62
<b>Σχήμα 5.2</b> – α) Καμπύλες εκπαίδευσης του μοντέλου SVR ως προς την τιμή της υπερ-παραμέτρου gamma. β) Χρόνος εκπαίδευσης του μοντέλου SVR ως προς την τιμή της υπερ-παραμέτρου gamma. ....	64
<b>Σχήμα 5.3</b> – α) Καμπύλες εκπαίδευσης του μοντέλου SVR ως προς την τιμή της υπερ-παραμέτρου C. β) Χρόνος εκπαίδευσης του μοντέλου SVR ως προς την τιμή της υπερ-παραμέτρου C.....	66
<b>Σχήμα 5.4</b> – α) Καμπύλες εκπαίδευσης του μοντέλου SVR ως προς την τιμή της υπερ-παραμέτρου epsilon. β) Χρόνος εκπαίδευσης του μοντέλου SVR ως προς την τιμή της υπερ-παραμέτρου epsilon.....	67
<b>Σχήμα 5.5</b> – α) Καμπύλες εκπαίδευσης του μοντέλου KNeighborsRegressor ως προς την τιμή της υπερ-παραμέτρου leaf_size. β) Χρόνος εκπαίδευσης του μοντέλου KNeighborsRegressor ως προς την τιμή της υπερ-παραμέτρου leaf_size.....	69

<b>Σχήμα 5.6</b> – α) Καμπύλες εκπαίδευσης του μοντέλου KNeighborsRegressor ως προς την τιμή της υπερ-παραμέτρου n_neighbors. β) Χρόνος εκπαίδευσης του μοντέλου KNeighborsRegressor ως προς την τιμή της υπερ-παραμέτρου n_neighbors. ....	71
<b>Σχήμα 5.7</b> – α) Καμπύλες εκπαίδευσης του μοντέλου KNeighborsRegressor ως προς την τιμή της υπερ-παραμέτρου p. β) Χρόνος εκπαίδευσης του μοντέλου KNeighborsRegressor ως προς την τιμή της υπερ-παραμέτρου p.....	72
<b>Σχήμα 5.8</b> – α) Καμπύλες εκπαίδευσης του μοντέλου RandomForestRegressor ως προς την τιμή της υπερ-παραμέτρου n_estimators. β) Χρόνος εκπαίδευσης του μοντέλου RandomForestRegressor ως προς την τιμή της υπερ-παραμέτρου n_estimators.....	74
<b>Σχήμα 5.9</b> – α) Καμπύλες εκπαίδευσης του μοντέλου RandomForestRegressor ως προς την τιμή της υπερ-παραμέτρου max_depth. β) Χρόνος εκπαίδευσης του μοντέλου RandomForestRegressor ως προς την τιμή της υπερ-παραμέτρου max_depth. ....	76
<b>Σχήμα 5.10</b> – α) Καμπύλες εκπαίδευσης του μοντέλου RandomForestRegressor ως προς την τιμή της υπερ-παραμέτρου min_samples_split. β) Χρόνος εκπαίδευσης του μοντέλου RandomForestRegressor ως προς την τιμή της υπερ-παραμέτρου min_samples_split. ....	78
<b>Σχήμα 5.11</b> – α) Καμπύλες εκπαίδευσης του μοντέλου RandomForestRegressor ως προς την τιμή της υπερ-παραμέτρου min_samples_leaf. β) Χρόνος εκπαίδευσης του μοντέλου RandomForestRegressor ως προς την τιμή της υπερ-παραμέτρου min_samples_leaf. ....	79
<b>Σχήμα 6.1</b> – Διαγράμματα του αιολικού πάρκου 1 (Aeiforiki) για το έτος 2017: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς. ....	83
<b>Σχήμα 6.2</b> – Διαγράμματα του αιολικού πάρκου 1 (Aeiforiki) για το έτος 2018: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς. ....	84
<b>Σχήμα 6.3</b> – Διαγράμματα του αιολικού πάρκου 2 (Dei) για το έτος 2017: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς. ....	86
<b>Σχήμα 6.4</b> – Διαγράμματα του αιολικού πάρκου 2 (Dei) για το έτος 2018: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς. ....	87
<b>Σχήμα 6.5</b> – Διαγράμματα του αιολικού πάρκου 3 (Diethnis) για το έτος 2017: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς. ....	89
<b>Σχήμα 6.6</b> – Διαγράμματα του αιολικού πάρκου 3 (Diethnis) για το έτος 2018: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς. ....	90
<b>Σχήμα 6.7</b> – Διαγράμματα του αιολικού πάρκου 4 (EuroWind) για το έτος 2017: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς. ....	92
<b>Σχήμα 6.8</b> – Διαγράμματα του αιολικού πάρκου 4 (EuroWind) για το έτος 2018: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς. ....	93
<b>Σχήμα 6.9</b> – Διαγράμματα του αιολικού πάρκου 5 (Atabyrou) για το έτος 2017: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς. ....	95

**Σχήμα 6.10** – Διαγράμματα του αιολικού πάρκου 5 (Atabyrou) για το έτος 2018: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς. ....96

## ΛΙΣΤΑ ΠΙΝΑΚΩΝ

<b>Πίνακας 2.1</b> – Ορίζοντας πρόβλεψης για διάφορους τομείς εφαρμογής της πρόβλεψης αιολικής ισχύος. [10].....	11
<b>Πίνακας 3.1</b> – Ενδεικτικό τμήμα της χρονοσειράς ανεμολογικών δεδομένων για το 1 <sup>ο</sup> αιολικό πάρκο (Aeiforiki). .....	13
<b>Πίνακας 4.1</b> – Αξιολόγηση του μοντέλου LinearRegression: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού $R^2$ και μέσο απόλυτο σφάλμα MAE.....	25
<b>Πίνακας 4.2</b> – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου Ridge που εξετάστηκαν. β) Αξιολόγηση του μοντέλου Ridge: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού $R^2$ και μέσο απόλυτο σφάλμα MAE. ....	26
<b>Πίνακας 4.3</b> – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου Lasso που εξετάστηκαν. β) Αξιολόγηση του μοντέλου Lasso: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού $R^2$ και μέσο απόλυτο σφάλμα MAE. ....	27
<b>Πίνακας 4.4</b> – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου ElasticNet που εξετάστηκαν. β) Αξιολόγηση του μοντέλου ElasticNet: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού $R^2$ και μέσο απόλυτο σφάλμα MAE.....	28
<b>Πίνακας 4.5</b> – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου BayesianRidge που εξετάστηκαν. β) Αξιολόγηση του μοντέλου BayesianRidge: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού $R^2$ και μέσο απόλυτο σφάλμα MAE.....	29
<b>Πίνακας 4.6</b> – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου SGDRegressor που εξετάστηκαν. β) Αξιολόγηση του μοντέλου SGDRegressor: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού $R^2$ και μέσο απόλυτο σφάλμα MAE.....	31
<b>Πίνακας 4.7</b> – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου KernelRidge που εξετάστηκαν. β) Αξιολόγηση του μοντέλου KernelRidge: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού $R^2$ και μέσο απόλυτο σφάλμα MAE.....	34
<b>Πίνακας 4.8</b> – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου SVR που εξετάστηκαν. β) Αξιολόγηση του μοντέλου SVR: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού $R^2$ και μέσο απόλυτο σφάλμα MAE. ....	38
<b>Πίνακας 4.9</b> – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου KNeighborsRegressor που εξετάστηκαν. β) Αξιολόγηση του μοντέλου KNeighborsRegressor: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού $R^2$ και μέσο απόλυτο σφάλμα MAE.....	41
<b>Πίνακας 4.10</b> – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου RadiusNeighborsRegressor που εξετάστηκαν. β) Αξιολόγηση του μοντέλου RadiusNeighborsRegressor: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού $R^2$ και μέσο απόλυτο σφάλμα MAE.....	42
<b>Πίνακας 4.11</b> – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου DecisionTreeRegressor που εξετάστηκαν. β) Αξιολόγηση του μοντέλου DecisionTreeRegressor: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού $R^2$ και μέσο απόλυτο σφάλμα MAE.....	45

<b>Πίνακας 4.12</b> – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου RandomForestRegressor που εξετάστηκαν. β) Αξιολόγηση του μοντέλου RandomForestRegressor: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού $R^2$ και μέσο απόλυτο σφάλμα MAE.....	47
<b>Πίνακας 4.13</b> – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου ExtraTreesRegressor που εξετάστηκαν. β) Αξιολόγηση του μοντέλου ExtraTreesRegressor: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού $R^2$ και μέσο απόλυτο σφάλμα MAE.....	49
<b>Πίνακας 4.14</b> – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου AdaBoostRegressor που εξετάστηκαν. β) Αξιολόγηση του μοντέλου AdaBoostRegressor: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού $R^2$ και μέσο απόλυτο σφάλμα MAE.....	51
<b>Πίνακας 4.15</b> – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου GradientBoostingRegressor που εξετάστηκαν. β) Αξιολόγηση του μοντέλου GradientBoostingRegressor: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού $R^2$ και μέσο απόλυτο σφάλμα MAE. ....	54
<b>Πίνακας 4.16</b> – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου MLPRegressor που εξετάστηκαν. β) Αξιολόγηση του μοντέλου MLPRegressor: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού $R^2$ και μέσο απόλυτο σφάλμα MAE.....	58
<b>Πίνακας 4.17</b> (επόμενη σελίδα) – Συγκεντρωτικός πίνακας με τα αποτελέσματα για όλα τα μοντέλα παλινδρόμησης που δοκιμάστηκαν.....	59
<b>Πίνακας 5.1</b> – Προσεγγιστικές τιμές για το μέσο απόλυτο σφάλμα του συνόλου δοκιμής και τον χρόνο εκπαίδευσης των βέλτιστων μοντέλων SVR, KNeighbors και Random Forest, όπως προέκυψαν από την ανάλυση αυτού του κεφαλαίου.....	80
<b>Πίνακας 6.1</b> – Αποτελέσματα προβλέψεων για το 1ο Αιολικό Πάρκο (Aeiforiki): στον πίνακα φαίνονται, για τα έτη 2017 και 2018, οι εκτιμήσεις για την συνολική (ετήσια) παραγόμενη από το πάρκο ισχύ και την απορριπτόμενη ισχύ, όπως προέκυψαν από το βέλτιστο μοντέλο SVR. Δίνεται επίσης η ετήσια ισχύς που διοχετεύεται στο δίκτυο.....	82
<b>Πίνακας 6.2</b> – Αποτελέσματα προβλέψεων για το 2ο Αιολικό Πάρκο (Dei): στον πίνακα φαίνονται, για τα έτη 2017 και 2018, οι εκτιμήσεις για την συνολική (ετήσια) παραγόμενη από το πάρκο ισχύ και την απορριπτόμενη ισχύ, όπως προέκυψαν από το βέλτιστο μοντέλο SVR. Δίνεται επίσης η ετήσια ισχύς που διοχετεύεται στο δίκτυο.....	85
<b>Πίνακας 6.3</b> – Αποτελέσματα προβλέψεων για το 3ο Αιολικό Πάρκο (Diethnis): στον πίνακα φαίνονται, για τα έτη 2017 και 2018, οι εκτιμήσεις για την συνολική (ετήσια) παραγόμενη από το πάρκο ισχύ και την απορριπτόμενη ισχύ, όπως προέκυψαν από το βέλτιστο μοντέλο SVR. Δίνεται επίσης η ετήσια ισχύς που διοχετεύεται στο δίκτυο.....	88
<b>Πίνακας 6.4</b> – Αποτελέσματα προβλέψεων για το 4ο Αιολικό Πάρκο (EuroWind): στον πίνακα φαίνονται, για τα έτη 2017 και 2018, οι εκτιμήσεις για την συνολική (ετήσια) παραγόμενη από το πάρκο ισχύ και την απορριπτόμενη ισχύ, όπως προέκυψαν από το βέλτιστο μοντέλο SVR. Δίνεται επίσης η ετήσια ισχύς που διοχετεύεται στο δίκτυο.....	91
<b>Πίνακας 6.5</b> – Αποτελέσματα προβλέψεων για το 5ο Αιολικό Πάρκο (Atabyrou): στον πίνακα φαίνονται, για τα έτη 2017 και 2018, οι εκτιμήσεις για την συνολική (ετήσια) παραγόμενη από το πάρκο ισχύ και την απορριπτόμενη ισχύ, όπως προέκυψαν από το βέλτιστο μοντέλο SVR. Δίνεται επίσης η ετήσια ισχύς που διοχετεύεται στο δίκτυο.....	94



# ΚΕΦΑΛΑΙΟ 1 – ΕΙΣΑΓΩΓΗ

---

## 1.1 – Αντικείμενο της Εργασίας

Η ανάπτυξη εναλλακτικών, ανανεώσιμων πηγών ενέργειας έχει γίνει επιτακτική ανάγκη τα τελευταία χρόνια, καθώς τα ορυκτά καύσιμα σταδιακά εξαντλούνται και η μόλυνση του περιβάλλοντος ωθεί σε στροφή προς τις λεγόμενες «πράσινες» μορφές ενέργειας οι οποίες είναι πολύ λιγότερο επιβαρυντικές για την ατμόσφαιρα. Η αιολική ενέργεια κατέχει περίοπτη θέση μεταξύ των ανανεώσιμων πηγών ενέργειας. Ο τυχαίος όμως χαρακτήρας του ανέμου και η επακόλουθη δυσκολία της πρόβλεψής του, αποτελούν σημαντικά εμπόδια για την αποτελεσματική αξιοποίηση του αιολικού δυναμικού και την ευρύτερη ανάπτυξη του κλάδου της αιολικής ενέργειας.

Είναι φανερό λοιπόν ότι υπάρχει μεγάλη ανάγκη για αξιόπιστες και ακριβείς ανεμολογικές προβλέψεις στον τομέα την αξιοποίησης της αιολικής ενέργειας για παραγωγή ισχύος. Σκοπός της εργασίας αυτής είναι να γίνει μια ρεαλιστική εκτίμηση της δυνατότητας παραγωγής ισχύος των 5 αιολικών πάρκων της Ρόδου, καθώς και της ετήσιας απορριπτόμενης ισχύος από τα πάρκα αυτά.

Τα δεδομένα που διαθέτουμε είναι οι χρονοσειρές των ανεμολογικών δεδομένων (ταχύτητα και διεύθυνση του ανέμου) για τα έτη 2017 και 2018, οι οποίες αποτελούνται από μετρήσεις που πραγματοποιήθηκαν ανά ώρα σε κάθε πάρκο, για όλη τη διάρκεια των δύο αυτών ετών. Ο διαχειριστής του δικτύου έχει καθορίσει εκ των προτέρων τη μέγιστη ισχύ η οποία θα πρέπει να παρέχεται στο δίκτυο κάθε ώρα, και η οποία εκφράζεται από τη μεταβλητή Set Point που είναι ένα ποσοστό της συνολικής εγκατεστημένης ισχύος του πάρκου και είναι γνωστή. Το πάρκο προγραμματίζεται να λειτουργεί έτσι ώστε κάθε ώρα να μπορεί να δώσει αυτή την ισχύ στο δίκτυο, πχ χρησιμοποιώντας κάποιες από τις ανεμογεννήτριές του για την τροφοδότηση του δικτύου με ισχύ, και τις υπόλοιπες για κάποιον άλλο σκοπό. Η ισχύς που τελικά διοχετεύεται στο δίκτυο κάθε ώρα, η οποία μπορεί να είναι και μικρότερη από τη μέγιστη που έχει καθορίσει εκ των προτέρων ο διαχειριστής (γιατί το δίκτυο μπορεί τελικά να χρειάζεται λιγότερη ισχύ), μετράται στον υποσταθμό που συνδέεται στο πάρκο και είναι γνωστή. Για τις ώρες εκείνες όπου Set Point = 1, ο διαχειριστής έχει ζητήσει όλη την ισχύ που μπορεί να παραχθεί από το πάρκο (η οποία σε ιδανικές συνθήκες ανέμου είναι ίση με την ονομαστική, αλλιώς είναι μικρότερη) και όλη η ισχύς που παράγεται από το πάρκο διοχετεύεται στο δίκτυο. Επομένως, η τιμή της ισχύος που μετράται στον υποσταθμό (και η οποία είναι γνωστή) ταυτίζεται σε αυτή την περίπτωση με τη συνολική παραγόμενη από το πάρκο ισχύ. Σε οποιαδήποτε άλλη περίπτωση, όπου Set Point < 1, δεν γνωρίζουμε τη συνολική ισχύ που παράγεται από το πάρκο, παρά μόνο την τιμή της ισχύος που μετράται στον υποσταθμό και χρησιμοποιείται από το δίκτυο. Σε αυτές τις περιπτώσεις θα επιχειρήσουμε να κάνουμε μια αξιόπιστη εκτίμηση της συνολικής ισχύος του πάρκου, από την οποία ύστερα, αφαιρώντας την ισχύ που χρησιμοποιείται από δίκτυο, προκύπτει η «απορριπτόμενη» ισχύς του πάρκου, δηλαδή η ισχύς που παράγεται αλλά δε χρησιμοποιείται από το δίκτυο. Οι τιμές αυτές θα υπολογιστούν σε ετήσιο επίπεδο,

δηλαδή θα γίνει εκτίμηση της ετήσιας δυνατότητας παραγωγής του πάρκου και της ετήσιας απορριπτόμενης ισχύος του.

Για να μπορέσουν να γίνουν αυτές οι εκτιμήσεις (προβλέψεις) της ισχύος, θα χρησιμοποιηθούν μοντέλα μηχανικής μάθησης (*machine learning*). Η μηχανική μάθηση αποτελεί έναν ραγδαία αναπτυσσόμενο κλάδο της επιστήμης των υπολογιστών, ο οποίος παρέχει πολύπλοκες μεθόδους και αλγορίθμους για τη διεξαγωγή αξιόπιστων προβλέψεων και την εύρεση αλληλοσυσχετίσεων μεταξύ συνόλων δεδομένων. Τα μοντέλα μηχανικής μάθησης αποτελούν ένα πολύ ισχυρό και τεχνολογικά ανεπτυγμένο εργαλείο προβλέψεων, που μπορεί να εφαρμοστεί στην πρόβλεψη της ταχύτητας και της ισχύος του ανέμου. Σε αυτή την εργασία θα εξεταστούν αναλυτικά διάφορα μοντέλα μηχανικής μάθησης και θα αξιολογηθούν ως προς την ικανότητά τους να παρέχουν αξιόπιστες προβλέψεις αιολικής ισχύος. Θα βρεθούν έτσι τα μοντέλα που είναι πιο αποτελεσματικά για την περίπτωση μας, και με χρήση του καλύτερου από αυτά θα εκτιμηθεί η ετήσια απορριπτόμενη ισχύς των 5 αιολικών πάρκων της Ρόδου.

### 1.2 – Δομή της Εργασίας

Η παρούσα διπλωματική εργασία αποτελείται από επτά κεφάλαια, τα οποία περιγράφονται συνοπτικά παρακάτω:

Στο παρόν κεφάλαιο (*Κεφάλαιο 1*) γίνεται μια σύντομη εισαγωγή στο αντικείμενο της εργασίας και περιγράφονται ο σκοπός και η δομή της.

Το *Κεφάλαιο 2* δίνει μια συνοπτική εικόνα του κλάδου της μηχανικής μάθησης (*machine learning*). Περιγράφονται οι βασικές κατηγορίες των προβλημάτων μηχανικής μάθησης, με έμφαση στα προβλήματα επιβλεπόμενης μάθησης, και δίνεται μια επισκόπηση των μεθόδων ταξινόμησης και παλινδρόμησης. Τέλος, εξετάζεται η εφαρμογή των μεθόδων μηχανικής μάθησης στον τομέα της πρόβλεψης της ισχύος του ανέμου.

Στο *Κεφάλαιο 3* γίνεται μια επισκόπηση της μεθόδου που θα χρησιμοποιηθεί για την ανάλυση. Περιγράφεται η μορφή των διαθέσιμων χρονοσειρών ανεμολογικών δεδομένων και ο τρόπος με τον οποίο θα γίνει η επεξεργασία τους, προκειμένου να έρθουν σε κατάλληλη μορφή για να χρησιμοποιηθούν σαν είσοδοι στα μοντέλα που θα εκπαιδεύσουμε. Περιγράφεται επίσης αναλυτικά η μέθοδος που θα χρησιμοποιηθεί για την εκπαίδευση των μοντέλων και ο τρόπος με τον οποίο θα γίνει η βέλτιστη ρύθμιση των υπερ-παραμέτρων τους. Τέλος, επιλέγονται τα κριτήρια με τα οποία θα αξιολογηθούν τα μοντέλα.

Στο *Κεφάλαιο 4* πραγματοποιείται η εκπαίδευση των μοντέλων. Εκπαιδεύονται διάφορα μοντέλα παλινδρόμησης, από απλά γραμμικά μοντέλα έως μοντέλα συνόλων (*ensemble methods*) και νευρωνικά δίκτυα, και ρυθμίζονται οι υπερ-παραμέτροί τους στις βέλτιστες τιμές τους. Περιγράφεται ο τρόπος λειτουργίας του κάθε μοντέλου, τα πλεονεκτήματα και τα

## 1. Εισαγωγή

---

μειονεκτημάτά του, καθώς και ο ρόλος των υπερ-παραμέτρων τους. Το κάθε μοντέλο αξιολογείται με κριτήρια το μέσο απόλυτο σφάλμα πρόβλεψης (mean absolute error) και τον συντελεστή προσδιορισμού  $R^2$ , τα οποία αποτελούν τα βασικά κριτήρια για την επιλογή των βέλτιστων μοντέλων.

Το *Κεφάλαιο 5* επικεντρώνεται στα μοντέλα τα οποία κρίθηκαν τα πιο αποτελεσματικά για το πρόβλημά μας: το μοντέλο παλινδρόμησης Διανουσμάτων Υποστήριξης (Support Vector Regressor - SVR) και τα μοντέλα KNeighbors και Random Forest. Κατασκευάζονται οι καμπύλες εκπαίδευσης του κάθε μοντέλου, ώστε να κατανοηθεί ο τρόπος που οι υπερ-παραμέτροί του επηρεάζουν το μέσο απόλυτο σφάλμα πρόβλεψης και τον χρόνο που απαιτείται για την εκπαίδευση του μοντέλου. Στο τέλος επιλέγεται το μοντέλο που παρουσιάζει την καλύτερη συνολική συμπεριφορά, το οποίο για την περίπτωση μας προκύπτει ότι είναι το SVR.

Στο *Κεφάλαιο 6* πραγματοποιούνται οι προβλέψεις για τα 5 αιολικά πάρκα. Συγκεκριμένα, γίνεται μια εκτίμηση, με χρήση του βέλτιστου μοντέλου SVR, της ετήσιας παραγόμενης από το κάθε πάρκο ισχύος, καθώς και της ετήσιας απορριπτόμενης ισχύος, για τα έτη 2017 και 2018. Κατασκευάζονται για κάθε πάρκο γραφικές παραστάσεις που απεικονίζουν τη μεταβολή των μεγεθών αυτών κατά τη διάρκεια του έτους, και εξάγονται συμπεράσματα για τα πάρκα.

Στο *Κεφάλαιο 7*, που αποτελεί τον επίλογο, παρουσιάζονται τα συμπεράσματα που προέκυψαν από τις προσομοιώσεις και τη συνολική έρευνα που πραγματοποιήθηκε στα πλαίσια της μεταπτυχιακής εργασίας. Παρουσιάζονται επίσης κάποια ανοιχτά θέματα και προτάσεις για μελλοντικές εργασίες.



## ΚΕΦΑΛΑΙΟ 2 – ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

---

### 2.1 – Γενικά

Ο σκοπός κάθε επιστήμης είναι η μελέτη των φαινομένων και η κατασκευή μοντέλων τα οποία να περιγράφουν όσο το δυνατόν πληρέστερα τα πειραματικά δεδομένα και τα φαινόμενα με τα οποία σχετίζονται. Με τον όρο *Μηχανική Μάθηση (Machine Learning)* αναφερόμαστε σε προγραμματιστικά μοντέλα (αλγόριθμους) με ρυθμιζόμενες παραμέτρους τα οποία έχουν τη δυνατότητα να εκπαιδεύονται αυτόματα και να μαθαίνουν από τα δεδομένα που τους παρέχονται, ώστε να βελτιώνουν τη συμπεριφορά τους [1]. Ο ορισμός της μηχανικής μάθησης δόθηκε για πρώτη φορά το 1959 από τον *Arthur Samuel*: «Μηχανική Μάθηση είναι το πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί για αυτό» [2]. Ένας πιο επίσημος ορισμός προτάθηκε από τον *Tom M. Mitchell*: «Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από εμπειρία  $E$  ως προς μια κλάση εργασιών  $T$  και ένα μέτρο επίδοσης  $P$ , αν η επίδοσή του σε εργασίες της κλάσης  $T$ , όπως αποτιμάται από το μέτρο  $P$ , βελτιώνεται με την εμπειρία  $E$ » [3].

Ο σκοπός της μηχανικής μάθησης λοιπόν είναι η δημιουργία μοντέλων (αλγόριθμων) που να μαθαίνουν από τα δεδομένα, και πιο συγκεκριμένα, να εντοπίζουν και να αποκαλύπτουν πιθανά κρυμμένα μοτίβα που σχετίζονται με τον μηχανισμό παραγωγής των δεδομένων. Οι πληροφορίες αυτές βοηθούν στη συνέχεια την ανάλυση και την κατανόηση της φύσης των δεδομένων, και μπορούν να χρησιμοποιηθούν ώστε να κάνουμε προβλέψεις για τα φαινόμενα με τα οποία σχετίζονται. Εκτός από τη μοντελοποίηση της κρυμμένης δομής των δεδομένων, ένας σημαντικός κλάδος της μηχανικής μάθησης συνίσταται στην ανάπτυξη αποδοτικών αλγορίθμων για το σχεδιασμό των μοντέλων, την ανάλυση και την πρόβλεψη. Αυτή η πτυχή της μηχανικής μάθησης αποκτά διαρκώς αυξανόμενη σημασία στην εποχή μας, όπου έχουμε να κάνουμε με τεράστιους όγκους δεδομένων τα οποία μπορεί επίσης να περιέχουν σημαντικές ποσότητες θορύβου και τιμών που λείπουν. Η ανάλυση των δεδομένων σε παρόμοιες περιπτώσεις απαιτεί αλγόριθμους οι οποίοι να είναι συγχρόνως υπολογιστικά αποδοτικοί και λειτουργικά εύρωστοι (robust). [4].

Γίνεται λοιπόν κατανοητό ότι η μέθοδος της μηχανικής μάθησης δίνει τη δυνατότητα σε ερευνητές, επιστήμονες, μηχανικούς και αναλυτές δεδομένων να εξάγουν αξιόπιστα συμπεράσματα και να αναδεικνύουν αλληλοσυσχετίσεις μέσω της μάθησης από ιστορικές τάσεις και σχέσεις στα δεδομένα. Για το λόγο αυτό τέτοιες μέθοδοι βρίσκονται εδώ και δεκαετίες στο επίκεντρο της επιστημονικής έρευνας, σε ποικίλους τομείς όπως η Τεχνητή Νοημοσύνη, η Αναγνώριση Προτύπων, η Στατιστική, η Εξόρυξη Δεδομένων, η Επιστήμη Υπολογιστών και η Επεξεργασία και Ανάλυση Σημάτων και Εικόνας.

Τα προβλήματα μηχανικής μάθησης συνήθως ταξινομούνται στις εξής κατηγορίες, ανάλογα με τη φύση του σήματος (δείγματος δεδομένων) με το οποίο τροφοδοτούμε τον αλγόριθμο: την *Επιβλεπόμενη Μάθηση (Supervised Learning)*, την *Μη Επιβλεπόμενη Μάθηση*

(*Unsupervised Learning*), την *Ενισχυτική Μάθηση (Reinforcement Learning)* και την *Ημι-επιβλεπόμενη Μάθηση (Semi-supervised Learning)*.

- Στην **Επιβλεπόμενη Μάθηση (Supervised Learning)** ή μάθηση με επίβλεψη, τα δεδομένα (training set) με τα οποία τροφοδοτούμε τον αλγόριθμο περιλαμβάνουν για κάθε δείγμα (sample) τις εισόδους (input features) και τις αντίστοιχες εξόδους – αποτελέσματα (labels). Στόχος είναι ο αλγόριθμος να εκπαιδευτεί με τα δεδομένα αυτά και να μάθει ένα γενικό κανόνα ώστε να αντιστοιχεί τις εισόδους με τα αποτελέσματα [5]. Κατασκευάζεται έτσι ένα μαθηματικό μοντέλο που μπορεί να χρησιμοποιηθεί για πρόβλεψη: όταν το τροφοδοτήσουμε με ένα καινούριο σύνολο εισόδων, θα μπορεί να προβλέψει τα αποτελέσματα, βασιζόμενο στην εμπειρία που απέκτησε από τα δεδομένα που χρησιμοποιήθηκαν για την εκπαίδευσή του [4]. Τα μοντέλα επιβλεπόμενης μάθησης έχουν ποικίλες εφαρμογές, από την Αναγνώριση Προτύπων και την Οπτική ή Φωνητική Αναγνώριση, μέχρι το Μάρκετινγκ και την Βιοχημεία.
- Στην **Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning)** ή μάθηση χωρίς επίβλεψη, αντίθετα, ο αλγόριθμος δεν έχει πρόσβαση στις μεταβλητές εξόδου (labels), παρά μόνο στα δεδομένα εισόδου (features) κάθε δείγματος. Ο αλγόριθμος εδώ προσπαθεί να βρει τις δομές των δεδομένων εισόδου, δημιουργώντας πιθανώς δικές του κλάσεις εξόδων. Η μη επιβλεπόμενη μάθηση χρησιμοποιείται για να αποκαλύψει κρυμμένα μοτίβα στα δεδομένα με στόχο τη συσταδοποίηση (clustering), δηλαδή την ταξινόμηση των δεδομένων σε κατηγορίες. Ένας βασικός τομέας εφαρμογής της μη επιβλεπόμενης μάθησης είναι η εκτίμηση πυκνότητας (density estimation) σε προβλήματα Στατιστικής.
- Στην **Ενισχυτική Μάθηση (Reinforcement Learning)** το πρόγραμμα πρέπει να πετύχει ένα συγκεκριμένο στόχο και μαθαίνει την επιθυμητή συμπεριφορά μέσω συνεχούς αλληλεπίδρασης με το περιβάλλον και με την ύπαρξη ενός «κριτή» ο οποίος του δίνει θετική ή αρνητική ανατροφοδότηση. Επιλέγει έτσι σε κάθε κατάσταση τη συμπεριφορά που θα μεγιστοποιήσει το κέρδος – ανταμοιβή, ανάλογα με αυτά που έχει μάθει. Η ενισχυτική μάθηση βρίσκει εφαρμογές στην ανάπτυξη στρατηγικής για παιχνίδια, πχ στο σκάκι, και στην αλληλεπίδραση προγραμμάτων με ανθρώπους.
- Μια ακόμα κατηγορία είναι η **Ημι-επιβλεπόμενη Μάθηση (Semi-supervised Learning)**, η οποία είναι κάτι ενδιάμεσο μεταξύ επιβλεπόμενης και μη επιβλεπόμενης μάθησης: σε αυτή την περίπτωση ο αλγόριθμος τροφοδοτείται με ένα σύνολο δεδομένων το οποίο περιέχει δείγματα με εισόδους – εξόδους αλλά και δείγματα (συχνά αρκετά) όπου οι έξοδοι – αποτελέσματα λείπουν. Η πρακτική αξία της ημι-επιβλεπόμενης μάθησης έγκειται στο γεγονός ότι σε πραγματικές εφαρμογές είναι συνήθως πολύ πιο εύκολο να βρεθούν δείγματα χωρίς εξόδους (labels), επομένως ένα μοντέλο ημι-επιβλεπόμενης μάθησης μπορεί να δώσει αποτελέσματα πολύ πιο γρήγορα και με μικρότερο κόστος [6].

## 2.2 – Επιβλεπόμενη Μάθηση

### 2.2.1 – Δημιουργία Συνόλου Εκπαίδευσης

Το βασικό πρώτο βήμα για την ανάπτυξη ενός μοντέλου μηχανικής μάθησης είναι να αποφασίσουμε πώς θα αναπαραστήσουμε τα χαρακτηριστικά ενός φαινομένου με σύνολα δεδομένων (datasets) τα οποία θα μπορούν να χρησιμοποιηθούν από το υπολογιστικό μοντέλο. Η διαδικασία αυτή αποτελεί το *Στάδιο Προεπεξεργασίας (Preprocessing Stage)* των δεδομένων.

Αρχικά πρέπει να αποφασίσουμε ποια χαρακτηριστικά μεγέθη του φαινομένου το περιγράφουν καλύτερα και είναι πιο χρήσιμα για την ανάλυση που πραγματοποιούμε. Επιλέγοντας αυτά τα χαρακτηριστικά, έστω  $l$  σε αριθμό, προκύπτει ένα διάνυσμα:

$$x = \{x_1, x_2, \dots, x_l\}$$

το οποίο ονομάζεται διάνυσμα χαρακτηριστικών και τα  $l$  στοιχεία του αποτελούν τα **χαρακτηριστικά (features)** ενός δείγματος. Το διάνυσμα χαρακτηριστικών αποτελεί την είσοδο (*input*) του μοντέλου. Συχνά αρχίζουμε με ένα μεγάλο αριθμό χαρακτηριστικών, και στην πορεία επιλέγουμε και κρατάμε μόνο τα πιο σχετικά χαρακτηριστικά (δηλαδή αυτά που έχουν το μεγαλύτερο αντίκτυπο στο φαινόμενο που μελετάμε), μέσω μιας διαδικασίας βελτιστοποίησης που είναι γνωστή ως *Επιλογή Χαρακτηριστικών (Feature Selection)* [4].

Αφού επιλέξουμε τις εισόδους του μοντέλου, το επόμενο βήμα σε ένα πρόβλημα επιβλεπόμενης μάθησης είναι η συλλογή των δεδομένων εξόδου για κάθε διάνυσμα εισόδου  $x$ . Κατασκευάζουμε έτσι για κάθε δείγμα το διάνυσμα:

$$y = \{y_1, y_2, \dots, y_k\}$$

κάθε στοιχείο του οποίου ονομάζεται **ετικέτα – στόχος (target label)** και αποτελεί μεταβλητή εξόδου (*output*) του μοντέλου. Οι ετικέτες σε ένα πρόβλημα μηχανικής μάθησης είναι οι ιδιότητες του φαινομένου τις οποίες θέλουμε να προβλέψουμε.

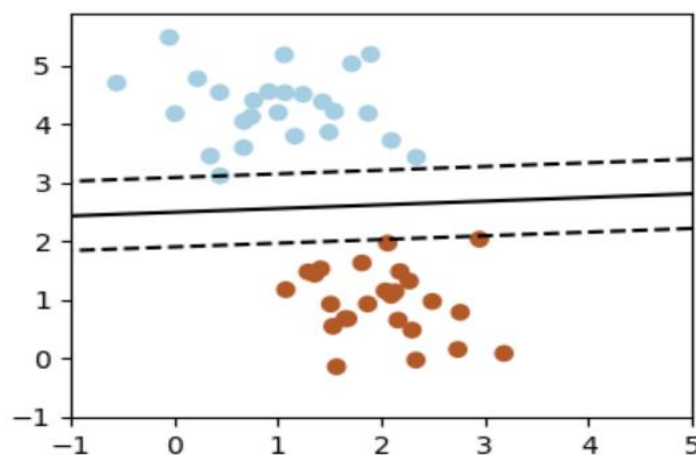
Εφόσον έχουμε αποφασίσει ποια χαρακτηριστικά (features) θα χρησιμοποιήσουμε και ποιες ιδιότητες (labels) θέλουμε να προβλέψουμε, πρέπει να κατασκευάσουμε το **σύνολο εκπαίδευσης (training set)** το οποίο θα χρησιμοποιηθεί για την εκπαίδευση του μοντέλου. Το σύνολο εκπαίδευσης αποτελείται από  $N$  ζεύγη  $(x_n, y_n)$ ,  $n = 1, 2, \dots, N$ , όπου  $N$  είναι ο αριθμός των δειγμάτων (samples) που έχουμε συγκεντρώσει. Για κάθε δείγμα, το οποίο περιγράφεται από το διάνυσμα χαρακτηριστικών του  $x_n$ , είναι γνωστό το διάνυσμα εξόδου του  $y_n$ . Βασιζόμενο στο σύνολο εκπαίδευσης που του παρέχουμε, το μοντέλο εκπαιδεύεται και κατασκευάζει μια συνάρτηση  $y = f(x)$ , μέσω της οποίας θα μπορεί στη συνέχεια να προβλέψει τις επιθυμητές εξόδους – αποτελέσματα  $y$ , αν το τροφοδοτήσουμε με ένα καινούριο διάνυσμα χαρακτηριστικών  $x$  του οποίου το διάνυσμα εξόδου (ετικέτες – στόχοι) μας είναι άγνωστο.

Οι δύο βασικές κατηγορίες των προβλημάτων επιβλεπόμενης μάθησης (supervised learning), ανάλογα με τις τιμές που μπορεί να πάρει η έξοδος, δηλαδή οι ετικέτες – στόχοι (labels) των δειγμάτων, είναι τα προβλήματα *Ταξινόμησης (Classification)* και τα προβλήματα *Παλινδρόμησης (Regression)*.

## 2.2.2 – Ταξινόμηση (Classification)

Στα προβλήματα Ταξινόμησης (Classification) εντάσσονται οι περιπτώσεις επιβλεπόμενης μάθησης που η έξοδος, δηλαδή η ετικέτα-στόχος (label) παίρνει τιμές από ένα διακριτό σύνολο τιμών, για παράδειγμα για ένα πρόβλημα δυαδικής ταξινόμησης η έξοδος μπορεί να πάρει μία από δύο πιθανές τιμές [4]. Ο αλγόριθμος που υλοποιεί την Ταξινόμηση ονομάζεται **Ταξινομητής (Classifier)** και σκοπός του είναι να κατηγοριοποιήσει τα δείγματα στις κατάλληλες κατηγορίες, οι οποίες αντιπροσωπεύονται από τις διακριτές τιμές που μπορεί να πάρει η έξοδος. Έτσι έχουμε τα προβλήματα δυαδικής ταξινόμησης (binary classification) όπου κάθε δείγμα κατηγοριοποιείται σε μία από δύο κατηγορίες, και τα προβλήματα ταξινόμησης πολλαπλών κλάσεων (multiclass classification) όπου κάθε δείγμα πρέπει να κατηγοριοποιηθεί σε μία από αρκετές πιθανές κλάσεις. Ένα παράδειγμα δυαδικής ταξινόμησης είναι τα φίλτρα Spam, τα οποία έχουν σαν εισόδους τα e-mails και οι ετικέτες στις οποίες τα κατηγοριοποιούν είναι “spam” και “non spam”.

Στο Σχήμα 2.1 απεικονίζεται ένα απλό πρόβλημα ταξινόμησης. Οι συντεταγμένες κάθε δείγματος (σημείου στο σχήμα) στους άξονες  $x$ ,  $y$  αποτελούν τα χαρακτηριστικά (features) του δείγματος, και το χρώμα του σημείου στο σχήμα αντιπροσωπεύει την κατηγορία (ετικέτα-στόχο) του δείγματος. Έχουμε δύο διαφορετικά χρώματα, επομένως πρόκειται για πρόβλημα δυαδικής ταξινόμησης. Ένας αλγόριθμος ταξινόμησης που θα εκπαιδευτεί με τα δείγματα του σχήματος μπορεί να χρησιμοποιηθεί για το σχεδιασμό ενός διαχωριστικού ορίου μεταξύ των δύο ομάδων δειγμάτων, όπως φαίνεται στο σχήμα. Έχουμε κατασκευάσει έτσι ένα μοντέλο το οποίο μπορεί να γενικεύσει σε νέα δεδομένα: με την προσθήκη ενός νέου δείγματος το οποίο δεν είναι ακόμη ταξινομημένο, το μοντέλο μπορεί να προβλέψει αν θα έχει κόκκινο ή μπλε χρώμα, ανάλογα με τις συντεταγμένες του και το αν βρίσκεται πάνω ή κάτω από τη διαχωριστική γραμμή που κατασκευάστηκε κατά την εκπαίδευση.



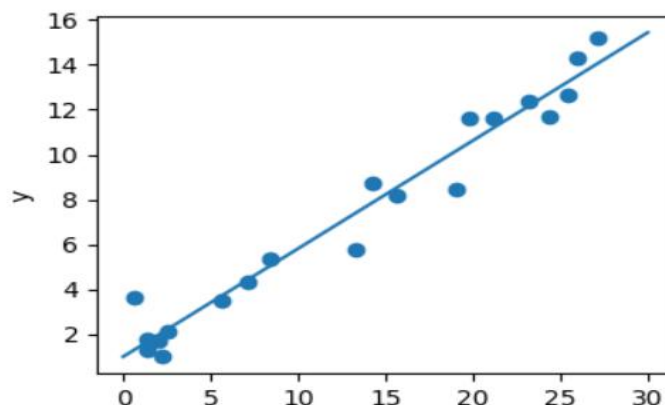
**Σχήμα 2.1** – Ένα απλό πρόβλημα Ταξινόμησης. Ο αλγόριθμος ταξινόμησης μπορεί να προβλέψει το χρώμα ενός νέου δείγματος, ανάλογα με τη θέση του ως προς τη διαχωριστική γραμμή. [7]



## 2.2.3 – Παλινδρόμηση (Regression)

Στα προβλήματα *Παλινδρόμησης (Regression)* ισχύει ότι και στα προβλήματα Ταξινόμησης, με τη διαφορά ότι τώρα η μεταβλητή εξόδου  $y$  δεν παίρνει διακριτές τιμές, αλλά μπορεί να πάρει οποιαδήποτε τιμή μέσα από ένα συνεχές σύνολο τιμών, για να παράδειγμα ένα υποσύνολο των πραγματικών αριθμών  $\mathbb{R}$  [4]. Ο αλγόριθμος που υλοποιεί την Παλινδρόμηση ονομάζεται *Παλινδρομητής (Regressor)* και σκοπός του είναι η εύρεση της συσχέτισης μεταξύ της εξαρτώμενης μεταβλητής εξόδου (label)  $y$  και των ανεξάρτητων μεταβλητών εισόδου (features) των δειγμάτων. Μέσω της παλινδρόμησης καθορίζεται η βέλτιστη συνάρτηση  $y = f(x)$  η οποία μοντελοποιεί με τον καλύτερο τρόπο τις σχέσεις μεταξύ των δεδομένων που έχουν δοθεί. Η συνάρτηση αυτή αποτελεί τη συνάρτηση παλινδρόμησης του προβλήματος. Έχουν αναπτυχθεί πολλές τεχνικές παλινδρόμησης, μεταξύ των οποίων κάποιες είναι παραμετρικές, όπως η γραμμική παλινδρόμηση (linear regression) ή η μέθοδος των ελαχίστων τετραγώνων (ordinary least squares), και κάποιες είναι μη παραμετρικές με την έννοια ότι η συνάρτηση παλινδρόμησης δεν καθορίζεται από ένα σύνολο παραμέτρων, αλλά από ένα σύνολο επιμέρους συναρτήσεων. Οι τεχνικές παλινδρόμησης χρησιμοποιούνται ευρέως σε εφαρμογές πρόβλεψης, όπως για παράδειγμα η πρόβλεψη της θερμοκρασίας σε σχέση με το υψόμετρο και την ατμοσφαιρική πίεση, ή η πρόβλεψη της ζήτησης ενός προϊόντος σε σχέση με τις διαφημιστικές δαπάνες.

Στο *Σχήμα 2.2* απεικονίζεται ένα απλό πρόβλημα παλινδρόμησης. Σε αυτή την περίπτωση, η μεταβλητή εισόδου είναι η συντεταγμένη  $x$  ενός δείγματος και η μεταβλητή εξόδου είναι η συντεταγμένη  $y$ . Πρόκειται ουσιαστικά για ένα πρόβλημα προσαρμογής καμπύλης (curve fitting): το μοντέλο εκπαιδεύεται με τα δείγματα του σχήματος, που αποτελούν το σύνολο εκπαίδευσης (training set), και βρίσκει τη συνάρτηση  $f$  η οποία περιγράφει με τον καλύτερο τρόπο τα δεδομένα αυτά, στη δική μας περίπτωση μια γραμμική συνάρτηση. Εφόσον η συνάρτηση αυτή είναι πλέον γνωστή, με την εμφάνιση ενός νέου δείγματος τετμημένης  $x$  (μεταβλητή εισόδου), το μοντέλο μπορεί πλέον να προβλέψει την τιμή της τεταγμένης  $y$  (μεταβλητή εξόδου), η οποία θα είναι:  $y = f(x)$ .



*Σχήμα 2.2* – Ένα απλό πρόβλημα Παλινδρόμησης. [7]

## 2.3 – Εφαρμογή στην Πρόβλεψη Αιολικής Ισχύος

Η αιολική ενέργεια αποτελεί έναν συνεχώς αναπτυσσόμενο βιομηχανικό τομέα ο οποίος έχει τη δυνατότητα να παρέχει καθαρή, ανανεώσιμη ενέργεια, εκμεταλλευόμενος την ισχύ του ανέμου. Κατά το 2015 στην Ευρωπαϊκή Ένωση ήταν εγκατεστημένα συνολικά 12800 MW αιολικής ισχύος [8]. Στην Ελλάδα ο αριθμός αυτός το 2014 έφτανε τα 4000 MW, με σκοπό το 2020 η εγκατεστημένη αιολική ισχύς να φτάσει τα 7500 MW [9]. Με τις αυξανόμενες απαιτήσεις για αιολική ενέργεια, αυξάνονται και οι απαιτήσεις για αξιόπιστες και ακριβείς προβλέψεις. Σημαντικός παράγοντας είναι επίσης ο ορίζοντας της πρόβλεψης, δηλαδή το χρονικό διάστημα από την παρούσα στιγμή που πραγματοποιείται η πρόβλεψη μέχρι την μελλοντική χρονική στιγμή στην οποία αναφέρεται.

Η πρόβλεψη της αιολικής ισχύος βρίσκει σημαντική εφαρμογή στην εξισορρόπηση του δικτύου. Η διατήρηση της τάσης και της συχνότητας του δικτύου σε σταθερό επίπεδο μπορεί να πραγματοποιηθεί μόνο αν υπάρχει σαφής εικόνα για τη διαθέσιμη εφεδρεία, συνεπώς οι αξιόπιστες προβλέψεις δίνουν τη δυνατότητα στον ρυθμιστή του δικτύου να προγραμματίζει πιο αποτελεσματικά την μελλοντική κατανομή της ισχύος. Ο ρόλος της αποθήκευσης της περισσευούμενης ενέργειας είναι επίσης σημαντικός. Οι ακριβείς προβλέψεις δίνουν τη δυνατότητα να προγραμματιστεί αποτελεσματικά η χρήση των κατάλληλων μπαταριών ή άλλων αποθηκευτικών μέσων, ώστε να μην πηγαίνει χαμένη η περισσευούμενη αιολική ενέργεια και να χρησιμοποιείται όταν υπάρχει αυξημένη ανάγκη σε ισχύ. Σημαντικός είναι επίσης ο ρόλος της πρόβλεψης αιολικής ισχύος και στις αγοραπωλησίες αιολικής ενέργειας. Η ενέργεια πωλείται για μελλοντικές στιγμές στο χρόνο, επομένως είναι επιτακτική η ύπαρξη αξιόπιστων προβλέψεων. Ο ορίζοντας της πρόβλεψης σε αυτές τις περιπτώσεις μπορεί να κυμαίνεται από λίγα λεπτά στο μέλλον (για παράδειγμα 30 λεπτά για τις αγορές ενέργειας) μέχρι και 1 εβδομάδα ή και περισσότερο (για παράδειγμα στον προγραμματισμό εφεδρείας του συστήματος ή τη δημιουργία χρονοδιαγράμματος συντήρησης εγκαταστάσεων παραγωγής ενέργειας). Οι ορίζοντες πρόβλεψης για διάφορους τομείς εφαρμογής φαίνονται στον Πίνακα 2.1. [10]

Οι τεχνικές πρόβλεψης της αιολικής ισχύος μπορούν να χωριστούν σε δύο κατηγορίες. Στην πρώτη ανήκουν οι τεχνικές οι οποίες βασίζονται στην *αριθμητική πρόβλεψη (numerical weather prediction – NWP)*. Οι τεχνικές αυτές στηρίζονται σε μαθηματικά μοντέλα υπολογισμού της αιολικής ισχύος που λαμβάνουν υπόψη τους την κατάσταση της ατμόσφαιρας, την ατμοσφαιρική πίεση και άλλους φυσικούς παράγοντες. Στη δεύτερη κατηγορία, η οποία είναι αυτή με την οποία θα ασχοληθούμε και στην παρούσα εργασία, ανήκουν οι τεχνικές *στατιστικής εκμάθησης (statistical learning)*, οι οποίες στηρίζονται σε ιστορικά ανεμολογικά δεδομένα από παλαιότερες καταγεγραμμένες χρονοσειρές. [11], [12]

Η μηχανική μάθηση βρίσκει εφαρμογή στις τεχνικές πρόβλεψης της δεύτερης κατηγορίας, δηλαδή τις τεχνικές στατιστικής εκμάθησης. Οι χρονοσειρές ανεμολογικών δεδομένων που έχουν συλλεχθεί από παλαιότερες μετρήσεις χρησιμοποιούνται για την εκπαίδευση μοντέλων μηχανικής μάθησης, τα οποία στη συνέχεια μπορούν να πραγματοποιήσουν προβλέψεις της αιολικής ισχύος ή της ταχύτητας του ανέμου. Σε αυτό τον

## 2. Μηχανική Μάθηση

τομέα εφαρμογής της μηχανικής μάθησης έχει γίνει εκτεταμένη έρευνα [13], [14] και έχει δειχθεί ότι μοντέλα όπως η παλινδρόμηση k-NN [15], [16] ή τα νευρωνικά δίκτυα [17], [18] είναι πολύ αποτελεσματικά για την πραγματοποίηση ανεμολογικών προβλέψεων. Επίσης, οι Treiber *et al.* [12] έδειξαν ότι η τεχνική Παλινδρόμησης Διανυσματικής Υποστήριξης (Support Vector Regression – SVR) παράγει πιο ακριβείς προβλέψεις για βραχυπρόθεσμους ορίζοντες πρόβλεψης (έως 3 ώρες) έναντι των αντίστοιχων τεχνικών αριθμητικής πρόβλεψης.

**Πίνακας 2.1** – Ορίζοντας πρόβλεψης για διάφορους τομείς εφαρμογής της πρόβλεψης αιολικής ισχύος. [10]

Horizon	Time Range	Applications
Very short-term	Few seconds – 30 minutes	Market clearing Trading Balancing Virtual Power Plants
Short-term	30 minutes – 6 hours	Load balancing Intraday trading Regulation
Medium-term	6 hours – 1 day	Day-Ahead trading Price optimization
Long-term	1 day – 1 week or more	Planning of reserve energy Scheduling of maintenance



## ΚΕΦΑΛΑΙΟ 3 – ΜΕΘΟΔΟΣ

### 3.1 – Δεδομένα

#### 3.1.1 – Διαθέσιμα Ανεμολογικά Δεδομένα

Στην παρούσα εργασία, όπως προαναφέρθηκε, η ανάλυσή μας θα επικεντρωθεί στα 5 αιολικά πάρκα της Ρόδου. Οι χρονοσειρές των ανεμολογικών δεδομένων που θα χρησιμοποιήσουμε έχουν συσταθεί από μετρήσεις που πραγματοποιήθηκαν σε κάθε πάρκο ανά μία ώρα, για το διάστημα από 1/1/2017 έως 1/1/2019, δηλαδή για χρονικό διάστημα 2 ετών. Επομένως, έχουμε συνολικά 17521 δείγματα για κάθε πάρκο. Στον Πίνακα 3.1 φαίνεται ένα τμήμα της χρονοσειράς ανεμολογικών δεδομένων για το 1<sup>ο</sup> πάρκο (Aeiforiki). Αντίστοιχες είναι οι χρονοσειρές και για τα υπόλοιπα 4 αιολικά πάρκα.

*Πίνακας 3.1 – Ενδεικτικό τμήμα της χρονοσειράς ανεμολογικών δεδομένων για το 1<sup>ο</sup> αιολικό πάρκο (Aeiforiki).*

Date	Wind Speed	Wind Direction	Power	Set Point	Installed Capacity
9/1/2017 12:00	2,50425	167,519	-0,496031	1	4
9/1/2017 13:00	4,16085	134,038	-0,391838	1	
9/1/2017 14:00	5,80418	120,893	-0,0788643	1	
9/1/2017 15:00	7,04339	111,795	0,307621	1	
9/1/2017 16:00	7,58551	111,645	0,522963	1	
9/1/2017 17:00	5,90899	110,059	-0,0151139	1	
9/1/2017 18:00	8,08075	111,619	0,778462	0,998278	
9/1/2017 19:00	9,60944	113,106	1,23541	0,586352	
9/1/2017 20:00	13,3096	126,583	0,896125	0,301357	
9/1/2017 21:00	10,5775	105,893	0,805499	0,269686	
9/1/2017 22:00	8,54353	105,38	0,645125	0,219138	
9/1/2017 23:00	10,2665	117,717	0,555826	0,180054	
10/1/2017 0:00	10,8106	120,886	0,44812	0,144694	
10/1/2017 1:00	13,4926	122,725	0,35046	0,121786	
10/1/2017 2:00	14,229	134,001	0,441063	0,149658	
10/1/2017 3:00	16,0917	122,045	0,391602	0,137867	
10/1/2017 4:00	18,5667	133,145	0,371591	0,132835	
10/1/2017 5:00	14,7359	131,998	0,37591	0,13151	
10/1/2017 6:00	15,1896	138,631	0,437754	0,152201	
10/1/2017 7:00	13,6068	161,697	0,496483	0,173135	
10/1/2017 8:00	12,7425	159,971	0,563849	0,197183	
10/1/2017 9:00	18,8685	150,85	0,635755	0,220742	
10/1/2017 10:00	8,78193	205,789	0,53406	0,266587	
10/1/2017 11:00	11,2703	248,718	0,880142	0,325105	
10/1/2017 12:00	10,764	246,987	0,874779	0,309154	

Όπως φαίνεται στον πίνακα, για κάθε ημερομηνία και ώρα έχουμε την ταχύτητα και τη διεύθυνση του ανέμου. Έχουμε επίσης την τιμή της ισχύος που έχει μετρηθεί στον υποσταθμό που συνδέεται στο πάρκο, και η οποία είναι η ισχύς που χρησιμοποιείται από το δίκτυο. Για τα δείγματα εκείνα που η ισχύς είναι αρνητική, στην πραγματικότητα απλά δεν έχει πραγματοποιηθεί μέτρηση και επομένως μπορούν να αγνοηθούν. Τα Set Points, με τιμές από 0 έως 1, εκφράζουν το ποσοστό της ονομαστικής ισχύος του πάρκου το οποίο έχει εκτιμηθεί από τον διαχειριστή του δικτύου ότι θα είναι η μέγιστη ανάγκη του δικτύου σε ισχύ για την αντίστοιχη ώρα (βλ. και την ανάλυση στην *Ενότητα 1.1*). Για τα δείγματα όπου το Set Point είναι 1, το δίκτυο απορροφά όλη την ισχύ που μπορεί να παραχθεί από το πάρκο, επομένως η μετρημένη τιμή της ισχύος (που φαίνεται στον *Πίνακα 3.1*) ταυτίζεται με τη συνολική παραγόμενη αιολική ισχύ του πάρκου. Για αυτά τα δείγματα η απορριπτόμενη ισχύς είναι μηδενική. Σε όλες τις άλλες περιπτώσεις, όπου το Set Point είναι μικρότερο του 1, η πραγματική παραγόμενη ισχύς του πάρκου είναι μεγαλύτερη από την τιμή του πίνακα, και επομένως έχουμε απόρριψη της περισσευούμενης ισχύος. Τέλος, για κάθε πάρκο γνωρίζουμε και την εγκατεστημένη (ονομαστική) ισχύ του.

#### 3.1.2 – Προεπεξεργασία Δεδομένων

Σύμφωνα με την ανάλυση που προηγήθηκε στην *Ενότητα 2.2.1*, απαραίτητο πρώτο στάδιο πριν ξεκινήσουμε την ανάπτυξη των μοντέλων μηχανικής μάθησης είναι η προεπεξεργασία των διαθέσιμων δεδομένων.

Οι μεταβλητές εισόδου στην περίπτωση μας είναι η ταχύτητα και η ισχύς του ανέμου. Αυτές οι δύο μεταβλητές αποτελούν τα *χαρακτηριστικά (features)* κάθε δείγματος, και επομένως το διάνυσμα χαρακτηριστικών είναι της μορφής:  $x = \{x_1, x_2\}$ . Ως έξοδος σε ένα πρόβλημα μηχανικής μάθησης θεωρείται η μεταβλητή για την οποία θέλουμε να πραγματοποιήσουμε προβλέψεις, δηλαδή στη δική μας περίπτωση η παραγόμενη ισχύς του αιολικού πάρκου. Επομένως έχουμε μόνο μια *ετικέτα-στόχο (label)*, την παραγόμενη ισχύ  $y$ . Εφόσον η μεταβλητή εξόδου είναι η παραγόμενη ισχύς του πάρκου, για την εκπαίδευση των μοντέλων θα πρέπει να χρησιμοποιήσουμε για κάθε πάρκο το υποσύνολο εκείνο της χρονοσειράς δεδομένων για το οποίο η τιμή της μετρημένης ισχύος ταυτίζεται με την παραγόμενη ισχύ του πάρκου, δηλαδή θα πάρουμε μόνο τα δείγματα για τα οποία: *Set Point* = 1. Αυτά τα δείγματα θα χρησιμοποιηθούν για να εκπαιδευτεί ένα μοντέλο, το οποίο στη συνέχεια θα είναι σε θέση να πραγματοποιήσει προβλέψεις για την παραγόμενη ισχύ των δειγμάτων που έχουν *Set Point* < 1.

Τελικά, από την επεξεργασία αυτή προκύπτει ένα σύνολο **8709 δειγμάτων**, για το καθένα από τα οποία γνωρίζουμε τις εισόδους (ταχύτητα και διεύθυνση ανέμου) και την έξοδο (παραγόμενη ισχύς του πάρκου). Ο κώδικας με τον οποίο υλοποιείται η διαδικασία που περιγράφηκε παρατίθεται στο *Παράρτημα Α*.

Παρατηρούμε ότι η περίπτωση μας ανήκει στα προβλήματα *επιβλεπόμενης μάθησης (supervised learning)*, εφόσον για την εκπαίδευση των μοντέλων θα χρησιμοποιηθούν δείγματα για τα οποία είναι γνωστή η τιμή της εξόδου. Επίσης, η μεταβλητή εξόδου (ισχύς) είναι συνεχής

μεταβλητή, η οποία μπορεί να πάρει οποιαδήποτε τιμή σε ένα διάστημα, επομένως η περίπτωση μας αποτελεί ένα πρόβλημα *Παλινδρόμησης (Regression)*, και οι προβλέψεις θα πραγματοποιηθούν από ένα μοντέλο παλινδρόμησης (Regressor).

#### 3.1.3 – Πλαίσιο Υλοποίησης

Για την υλοποίηση όλων των αλγορίθμων και των μοντέλων στην παρούσα εργασία χρησιμοποιήθηκαν τα εξής εργαλεία:

- Περιβάλλον **Anaconda 5.3.0** (<https://www.anaconda.com>)
- **Python 3.7.0** (<https://www.python.org/>)
- **Scikit-Learn 0.19.2** (<https://scikit-learn.org/stable/>)
- **NumPy 1.15.1** (<https://numpy.org>)
- **Pandas 0.23.4** (<https://pandas.pydata.org/>)
- **Matplotlib 2.2.3** (<https://matplotlib.org/index.html>)

Όλα τα παραπάνω αποτελούν λογισμικό ανοικτού κώδικα (open source). Θα θέλαμε να ευχαριστήσουμε όλους όσους έχουν συνεισφέρει στην ανάπτυξη των πακέτων αυτών.

## 3.2 – Μέθοδος Εκπαίδευσης των Μοντέλων Παλινδρόμησης

Στην ενότητα αυτή θα περιγραφεί η μέθοδος με την οποία θα πραγματοποιηθεί η εκπαίδευση των μοντέλων παλινδρόμησης. Στο *Σχήμα 3.2* στο τέλος της ενότητας δίνεται ένα σχεδιάγραμμα όπου απεικονίζεται η πορεία που θα ακολουθήσουμε. Ο κώδικας για τα επιμέρους βήματα που περιγράφονται δίνεται στο *Παράρτημα Α*.

### 3.2.1 – Διαχωρισμός σε Σύνολο Εκπαίδευσης και Σύνολο Δοκιμής

Έπειτα από την ανάλυση που προηγήθηκε, έχουμε απομονώσει από τη συνολική χρονοσειρά του ανέμου τα δείγματα εκείνα τα οποία έχουν *Set Point = 1*. Έχει προκύψει έτσι το διάνυσμα εισόδου του μοντέλου, *X\_data*, το οποίο περιλαμβάνει τα χαρακτηριστικά (features) κάθε δείγματος, δηλαδή ταχύτητα και διεύθυνση του ανέμου, και το διάνυσμα εξόδου, *Y\_data*, το οποίο αποτελείται από την παραγόμενη ισχύ του αιολικού πάρκου για κάθε δείγμα. Τα δύο αυτά σύνολα δεδομένων θα χρησιμοποιηθούν για την *εκπαίδευση* και την *αξιολόγηση* του κάθε μοντέλου παλινδρόμησης.

Ένα σημαντικό ζήτημα στα προβλήματα τεχνικής μάθησης είναι το γεγονός ότι οι επιδόσεις ενός μοντέλου μπορεί να διαφέρουν σημαντικά ανάλογα με την επιλογή του συνόλου δεδομένων που θα χρησιμοποιηθεί για την εκπαίδευση του μοντέλου [4]. Η πρακτική της εκπαίδευσης ενός μοντέλου με ένα σύνολο δεδομένων και έπειτα η αξιολόγηση του μοντέλου πάνω στο ίδιο σύνολο δεδομένων αποτελεί μεθοδολογικό λάθος: ένα μοντέλο μπορεί να εκπαιδευτεί έτσι ώστε να έχει απόλυτη ακρίβεια στις προβλέψεις του πάνω σε κάποιο συγκεκριμένο σύνολο δεδομένων, αλλά να αποτυγχάνει να εκτελέσει αξιόπιστες προβλέψεις σε οποιοδήποτε άλλο σύνολο δεδομένων που δεν το έχει ξαναδεί. Το μοντέλο δηλαδή έχει πολύ καλή απόδοση στα δεδομένα με τα οποία εκπαιδεύτηκε, αλλά δυσκολεύεται να γενικεύσει σε καινούρια δεδομένα, κάτι το οποίο αποτελεί τον πρωταρχικό σκοπό της μηχανικής μάθησης. Το φαινόμενο αυτό είναι γνωστό ως *Overfitting* [19], [20], [21]. Για την αποφυγή αυτού του προβλήματος, η συνηθισμένη πρακτική σε προβλήματα επιβλεπόμενης μάθησης είναι να χωρίζουμε το αρχικό σύνολο δεδομένων σε δύο επιμέρους σύνολα: το πρώτο από αυτά (και συνήθως το μεγαλύτερο) χρησιμοποιείται για την εκπαίδευση του μοντέλου και ονομάζεται *σύνολο εκπαίδευσης (training set)*, ενώ το δεύτερο χρησιμοποιείται αποκλειστικά για την αξιολόγηση της απόδοσης του μοντέλου και ονομάζεται *σύνολο δοκιμής (test set)* [4], [22]. Προϋπόθεση για την εφαρμογή αυτής της πρακτικής είναι η ύπαρξη ικανοποιητικού αριθμού δειγμάτων, έτσι ώστε το μέγεθος των συνόλων εκπαίδευσης και δοκιμής να είναι επαρκώς μεγάλο, αναφορικά με την πολυπλοκότητα του προβλήματος που αντιμετωπίζουμε.

Επομένως, το επόμενο βήμα είναι να χωρίσουμε τα διανύσματα εισόδου και εξόδου, *X\_data* και *Y\_data* αντίστοιχα, σε σύνολο εκπαίδευσης (*X\_train, y\_train*) και σύνολο δοκιμής (*X\_test, y\_test*). Αυτό θα γίνει με τη βοήθεια της συνάρτησης *train\_test\_split()* [22] του Scikit-learn, η οποία λαμβάνει ως ορίσματα τα αρχικά σύνολα εισόδου και εξόδου και τα χωρίζει με τυχαίο τρόπο (ώστε να εξασφαλιστεί η αμεροληψία του διαχωρισμού) σε σύνολο εκπαίδευσης και σύνολο δοκιμής, διατηρώντας μια αναλογία που ορίζουμε εμείς μεταξύ του



αριθμού των δειγμάτων στα δύο σύνολα. Επιλέγουμε να κάνουμε το διαχωρισμό με αναλογία  $test/train = 0.2$  , δηλαδή το 20% των δειγμάτων (**1742** δείγματα) χρησιμοποιείται για την κατασκευή του συνόλου δοκιμής και το υπόλοιπο 80% (**6967** δείγματα) για το σύνολο εκπαίδευσης.

Είναι σημαντικό αυτά τα σύνολα να κανονικοποιηθούν προτού χρησιμοποιηθούν για την εκπαίδευση μοντέλων: τα περισσότερα μοντέλα μηχανικής μάθησης δεν συμπεριφέρονται με τον επιθυμητό τρόπο όταν τα χαρακτηριστικά (features) των δειγμάτων τους περιέχουν ανωμαλίες. **Κανονικοποίηση (Standardization)** ονομάζεται η μετατροπή ενός συνόλου δεδομένων ώστε η κατανομή τους να μοιάζει λίγο ή πολύ με αυτή της κανονικής κατανομής. Υπάρχουν διάφοροι τρόποι κανονικοποίησης ενός συνόλου δεδομένων [23]. Εμείς θα χρησιμοποιήσουμε τον αλγόριθμο **MinMaxScaler**( ) του Scikit-learn, ο οποίος κανονικοποιεί τις τιμές των χαρακτηριστικών (features) των δειγμάτων ώστε να κυμαίνονται μεταξύ 0 και 1. Με τον τρόπο αυτό, η μέγιστη απόλυτη τιμή ενός χαρακτηριστικού του δείγματος είναι η μονάδα. Είναι σημαντικό να σημειώσουμε ότι ο ίδιος μετασχηματιστής MinMaxScaler που θα εφαρμοστεί στο διάλυμα χαρακτηριστικών  $X_{train}$  του συνόλου εκπαίδευσης, θα πρέπει να χρησιμοποιηθεί και στο διάλυμα χαρακτηριστικών  $X_{test}$  του συνόλου δοκιμής, έτσι ώστε να έχουμε αξιόπιστα αποτελέσματα κατά την αξιολόγηση. Τα διανύσματα εξόδου (labels) των συνόλων εκπαίδευσης και δοκιμής,  $y_{train}$  και  $y_{test}$  αντίστοιχα, δεν είναι ανάγκη να μετασχηματιστούν.

#### 3.2.2 – Βέλτιστη Ρύθμιση των Υπερ-παραμέτρων (Hyperparameter Tuning)

Ένα από τα πιο σημαντικά στάδια κατά την ανάπτυξη ενός μοντέλου παλινδρόμησης είναι η βέλτιστη ρύθμιση των **υπερ-παραμέτρων** του (**hyperparameter tuning**). Οι υπερ-παραμέτροι ενός μοντέλου μηχανικής μάθησης είναι εκείνες οι παράμετροι του μοντέλου που η τιμή τους πρέπει να ρυθμιστεί με βέλτιστο τρόπο από τον χειριστή του προτού αρχίσει η διαδικασία εκπαίδευσης, σε αντίθεση με τις απλές παραμέτρους του μοντέλου που η βέλτιστη τιμή τους υπολογίζεται αυτόματα από το μοντέλο κατά την εκπαίδευση (όπως οι παράμετροι  $a$  και  $\beta$  σε ένα πρόβλημα γραμμικής παλινδρόμησης  $y = a + \beta x$ ) [24]. Ένα παράδειγμα υπερ-παραμέτρων αποτελούν ο αριθμός των δέντρων ( $n\_estimators$ ) και ο αριθμός των χαρακτηριστικών ( $n\_features$ ) σε έναν αλγόριθμο Random Forest Regression.

Κατά την προσπάθεια ρύθμισης των υπερ-παραμέτρων του μοντέλου ώστε να πετύχουμε τη μεγαλύτερη δυνατή ακρίβεια πρόβλεψης στο σύνολο δοκιμής (test set), εγκυμονεί και πάλι ο κίνδυνος του Overfitting, αυτή τη φορά όμως στο **σύνολο δοκιμής**, καθώς οι υπερ-παραμέτροι μπορούν να πειραχτούν αυθαίρετα από τον χρήστη ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα [25]. Με αυτό τον τρόπο, η γνώση του συνόλου δοκιμής «εισχωρεί» στο μοντέλο, αφαιρώντας του την ικανότητα να γενικεύει σε νέα δεδομένα. Για να λυθεί αυτό το πρόβλημα, ένα υποσύνολο του **συνόλου εκπαίδευσης**, το οποίο ονομάζεται **σύνολο αξιολόγησης (validation set)**, μπορεί σε κάθε στάδιο ρύθμισης των υπερ-παραμέτρων να κρατηθεί επί μέρους για την αξιολόγηση του μοντέλου, αφού αυτό εκπαιδευτεί πρώτα στα υπόλοιπα δείγματα του συνόλου

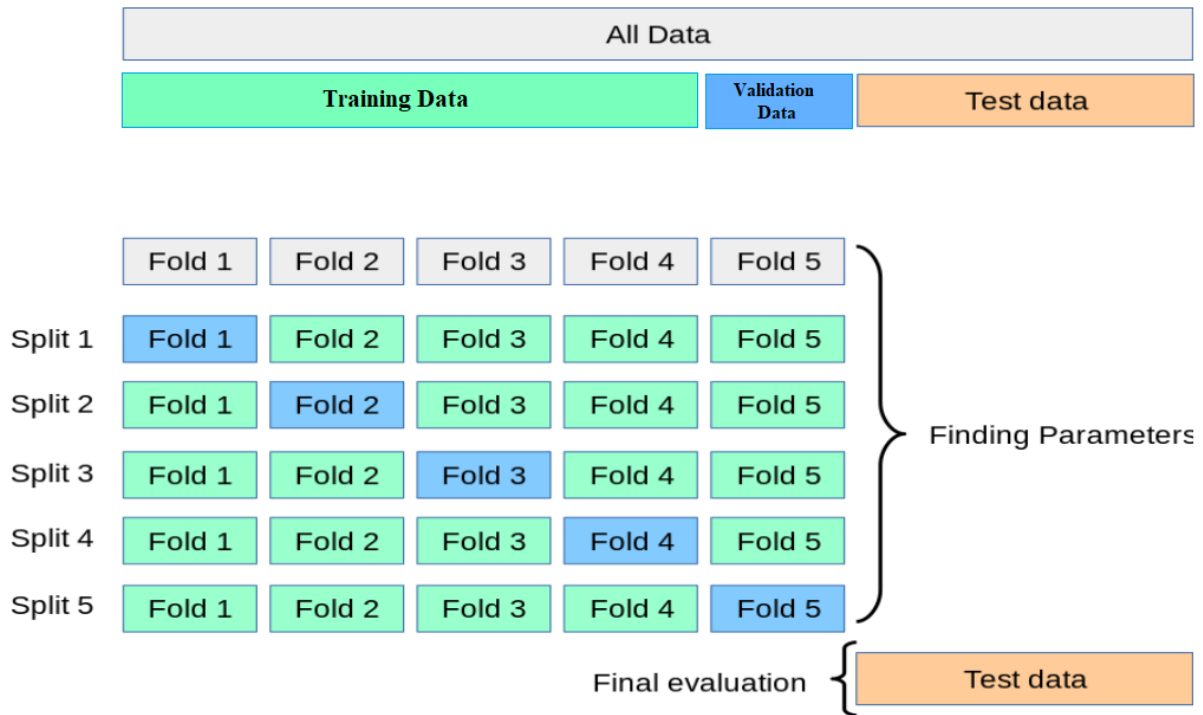
εκπαίδευσης. Στη συνέχεια, αφού έχουν ρυθμιστεί οι υπερ-παράμετροι με βέλτιστο τρόπο, μπορεί να πραγματοποιηθεί μια τελική αξιολόγηση πάνω στο σύνολο δοκιμής. Με τον τρόπο αυτό πετυχαίνουμε μεγαλύτερη αξιοπιστία του μοντέλου, από την άλλη όμως έχουμε ήδη χωρίσει το αρχικό σύνολο δεδομένων σε τρία υποσύνολα (training set, validation set, test set), μειώνοντας έτσι αρκετά τον αριθμό των δειγμάτων που θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου, κάτι που μπορεί να αποτελέσει πρόβλημα στην περίπτωση που τα δεδομένα μας δεν περιλαμβάνουν αρκετά μεγάλο αριθμό δειγμάτων. [22]

Μία λύση για αυτό το πρόβλημα αποτελεί η τεχνική **Cross-Validation** (CV για συντομία) [4], [22]. Με αυτή την τεχνική δε χρειάζεται πλέον να κρατήσουμε ένα τμήμα του συνόλου εκπαίδευσης σαν σύνολο αξιολόγησης. Το σύνολο δοκιμής παρόλα αυτά πρέπει και πάλι να υπάρχει, ώστε να χρησιμοποιηθεί για την τελική αξιολόγηση. Κατά τη βασική πρακτική, που ονομάζεται *k-fold Cross-Validation*, το σύνολο εκπαίδευσης χωρίζεται σε  $k$  υποσύνολα και ακολουθείται η παρακάτω διαδικασία  $k$  φορές:

- Το μοντέλο εκπαιδεύεται χρησιμοποιώντας  $k - 1$  από τα υποσύνολα σαν σύνολο εκπαίδευσης.
- Το μοντέλο που προκύπτει αξιολογείται με το υποσύνολο που απομένει, το οποίο λειτουργεί ως σύνολο αξιολόγησης (validation set) για αυτή την επανάληψη.

Η διαδικασία αυτή απεικονίζεται στο *Σχήμα 3.1* για  $k = 5$  (5-fold Cross-Validation). Η τιμή του  $k$  που θα επιλεγεί εξαρτάται από πολλούς παράγοντες, όμως τυπικές τιμές της είναι της τάξεως του 5 έως 10 [4]. Η συνολική αξιολόγηση του μοντέλου προκύπτει από τη μέση τιμή των επιμέρους αξιολογήσεων που προέκυψαν κατά τις  $k$  επαναλήψεις. Η διαδικασία αυτή μπορεί να επαναληφθεί πολλές φορές, για κάθε τιμή των υπερ-παραμέτρων που θέλουμε να δοκιμάσουμε, ώστε να επιλεγθούν τελικά οι βέλτιστες τιμές. Στη συνέχεια, το βέλτιστο μοντέλο μπορεί να χρησιμοποιηθεί στο σύνολο δοκιμής (test set) για την τελική αξιολόγησή του. Είναι φανερό ότι αυτή η πρακτική έχει μεγάλο υπολογιστικό κόστος και μπορεί να είναι πολύ χρονοβόρα: για τη δοκιμή 10 διαφορετικών τιμών μίας μόνο υπερ-παραμέτρου, χρησιμοποιώντας 5-fold CV, το μοντέλο πρέπει να εκπαιδευτεί και να αξιολογηθεί 50 φορές. Έχει όμως το πλεονέκτημα ότι δε δεσμεύει μεγάλο μέρος των διαθέσιμων δειγμάτων για την αξιολόγηση, κάτι που αποτελεί σημαντικό προτέρημα σε προβλήματα όπου ο αριθμός δειγμάτων είναι περιορισμένος.

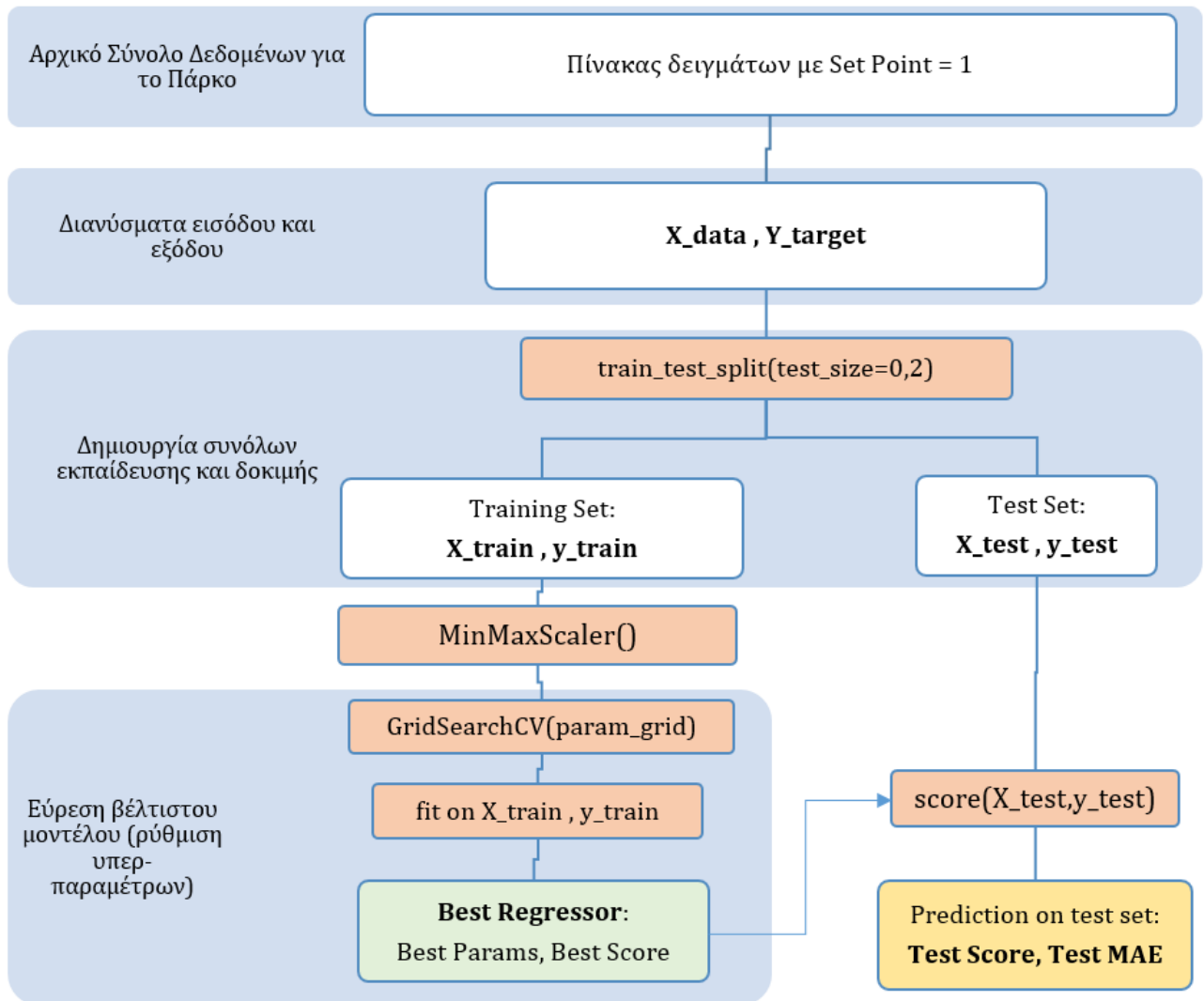
Στην ανάλυση που θα ακολουθήσει θα χρησιμοποιήσουμε κατά κύριο λόγο 5-fold Cross-Validation, δηλαδή  $k = 5$ , καθώς αυτή η τιμή του  $k$  αποτελεί ένα καλό συμβιβασμό μεταξύ αξιοπιστίας της προσομοίωσης και του χρόνου που απαιτείται για την ολοκλήρωσή της. Μεγαλύτερες τιμές του  $k$ , για παράδειγμα  $k = 10$ , οδηγούν για κάποια μοντέλα σε απαγορευτικά υψηλούς χρόνους προσομοίωσης, τουλάχιστον για τα πλαίσια της εργασίας αυτής.



**Σχήμα 3.1** – Τα διαδοχικά στάδια της τεχνικής  $k$ -fold Cross-Validation, για  $k = 5$ . [22]

Για να εκτελέσουμε τη βέλτιστη ρύθμιση των υπερ-παραμέτρων των μοντέλων που θα δοκιμάσουμε, θα χρησιμοποιήσουμε τη μέθοδο **GridSearchCV**( ) του *Scikit-learn*. Ο αλγόριθμος αυτός παίρνει σαν ορίσματα το μοντέλο παλινδρόμησης και τα σύνολα τιμών των υπερ-παραμέτρων του που θέλουμε να δοκιμάσουμε, την τεχνική Cross-Validation που θα εφαρμόσει (πχ 5-fold CV) και τον τρόπο με τον οποίο θέλουμε να γίνει η αξιολόγηση (scoring) του μοντέλου (βλ. *Ενότητα 3.3*). Για κάθε δυνατό συνδυασμό των τιμών των υπερ-παραμέτρων που έχει πάρει σαν όρισμα, ο αλγόριθμος εκτελεί την τεχνική Cross-Validation που έχουμε ορίσει (5-fold Cross-Validation στη περίπτωση μας) και αξιολογεί το συνδυασμό σύμφωνα με το μέτρο αξιολόγησης που ορίσαμε. Αφού εξετάσει όλους τους συνδυασμούς, ο αλγόριθμος επιλέγει το μοντέλο με εκείνες τις υπερ-παραμέτρους που έδωσαν το υψηλότερο Cross-Validation Score. Εξαιρεση από τη διαδικασία αυτή αποτελούν ορισμένα μοντέλα παλινδρόμησης που διαθέτουν δική τους τεχνική εύρεσης των βέλτιστων υπερ-παραμέτρων (όπως οι παραλλαγές *RidgeCV* και *LassoCV* των μοντέλων *Ridge* και *Lasso* αντίστοιχα). Θα χρησιμοποιήσουμε τις εξειδικευμένες αυτές τεχνικές για τα μοντέλα που είναι διαθέσιμες, καθώς δίνουν τη δυνατότητα για πιο γρήγορη και αποτελεσματική βελτιστοποίηση των υπερ-παραμέτρων σε σύγκριση με τη μέθοδο GridSearchCV. [26]

Το τελευταίο στάδιο της διαδικασίας περιλαμβάνει την τελική αξιολόγηση του μοντέλου χρησιμοποιώντας τα δείγματα του *συνόλου δοκιμής (test set)*. Το βέλτιστο μοντέλο που επιλέχθηκε από τον αλγόριθμο GridSearchCV (δηλαδή το μοντέλο με τις βέλτιστες τιμές υπερ-παραμέτρων) εφαρμόζεται στα δεδομένα του συνόλου δοκιμής και αξιολογείται σύμφωνα με τα κριτήρια που θα αναφερθούν στην *Ενότητα 3.3*.



*Σχήμα 3.2 – Συνοπτικό σχεδιάγραμμα της διαδικασίας που περιγράφηκε σε αυτή την ενότητα, για την εκπαίδευση και την αξιολόγηση των μοντέλων παλινδρόμησης.*

### 3.3 - Κριτήρια Αξιολόγησης των Μοντέλων

Από τη στιγμή που έχουμε ρυθμίσει τις υπερ-παραμέτρους και έχουμε κατασκευάσει το βέλτιστο μοντέλο, αυτό που μένει είναι η τελική αξιολόγηση του μοντέλου αυτού πάνω στο σύνολο δοκιμής (*test set*), δηλαδή η αξιολόγηση της ικανότητάς του να πραγματοποιεί ακριβείς προβλέψεις σε καινούρια δεδομένα. Υπάρχουν διάφοροι τρόποι να αξιολογηθεί ένα μοντέλο παλινδρόμησης, όπως το μέσο απόλυτο σφάλμα (*mean absolute error*), το μέσο τετραγωνικό σφάλμα (*mean squared error*) ή ο συντελεστής προσδιορισμού  $R^2$  (*coefficient of determination*) [27]. Η επιλογή του κατάλληλου τρόπου εξαρτάται από τις παραμέτρους του κάθε προβλήματος. Στην ανάλυση που ακολουθεί, θα χρησιμοποιήσουμε κυρίως το μέσο απόλυτο σφάλμα, αλλά και τον συντελεστή προσδιορισμού  $R^2$ .

Το μέσο απόλυτο σφάλμα (*Mean Absolute Error – MAE*) αναφέρεται στις προβλέψεις που γίνονται στο σύνολο δοκιμής (*test set*) και ορίζεται ως η μέση τιμή της απόλυτης διαφοράς μεταξύ της πρόβλεψης που μοντέλου  $\hat{y}_i$ , και της πραγματικής παρατήρησης  $y_i$  (θεωρώντας ότι όλες οι διαφορές έχουν το ίδιο βάρος):

$$MAE = \frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}} |y_i - \hat{y}_i| \quad (3.1)$$

όπου  $n_{samples}$  ο αριθμός των γραμμών (δειγμάτων) του διανύσματος εξόδου [28]. Το μέσο απόλυτο σφάλμα είναι ένα μέτρο του μεγέθους του σφάλματος μιας πρόβλεψης, επομένως, εφόσον μετράει σφάλματα, είναι επιθυμητό να έχει τη χαμηλότερη δυνατή τιμή. Το μέσο απόλυτο σφάλμα δίνει μια πιο καθαρή και αδιαμφισβήτητη εικόνα του μεγέθους του σφάλματος, σε σύγκριση με άλλες τεχνικές όπως η ρίζα του μέσου τετραγωνικού σφάλματος (*root mean squared error – RMSE*), που η τιμή της μπορεί να επηρεάζεται από παράγοντες που δεν έχουν άμεση σχέση με την πρόβλεψη [28], [29]. Για το λόγο αυτό το μέσο απόλυτο σφάλμα χρησιμοποιείται ευρέως για την εκτίμηση των προβλέψεων που βασίζονται σε χρονοσειρές δεδομένων [30]. Επομένως, επιλέγεται και στη δική μας περίπτωση το μέσο απόλυτο σφάλμα σαν βασικό κριτήριο αξιολόγησης των μοντέλων.

Ο συντελεστής προσδιορισμού  $R^2$  (*coefficient of determination*) ορίζεται από την παρακάτω σχέση [27], χρησιμοποιώντας τους ίδιους συμβολισμούς με την Εξίσωση 3.1:

$$R^2 = 1 - \frac{\sum_{i=1}^{n_{samples}} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n_{samples}} (y_i - \bar{y})^2} \quad (3.2)$$

όπου  $\bar{y}$  είναι η μέση τιμή των τιμών των πραγματικών παρατηρήσεων:

$$\bar{y} = \frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}} y_i \quad (3.3)$$

Ο συντελεστής προσδιορισμού αποτελεί ένα μέτρο του πόσο καλά το μοντέλο έχει προσαρμοστεί στα δεδομένα με τα οποία εκπαιδεύτηκε (*goodness of fit*), και συνεπώς του πόσο αποτελεσματικά θα μπορεί να πραγματοποιήσει προβλέψεις σε νέα δεδομένα [31]. Η καλύτερη δυνατή τιμή του συντελεστή προσδιορισμού είναι το 1, που σημαίνει ότι το μοντέλο παλινδρόμησης πραγματοποιεί απόλυτα ακριβείς προβλέψεις στα δεδομένα. Η τιμή του μπορεί επίσης να είναι αρνητική, καθώς ένα μοντέλο μπορεί να είναι αυθαίρετα κακό στις προβλέψεις [27].

Συνοψίζοντας, η διαδικασία που θα ακολουθήσουμε για να αξιολογήσουμε το κάθε μοντέλο παλινδρόμησης έχει ως εξής:

- Αρχικά, για την εύρεση των βέλτιστων υπερ-παραμέτρων, εφαρμόζεται στο μοντέλο η μέθοδος GridSearchCV, όπως αναφέρθηκε στην *Ενότητα 3.2.2*, με κριτήριο αξιολόγησης τη βελτιστοποίηση του *συντελεστή προσδιορισμού  $R^2$* .
- Στη συνέχεια, αφού έχουμε βρει το βέλτιστο μοντέλο (δηλαδή με τις βέλτιστες υπερ-παραμέτρους), το χρησιμοποιούμε για προβλέψεις στο σύνολο δοκιμής (test set) και υπολογίζεται ο *συντελεστής προσδιορισμού  $R^2$*  και το *μέσο απόλυτο σφάλμα (MAE)* των προβλέψεων. Η τιμή του *μέσου απόλυτου σφάλματος* θα είναι το τελικό κριτήριο αξιολόγησης του μοντέλου, και θα αποτελέσει το μέτρο σύγκρισης της απόδοσης μεταξύ των διαφορετικών μοντέλων παλινδρόμησης που θα δοκιμάσουμε.

### 4.1 – Εισαγωγή

Σε αυτό το κεφάλαιο θα πραγματοποιηθούν δοκιμές σε διάφορα μοντέλα παλινδρόμησης, με σκοπό να βρούμε τα πιο κατάλληλα για την περίπτωσή μας. Η διαδικασία που θα ακολουθηθεί για την εκπαίδευση του κάθε μοντέλου είναι αυτή που περιγράφεται στην *Ενότητα 3.2*, δηλαδή θα διαχωριστούν τα διαθέσιμα δείγματα σε σύνολο εκπαίδευσης (training set) και σύνολο δοκιμής (test set), θα κανονικοποιηθούν ώστε οι τιμές τους να κυμαίνονται από 0 έως 1, και στη συνέχεια το σύνολο εκπαίδευσης θα χρησιμοποιηθεί για την εύρεση των βέλτιστων υπερ-παραμέτρων του μοντέλου (hyper-parameter tuning). Το βέλτιστο μοντέλο που θα προκύψει από αυτή τη διαδικασία θα αξιολογηθεί πάνω στο σύνολο δοκιμής, με τον τρόπο που περιγράφεται στην *Ενότητα 3.3*.

Να σημειωθεί ότι ο διαχωρισμός των δειγμάτων σε σύνολο εκπαίδευσης και σύνολο δοκιμής γίνεται με τυχαίο τρόπο, χρησιμοποιώντας τον αλγόριθμο `train_test_split` του Scikit-learn, προκειμένου να αποφευχθεί οποιαδήποτε μεροληπτική τάση στη διαδικασία. Για το λόγο αυτό, ο διαχωρισμός πραγματοποιείται μία φορά μόνο στην αρχή, και τα δύο σύνολα που προκύπτουν αποθηκεύονται και χρησιμοποιούνται στη συνέχεια αυτούσια για όλα τα μοντέλα. Αυτό γίνεται έτσι ώστε όλα τα μοντέλα να εκπαιδεύονται και να αξιολογούνται στα ίδια ακριβώς σύνολα, και να έχει έτσι ουσιαστικό νόημα η σύγκρισή τους (η χρήση βέβαια της τεχνικής Cross-Validation περιορίζει κατά πολύ αυτή την επίδραση, καθώς προσφέρει πιο γενικευμένα αποτελέσματα απαλλαγμένα από μεροληπτικές τάσεις – bias).

Επίσης, όλα τα αποτελέσματα και οι προσομοιώσεις αυτού του κεφαλαίου αναφέρονται στο 1<sup>ο</sup> μόνο αιολικό πάρκο. Σκοπός του κεφαλαίου είναι η διερεύνηση των διαφορών μοντέλων παλινδρόμησης ώστε να ανακαλύψουμε *ποιο προσαρμόζεται καλύτερα στα διαθέσιμα δεδομένα μας και πραγματοποιεί πιο ακριβείς προβλέψεις*. Τα διαθέσιμα ανεμολογικά δεδομένα όλων των πάρκων έχουν την ίδια μορφή (αριθμός και τύπος χαρακτηριστικών-features και ετικετών-labels), και οι τιμές των μεταβλητών τους είναι της ίδιας τάξης μεγέθους, καθώς αναφέρονται σε συγκεκριμένα φυσικά μεγέθη. Επίσης, τα σύνολα δεδομένων και για τα 5 πάρκα έχουν την ίδια ποιότητα (μηδενικός αριθμός missing values – Not A Number). Επομένως είναι ασφαλές να συμπεράνουμε ότι αρκεί να πραγματοποιηθεί η μελέτη στο ένα μόνο από τα πέντε πάρκα, καθώς τα βέλτιστα μοντέλα για αυτό το πάρκο θα είναι τα βέλτιστα και για τα υπόλοιπα 4 πάρκα, και ο σκοπός μας εδώ συνίσταται στη *σύγκριση* μόνο των μοντέλων.

## 4.2 – Μοντέλα Γραμμικής Παλινδρόμησης

Η *Γραμμική Παλινδρόμηση (Linear Regression)* αποτελεί ένα τρόπο μοντελοποίησης της σχέσης μεταξύ μιας μεταβλητής (ή ενός διανύσματος) εξόδου και μίας ή περισσότερων εισόδων (ανεξάρτητες μεταβλητές). Χρησιμοποιείται στις περιπτώσεις που η σχέση αυτή αναμένεται να είναι γραμμική, δηλαδή, χρησιμοποιώντας μαθηματικό συμβολισμό, αν  $\hat{y}$  είναι η πρόβλεψη της τιμής της εξόδου και  $x = (x_1, x_2, \dots, x_p)$  είναι το διάνυσμα των μεταβλητών εισόδου:

$$\hat{y}(w, x) = w_0 + w_1x_1 + \dots + w_px_p \quad (4.1)$$

όπου  $w = (w_0, w_1, \dots, w_p)$  είναι οι *παράμετροι* του μοντέλου γραμμικής παλινδρόμησης. Η παράμετρος  $w_0$  ονομάζεται και *τομή (intercept)*, ενώ οι υπόλοιπες παράμετροι ( $w_1, w_2, \dots, w_p$ ) αποτελούν τους *συντελεστές παλινδρόμησης*.

### 4.2.1 – Μέθοδος Κανονικών Ελαχίστων Τετραγώνων (Ordinary Least Squares)

Η μέθοδος των *Κανονικών Ελαχίστων Τετραγώνων (Ordinary Least Squares Regression)* εκτιμά τις τιμές των παραμέτρων  $w$  του μοντέλου έτσι ώστε να ελαχιστοποιήσει το άθροισμα τετραγώνων των διαφορών μεταξύ των πραγματικών τιμών της μεταβλητής εξόδου και των αντίστοιχων τιμών που προκύπτουν από τη πρόβλεψη του γραμμικού μοντέλου [32]. Με μαθηματικό συμβολισμό, το πρόβλημα έγκειται στην ελαχιστοποίηση της συνάρτησης απωλειών:

$$L(X, y) = \sum_{i=1}^{n_{samples}} (\hat{y}_i - y_i)^2 = \sum_{i=1}^{n_{samples}} (w \cdot X_i - y_i)^2 \quad (4.2)$$

Το μοντέλο παλινδρόμησης του Scikit-learn που υλοποιεί αυτή τη μέθοδο είναι το *linear\_model.LinearRegression* [32]. Το μοντέλο αυτό είναι από τα πιο απλά μοντέλα παλινδρόμησης και δεν διαθέτει υπερ-παραμέτρους για ρύθμιση (οι απλοί παράμετροι του μοντέλου, δηλαδή τα στοιχεία του διανύσματος  $w$ , ρυθμίζονται αυτόματα από το ίδιο το μοντέλο κατά την εκπαίδευσή του). Επομένως δε χρειάζεται σε αυτή την περίπτωση να χρησιμοποιήσουμε τον αλγόριθμο GridSearchCV( ), θα εκτελέσουμε όμως κανονικά Cross-Validation για να έχουμε μια αξιόπιστη εκτίμηση της ικανότητας πρόβλεψης του μοντέλου. Στον Πίνακα 4.1 δίνονται τα αποτελέσματα. Επειδή οι χρόνοι προσομοίωσης για αυτό το μοντέλο είναι σημαντικά μικρότεροι από άλλα, πιο σύνθετα μοντέλα, δοκιμάστηκε εδώ η τεχνική k-fold CV και για  $k = 10$ , εκτός από  $k = 5$ . Από τα αποτελέσματα του πίνακα παρατηρούμε ότι η αξιολόγηση του Cross-Validation ( $R^2$  score) είναι ελαφρώς μικρότερη για  $k = 10$  και βρίσκεται πιο κοντά στο Score του συνόλου δοκιμής, κάτι το οποίο είναι αναμενόμενο γιατί για μεγαλύτερο  $k$  η τεχνική Cross-Validation μας δίνει μια πιο μετριοπαθή εκτίμηση, η οποία βρίσκεται πιο κοντά στην πραγματική τιμή που θα έχει το Score όταν το μοντέλο τροφοδοτηθεί με δεδομένα που δεν έχει ξαναδεί προηγουμένως.



**Πίνακας 4.1** – Αξιολόγηση του μοντέλου *LinearRegression*: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού  $R^2$  και μέσο απόλυτο σφάλμα MAE.

Ordinary Least Squares (Linear Regression)		
	cv = 5	cv = 10
<b>Cross Validation Method:</b>	GridSearchCV(cv=5, scoring='r2')	GridSearchCV(cv=10, scoring='r2')
<b>Fixed Hyper-parameters:</b>	<i>normalize=False, fit_intercept=True</i>	<i>normalize=False, fit_intercept=True</i>
<b>Tuned Hyper-parameters:</b>	–	–
<b>CV Score (<math>R^2</math>):</b>	0,88945248	0,88838
<b>Test Set <math>R^2</math> Score:</b>	0,883054	0,883054
<b>Test Set M.A.E. :</b>	0,2253	0,2253

#### 4.2.2 – Ridge Regression

Η μέθοδος παλινδρόμησης *Ridge Regression* χρησιμοποιεί την ίδια λογική με τη μέθοδο των κανονικών ελαχίστων τετραγώνων, εισάγει όμως και μια παράμετρο ομαλοποίησης *alpha* (*regularization parameter*). Η παράμετρος *alpha* έχει θετική τιμή και ο ρόλος της είναι να μειώνει το πλάτος των συντελεστών παλινδρόμησης  $w$  του γραμμικού μοντέλου, με αποτέλεσμα να μειώνει έτσι τη μεταβλητότητα (variance) των εκτιμήσεων που πραγματοποιεί το μοντέλο και τον κίνδυνο του Overfitting [32], [33]. Ο στόχος τώρα είναι η ελαχιστοποίηση της συνάρτησης απωλειών:

$$L(X, y) = \sum_{i=1}^{n_{samples}} (w \cdot X_i - y_i)^2 + \alpha \cdot \sum_{j=1}^p w_j^2 \quad (4.3)$$

Όταν η παράμετρος *alpha* είναι μηδέν, η εξίσωση 4.3 γίνεται ίδια με την εξίσωση 4.2 και έχουμε το απλό πρόβλημα των κανονικών ελαχίστων τετραγώνων. Όσο η τιμή της παραμέτρου *alpha* αυξάνεται, τόσο περισσότερο μειώνονται τα πλάτη των συντελεστών  $w$  και για πολύ χαμηλές τιμές τους το μοντέλο χάνει την ικανότητα να πραγματοποιεί αξιόπιστες προβλέψεις (φαινόμενο Underfitting). Επομένως, η τιμή της παραμέτρου *alpha* πρέπει να επιλεγθεί προσεχτικά σε μια ενδιάμεση τιμή. Η τεχνική κανονικοποίησης που χρησιμοποιείται από τη *Ridge Regression* ονομάζεται *L2 κανονικοποίηση (L2 regularization)* [32], [34], [35].

Το μοντέλο παλινδρόμησης του Scikit-learn που υλοποιεί αυτή τη μέθοδο είναι το *linear\_model.Ridge* [32]. Η υπερ-πάρμετρος που χρειάζεται ρύθμιση είναι η παράμετρος ομαλοποίησης *alpha*, και θα ρυθμιστεί με χρήση του *RidgeCV*, μιας παραλλαγής του μοντέλου *Ridge* που εκτελεί Cross-Validation για την εύρεση της βέλτιστης τιμής του *alpha*. Ουσιαστικά λειτουργεί με τον ίδιο τρόπο με τον αλγόριθμο *GridSearchCV*, απλά η τεχνική Cross-Validation που χρησιμοποιεί είναι η *GCV* (Generalized Cross-Validation) αντί για *k-fold Cross-Validation*.

#### 4. Δοκιμές Μοντέλων Παλινδρόμησης

Στον Πίνακα 4.2α δίνεται το εύρος των τιμών της παραμέτρου alpha που χρησιμοποιήθηκαν για την αναζήτηση της βέλτιστης τιμής, και στον Πίνακα 4.2β παρουσιάζονται τα αποτελέσματα του μοντέλου.

**Πίνακας 4.2** – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου Ridge που εξετάστηκαν. β) Αξιολόγηση του μοντέλου Ridge: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού  $R^2$  και μέσο απόλυτο σφάλμα MAE.

α)

	Minimum Value	Maximum Value
<b>Hyper-parameter</b>		
alpha	$10^{-6}$	$10^6$

β)

Ridge	
<b>Cross Validation Method:</b>	RidgeCV(cv=GCV, scoring = 'r2')
<b>Fixed Hyper-parameters:</b>	–
<b>Tuned Hyper-parameters:</b>	alpha = 0.01
<b>CV Score (<math>R^2</math>):</b>	0,88967542
<b>Test Set <math>R^2</math> Score:</b>	0,883054
<b>Test Set M.A.E. :</b>	0,2253

#### 4.2.3 – Lasso

Η μέθοδος παλινδρόμησης *Lasso* (*Least Absolute Shrinkage and Selection Operator*) είναι και αυτή μια παραλλαγή της μεθόδου κανονικών ελαχίστων τετραγώνων, η οποία πραγματοποιεί ελάττωση του πλάτους των συντελεστών παλινδρόμησης  $w$  μέσω μιας παραμέτρου ομαλοποίησης  $alpha$  (όπως στο μοντέλο Ridge), αλλά ταυτόχρονα πραγματοποιεί και επιλογή χαρακτηριστικών (*feature selection*), επιλέγοντας τα πιο σημαντικά χαρακτηριστικά του δείγματος και αγνοώντας εκείνα που δεν έχουν σημαντική επίδραση στη πρόβλεψη [36], [32]. Η εξίσωση απωλειών που πρέπει εδώ να ελαχιστοποιηθεί είναι η:

$$L(X, y) = \sum_{i=1}^{n_{samples}} (w \cdot X_i - y_i)^2 + alpha \cdot \sum_{j=1}^p |w_j| \quad (4.4)$$

Για την παράμετρο ομαλοποίησης  $alpha$  ισχύει ότι και στο μοντέλο Ridge. Η διαφορά των δύο μοντέλων είναι ότι το μοντέλο Lasso χρησιμοποιεί *L1 κανονικοποίηση* (*L1 Regularization*), σε αντίθεση με το Ridge που χρησιμοποιεί την τεχνική *L2*. Η κανονικοποίηση *L2* μειώνει τα πλάτη των συντελεστών αλλά δεν τα μηδενίζει. Αντίθετα, η κανονικοποίηση *L1* του Lasso μηδενίζει τους συντελεστές για εκείνα τα χαρακτηριστικά (*features*) που έχουν αμελητέα επίδραση στην

#### 4. Δοκιμές Μοντέλων Παλινδρόμησης

πρόβλεψη, και με αυτό τον τρόπο πραγματοποιεί επιλογή χαρακτηριστικών (feature selection) [34], [35].

Το μοντέλο παλινδρόμησης του Scikit-learn που υλοποιεί αυτή τη μέθοδο είναι το *linear\_model.Lasso* [32]. Η υπερ-παράμετρος που χρειάζεται ρύθμιση είναι και εδώ η παράμετρος ομαλοποίησης  $\alpha$ , και θα ρυθμιστεί με χρήση του *LassoCV*, μιας παραλλαγής του μοντέλου Lasso που εκτελεί Cross-Validation για την εύρεση της βέλτιστης τιμής του  $\alpha$ . Ουσιαστικά λειτουργεί με τον ίδιο τρόπο με τον αλγόριθμο *GridSearchCV*, αλλά η τεχνική Cross-Validation που χρησιμοποιεί είναι η 3-fold CV. Στον Πίνακα 4.3α δίνεται το εύρος των τιμών της παραμέτρου  $\alpha$  που χρησιμοποιήθηκαν για την αναζήτηση της βέλτιστης τιμής, και στον Πίνακα 4.3β παρουσιάζονται τα αποτελέσματα του μοντέλου.

**Πίνακας 4.3** – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου Lasso που εξετάστηκαν. β) Αξιολόγηση του μοντέλου Lasso: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού  $R^2$  και μέσο απόλυτο σφάλμα MAE.

α)

	Minimum Value	Maximum Value
<b>Hyper-parameter</b>		
<i>alpha</i>	$10^{-7}$	$10^4$

β)

Lasso	
<b>Cross Validation Method:</b>	LassoCV(cv = 3)
<b>Fixed Hyper-parameters:</b>	–
<b>Tuned Hyper-parameters:</b>	<i>alpha</i> = $10^{-7}$
<b>CV Score (<math>R^2</math>):</b>	0,88967542
<b>Test Set <math>R^2</math> Score:</b>	0,883054
<b>Test Set M.A.E. :</b>	0,2253

Παρατηρούμε ότι ο συντελεστής προσδιορισμού  $R^2$  και το μέσο απόλυτο σφάλμα MAE για το βέλτιστο μοντέλο Lasso είναι ακριβώς ίδια με τις αντίστοιχες τιμές που βρέθηκαν για το βέλτιστο μοντέλο Ridge.

#### 4.2.4 – Elastic Net

Η μέθοδος *Elastic Net* είναι μια τεχνική γραμμικής παλινδρόμησης που συνδυάζει την L1 κανονικοποίηση του μοντέλου Lasso με την L2 κανονικοποίηση του μοντέλου Ridge [32], [37]. Η εξίσωση απωλειών που πρέπει να ελαχιστοποιηθεί σε αυτή την περίπτωση είναι η:

$$L(X, y) = \sum_{i=1}^{n_{samples}} (w \cdot X_i - y_i)^2 + \alpha \cdot \rho \cdot \sum_{j=1}^p |w_j| + \frac{\alpha \cdot (1 - \rho)}{2} \cdot \sum_{j=1}^p w_j^2 \quad (4.5)$$

Η οποία όπως φαίνεται συνδυάζει γραμμικά τους όρους της L1 και της L2 κανονικοποίησης, με αναλογία που καθορίζεται από το συντελεστή  $\rho$ .

Το μοντέλο παλινδρόμησης του Scikit-learn που υλοποιεί αυτή τη μέθοδο είναι το *linear\_model.ElasticNet* [32]. Η αναλογία L1 προς L2 κανονικοποίησης που θα εφαρμοστεί καθορίζεται από την υπερ-παράμετρο *l1\_ratio*: για *l1\_ratio* = 0 έχουμε αποκλειστικά L2 κανονικοποίηση (Ridge Regression), ενώ για *l1\_ratio* = 1 έχουμε αποκλειστικά L1 κανονικοποίηση (Lasso Regression). Η υπερ-παράμετρος αυτή, μαζί με την παράμετρο ομαλοποίησης *alpha*, πρέπει να ρυθμιστούν στη βέλτιστη τιμή τους με Cross-Validation. Αυτό επιτυγχάνεται με μια παραλλαγή του μοντέλου ElasticNet που ονομάζεται ElasticNetCV, και το οποίο εκτελεί 3-fold CV για να βρει τις βέλτιστες τιμές των υπερ-παραμέτρων *alpha* και *l1\_ratio*. Στον Πίνακα 4.4α δίνεται το εύρος των τιμών των υπερ-παραμέτρων που χρησιμοποιήθηκαν για την αναζήτηση του βέλτιστου μοντέλου, και στον Πίνακα 4.4β παρουσιάζονται τα αποτελέσματα του μοντέλου.

**Πίνακας 4.4** – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου ElasticNet που εξετάστηκαν. β) Αξιολόγηση του μοντέλου ElasticNet: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού  $R^2$  και μέσο απόλυτο σφάλμα MAE.

α)

	Minimum Value	Maximum Value
<b>Hyper-parameter</b>		
<i>alpha</i>	$10^{-7}$	$10^4$
<i>l1_ratio</i>	0.1	$10^{-7}$

β)

ElasticNet	
<b>Cross Validation Method:</b>	ElasticNetCV(cv = 3)
<b>Fixed Hyper-parameters:</b>	–
<b>Tuned Hyper-parameters:</b>	<i>alpha</i> = $10^{-7}$ , <i>l1_ratio</i> = 1
<b>CV Score (<math>R^2</math>):</b>	0,88967542
<b>Test Set <math>R^2</math> Score:</b>	0,883054
<b>Test Set M.A.E. :</b>	0,2253

Παρατηρούμε ότι τα αποτελέσματα είναι και πάλι ίδια με τα βέλτιστα μοντέλα Lasso και Ridge. Παρατηρούμε επίσης ότι η βέλτιστη τιμή του *l1\_ratio* είναι το 1, που σημαίνει ότι το μοντέλο πραγματοποιεί αποκλειστικά L1 κανονικοποίηση και συμπεριφέρεται όπως το μοντέλο Lasso.

## 4.2.5 – Bayesian Ridge Regression

Η μέθοδος της παλινδρόμησης κατά Bayes (*Bayesian Linear Regression*) εφαρμόζει το πλαίσιο της στατιστικής Bayes στη γραμμική παλινδρόμηση: οι συντελεστές παλινδρόμησης  $w$  θεωρείται ότι είναι τυχαίες μεταβλητές οι οποίες πρέπει να εκτιμηθούν από τα διαθέσιμα δείγματα, υποθέτοντας μια γνωστή αρχική κατανομή τους (*prior distribution*) [32], [38]. Με τον τρόπο αυτό το μοντέλο μπορεί να προσαρμόζεται στα εκάστοτε δεδομένα του προβλήματος.

Το μοντέλο παλινδρόμησης του Scikit-learn που θα χρησιμοποιηθεί είναι το *linear\_model.BayesianRidge*, το οποίο έχει την ίδια αρχή λειτουργίας με το κλασσικό μοντέλο Ridge [32]. Οι παράμετροι του μοντέλου *alpha* (παράμετρος ομαλοποίησης) και *lambda* (παράμετρος ακρίβειας για τα βάρη των συντελεστών) υπολογίζονται αυτόματα από το μοντέλο κατά την εκπαίδευσή του στα διαθέσιμα δεδομένα. Οι αρχικές κατανομές των *alpha* και *lambda* θεωρείται ότι είναι κατανομές Γάμμα, οι οποίες χαρακτηρίζονται από μια παράμετρο σχήματος (*shape parameter*) και μια παράμετρο κλίμακας (*scale parameter*). Επομένως, οι υπερ-παράμετροι του μοντέλου που πρέπει να ρυθμιστούν με Cross-Validation είναι οι *alpha\_1, alpha\_2, lambda\_1, lambda\_2* που αποτελούν τις παραμέτρους σχήματος (*alpha\_1, lambda\_1*) και κλίμακας (*alpha\_2, lambda\_2*) της αρχικής κατανομής των *alpha* και *lambda*. Για την εύρεση των βέλτιστων τιμών τους θα χρησιμοποιηθεί ο αλγόριθμος GridSearchCV με 5-fold CV. Στον Πίνακα 4.5α δίνεται το εύρος των τιμών των υπερ-παραμέτρων που χρησιμοποιήθηκαν για την αναζήτηση του βέλτιστου μοντέλου, και στον Πίνακα 4.5β παρουσιάζονται τα αποτελέσματα του μοντέλου.

**Πίνακας 4.5** – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου *BayesianRidge* που εξετάστηκαν. β) Αξιολόγηση του μοντέλου *BayesianRidge*: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού  $R^2$  και μέσο απόλυτο σφάλμα MAE.

α)

	Minimum Value	Maximum Value
<b>Hyper-parameter</b>		
<i>alpha_1, alpha_2</i>	$10^{-10}$	10
<i>lambda_1, lambda_2</i>	$10^{-10}$	10

β)

<b>BayesianRidge</b>	
<b>Cross Validation Method:</b>	GridSearchCV(cv = 5, scoring = 'r2')
<b>Fixed Hyper-parameters:</b>	–
<b>Tuned Hyper-parameters:</b>	<i>alpha_1</i> = $10^{-10}$ , <i>alpha_2</i> = 0.01, <i>lambda_1</i> = 0.01, <i>lambda_2</i> = $10^{-10}$
<b>CV Score (<math>R^2</math>):</b>	0,88945249
<b>Test Set <math>R^2</math> Score:</b>	0,883054
<b>Test Set M.A.E. :</b>	0,2253

Παρατηρούμε ότι η αξιολόγηση στο σύνολο δοκιμής δίνει τα ίδια αποτελέσματα με τα βέλτιστα μοντέλα Lasso, Ridge και ElasticNet. Αξίζει επίσης να σημειωθεί ότι κατά την πραγματοποίηση της έρευνας των βέλτιστων υπερ-παραμέτρων διαπιστώθηκε ότι η τιμή τους δεν επηρεάζει σημαντικά την βαθμολόγηση του μοντέλου.

#### 4.2.6 – Stochastic Gradient Descent (SGD)

Η μέθοδος *Stochastic Gradient Descent (SGD)* είναι μια επαναληπτική μέθοδος παλινδρόμησης, η οποία σε κάθε επανάληψη δοκιμάζει κάποιες τιμές των συντελεστών παλινδρόμησης  $w$ , υπολογίζει την τιμή της συνάρτησης απωλειών, και στη συνέχεια επιλέγει νέες τιμές των συντελεστών ώστε να δίνουν ελαφρώς χαμηλότερη τιμή απωλειών. Επαναλαμβάνοντας αυτή τη διαδικασία αρκετές φορές, προκύπτουν οι τιμές των συντελεστών που ελαχιστοποιούν τη συνάρτηση απωλειών [39]. Η μέθοδος SGD είναι μια απλή και αποδοτική μέθοδος εκπαίδευσης μοντέλων παλινδρόμησης, ιδιαίτερα χρήσιμη για προβλήματα με πολύ μεγάλο αριθμό δειγμάτων (>10000) ή χαρακτηριστικών (features) των δειγμάτων. Για προβλήματα με λιγότερα δείγματα, τεχνικές όπως οι Ridge, Lasso και ElasticNet θεωρούνται πιο κατάλληλες [32]. Τα μειονεκτήματα της μεθόδου είναι ότι περιλαμβάνει έναν αριθμό υπερ-παραμέτρων που πρέπει να ρυθμιστούν, και ότι είναι ευαίσθητη στη μορφή του συνόλου δεδομένων (ομοιομορφία τιμών των χαρακτηριστικών των δειγμάτων κλπ). Επομένως είναι σημαντικό τα δεδομένα να έχουν κανονικοποιηθεί πριν χρησιμοποιηθούν για εκπαίδευση.

Το μοντέλο του Scikit-learn που υλοποιεί τη μέθοδο παλινδρόμησης SGD είναι το *linear\_model.SGDRegressor* [32]. Οι υπερ-παραμέτροι του μοντέλου που πρέπει να ρυθμιστούν είναι οι εξής:

- **loss**: η συνάρτηση απωλειών που θα χρησιμοποιηθεί. Σε αυτή την υπερ-παραμέτρο θα δώσουμε τη σταθερή τιμή 'squared\_loss', που υποδηλώνει τη συνάρτηση απωλειών της συνηθισμένης μεθόδου κανονικών ελαχίστων τετραγώνων.
- **penalty**: ο όρος κανονικοποίησης (regularization term) που θα χρησιμοποιηθεί στη συνάρτηση απωλειών. Οι τιμές που μπορεί να πάρει είναι 'l1', 'l2' και 'elasticnet'.
- **alpha**: η παράμετρος ομαλοποίησης που αναφέρθηκε και στα προηγούμενα μοντέλα (Ridge, Lasso κλπ).
- **l1\_ratio**: καθορίζει την αναλογία L1 προς L2 κανονικοποίησης που θα εφαρμοστεί. Για  $l1\_ratio = 0$  έχουμε L2 κανονικοποίηση (Ridge), για  $l1\_ratio = 1$  έχουμε L1 κανονικοποίηση (Lasso), ενώ για  $0 < l1\_ratio < 1$  έχουμε μίξη των δύο (ElasticNet).
- **max\_iter**: ο μέγιστος αριθμός των επαναλήψεων που θα εκτελεστούν στο σύνολο εκπαίδευσης (epochs).

Για την εύρεση των βέλτιστων τιμών των υπερ-παραμέτρων θα χρησιμοποιηθεί ο αλγόριθμος GridSearchCV με 5-fold CV. Για την απαιτούμενη κανονικοποίηση του συνόλου δεδομένων χρησιμοποιούμε τη μέθοδο *MinMaxScaler* του Scikit-learn (βλ. Ενότητα 3.2.1). Στον Πίνακα 4.6α δίνεται το εύρος των τιμών των υπερ-παραμέτρων που χρησιμοποιήθηκαν για την

#### 4. Δοκιμές Μοντέλων Παλινδρόμησης

αναζήτηση του βέλτιστου μοντέλου, και στον Πίνακα 4.6β παρουσιάζονται τα αποτελέσματα του μοντέλου.

**Πίνακας 4.6** – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου *SGDRegressor* που εξετάστηκαν. β) Αξιολόγηση του μοντέλου *SGDRegressor*: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού  $R^2$  και μέσο απόλυτο σφάλμα *MAE*.

α)

	Minimum Value	Maximum Value
<b>Hyper-parameter</b>		
<i>penalty</i>	'l1', 'l2', 'elasticnet'	
<i>alpha</i>	$10^{-6}$	10
<i>l1_ratio</i>	0	1
<i>max_iter</i>	100	1000

β)

<b>SGDRegressor</b>	
<b>Cross Validation Method:</b>	GridSearchCV(cv = 5, scoring = 'r2')
<b>Fixed Hyper-parameters:</b>	<i>loss</i> = 'squared_loss'
<b>Tuned Hyper-parameters:</b>	<i>penalty</i> = 'l1', <i>alpha</i> = $10^{-5}$ , <i>l1_ratio</i> = 0, <i>max_iter</i> = 1000
<b>CV Score (<math>R^2</math>):</b>	0,88945930
<b>Test Set <math>R^2</math> Score:</b>	0,883059
<b>Test Set M.A.E. :</b>	0,2253

## 4.3 – Παλινδρόμηση Kernel Ridge

### 4.3.1 – Συναρτήσεις Πυρήνα

Η μέθοδος πυρήνα (*kernel trick*) είναι ένας τρόπος για να αντιμετωπίσουμε μη-γραμμικά προβλήματα μοντελοποίησης. Η τεχνική αυτή συνίσταται στη «μεταφορά» των μεταβλητών εισόδου (διάνυσμα  $x$ ) από τον αρχικό χώρο σε ένα χώρο Hilbert υψηλότερων διαστάσεων (ή και άπειρων διαστάσεων), στον οποίο το αρχικό μη-γραμμικό πρόβλημα μετατρέπεται σε γραμμικό [4]. Στο Σχήμα 4.1 φαίνεται η εφαρμογή της τεχνικής αυτής σε ένα απλό πρόβλημα ταξινόμησης.

Η μεταφορά από τον αρχικό χώρο, έστω  $\mathbb{R}^l$ , στο χώρο Hilbert υψηλότερων διαστάσεων, έστω  $\mathbb{R}^K$ , πραγματοποιείται μέσω μιας απεικόνισης:

$$\varphi(x): \mathbb{R}^l \rightarrow \mathbb{R}^K, \quad K > l \quad (4.6)$$

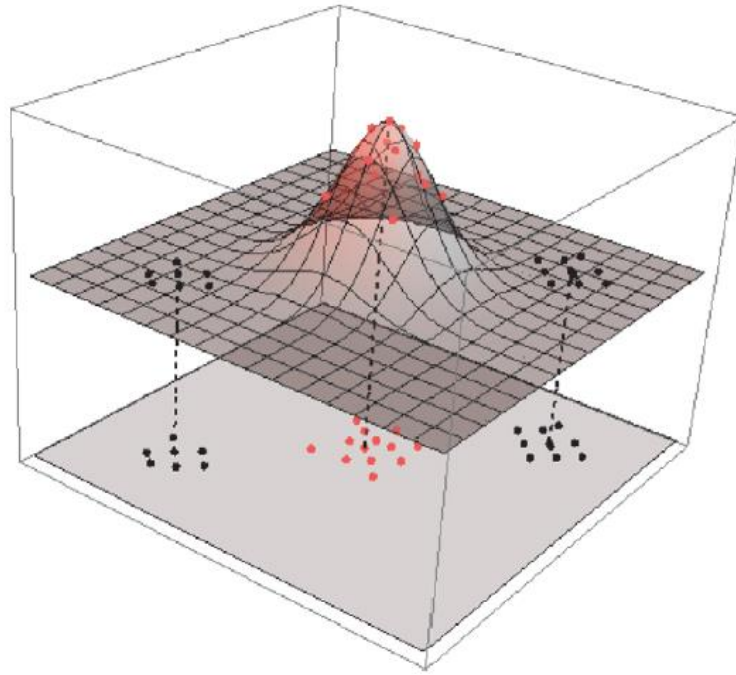
Ο υπολογισμός των  $\varphi(x)$  μπορεί να είναι εξαιρετικά πολύπλοκος, ιδιαίτερα για χώρους πολύ υψηλών διαστάσεων (μεγάλες τιμές του  $K$ ). Για την απλοποίηση των υπολογισμών, η μέθοδος πυρήνα (*kernel trick*) χρησιμοποιεί τη λεγόμενη *συνάρτηση πυρήνα* (*kernel function*)  $k: \mathbb{R}^l \times \mathbb{R}^l \rightarrow \mathbb{R}$  η οποία ορίζεται ως το εσωτερικό γινόμενο δύο οποιονδήποτε διανυσμάτων εισόδου  $x, x' \in \mathbb{R}^l$ :

$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle \quad (4.7)$$

Αν θεωρήσουμε ότι  $x$  είναι μια είσοδος με γνωστή έξοδο και  $x'$  μια καινούρια έξοδος για την οποία θέλουμε να εκτελέσουμε πρόβλεψη, το εσωτερικό γινόμενο  $\langle \varphi(x), \varphi(x') \rangle$  αποτελεί ένα μέτρο της ομοιότητας μεταξύ τους [40]. Ο υπολογισμός των  $\varphi(x), \varphi(x')$  ώστε να υπολογισθεί στη συνέχεια το εσωτερικό τους γινόμενο, μπορεί να είναι πολύ απαιτητικός όπως προαναφέρθηκε. Το εσωτερικό γινόμενο όμως είναι ένας αριθμός (βαθμωτό μέγεθος), επομένως μπορούμε να αποφύγουμε τους πολύπλοκους υπολογισμούς ενός χώρου υψηλών διαστάσεων βρίσκοντας μια κατάλληλη συνάρτηση πυρήνα που να δίνει το ίδιο αποτέλεσμα. Οι συναρτήσεις πυρήνα που συνήθως χρησιμοποιούνται είναι οι εξής [4], [32]:

- Γραμμική (linear):  $\langle x, x' \rangle$
- Πολυωνυμική (polynomial):  $(\gamma \langle x, x' \rangle + r)^d$
- Γκαουσιανή rbf (Radial Basis Function):  $e^{-\gamma \|x - x'\|^2}$
- Σιγμοειδής (sigmoid):  $\tanh(\gamma \langle x, x' \rangle + r)$





**Σχήμα 4.1** – Τεχνική μεταφοράς σε επίπεδο Hilbert υψηλότερων διαστάσεων για ένα απλό πρόβλημα ταξινόμησης: οι κουκκίδες (κόκκινες για τη μία κλάση και μαύρες για την άλλη) δε μπορούν να διαχωριστούν γραμμικά στο αρχικό δισδιάστατο επίπεδο. Μετά τη μεταφορά τους σε ένα τρισδιάστατο επίπεδο, μπορεί εύκολα να σχεδιαστεί ένα επίπεδο που να χωρίζει τις κουκκίδες των δύο κλάσεων. [4]

### 4.3.2 – Kernel Ridge Regression

Η μέθοδος παλινδρόμησης *Kernel Ridge* (*Kernel Ridge Regression*) συνδυάζει την παλινδρόμηση Ridge (γραμμική μέθοδος ελαχίστων τετραγώνων με L2 κανονικοποίηση – Ενότητα 4.2.2) με τη μέθοδο πυρήνα (*kernel trick*) [4], [32].

Το μοντέλο του Scikit-learn που χρησιμοποιείται για την υλοποίηση της μεθόδου παλινδρόμησης Kernel Ridge είναι το `kernel_ridge.KernelRidge` [32]. Η μορφή του μοντέλου είναι παρόμοια με το μοντέλο SVR (Support Vector Regression), χρησιμοποιείται όμως διαφορετική συνάρτηση απωλειών: το μοντέλο KernelRidge χρησιμοποιεί συνάρτηση απωλειών τετραγωνικού σφάλματος, ενώ το SVR χρησιμοποιεί τη συνάρτηση απωλειών ε-insensitive. Και τα δύο μοντέλα εφαρμόζουν L2 κανονικοποίηση. Το μοντέλο KernelRidge είναι πιο κατάλληλο για χρήση σε σύνολα εκπαίδευσης μεσαίου μεγέθους (λιγότερα από 1000 δείγματα), καθώς εκπαιδεύεται πιο γρήγορα. Για μεγαλύτερα σύνολα εκπαίδευσης με πολλά δείγματα, το SVR έχει καλύτερες επιδόσεις. Όσον αφορά το χρόνο που απαιτείται για την πρόβλεψη, το SVR είναι πιο γρήγορο για όλα τα μεγέθη του συνόλου εκπαίδευσης [32].

Οι υπερ-παράμετροι του μοντέλου που πρέπει να ρυθμιστούν είναι οι εξής:

#### 4. Δοκιμές Μοντέλων Παλινδρόμησης

- **alpha**: η παράμετρος ομαλοποίησης του μοντέλου. Έχει την ίδια λειτουργία όπως και στα μοντέλα Ridge και Lasso (μικρές θετικές τιμές της μειώνουν τη μεταβλητότητα των εκτιμήσεων, πολύ υψηλές τιμές οδηγούν σε Overfitting).
- **kernel**: η συνάρτηση πυρήνα που θα χρησιμοποιηθεί από τον αλγόριθμο. Οι συναρτήσεις που θα δοκιμαστούν είναι οι 'linear' (γραμμική), 'rbf' (radial basis function), 'poly' (πολυωνυμική) και 'sigmoid' (σιγμοειδής).
- **gamma**: ο συντελεστής πυρήνα (kernel coefficient) για τους πυρήνες 'rbf', 'poly' και 'sigmoid'. Η αύξηση του συντελεστή αυτού οδηγεί σε καλύτερη προσαρμογή του μοντέλου στα δεδομένα εκπαίδευσης. Πολύ υψηλές τιμές του μπορεί να οδηγήσουν σε Overfitting.
- **coef0**: ο συντελεστής  $\gamma$  για τους πυρήνες 'poly' και 'sigmoid' (βλ. Ενότητα 4.3.1).
- **degree**: ο βαθμός της πολυωνυμικής συνάρτησης που χρησιμοποιείται στον πυρήνα 'poly'.

Η αναζήτηση των βέλτιστων τιμών των υπερ-παραμέτρων θα γίνει με τον αλγόριθμο GridSearchCV χρησιμοποιώντας 5-fold CV. Στον Πίνακα 4.7α δίνεται το εύρος των τιμών των υπερ-παραμέτρων που χρησιμοποιήθηκαν για την αναζήτηση του βέλτιστου μοντέλου, και στον Πίνακα 4.7β παρουσιάζονται τα αποτελέσματα του μοντέλου.

**Πίνακας 4.7** – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου KernelRidge που εξετάστηκαν. β) Αξιολόγηση του μοντέλου KernelRidge: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού  $R^2$  και μέσο απόλυτο σφάλμα MAE.

α)

	Minimum Value	Maximum Value
<b>Hyper-parameter</b>		
alpha	$10^{-3}$	10
kernel	'rbf', 'linear', 'poly', 'sigmoid'	
gamma	$10^{-2}$	1000
coef0	0,1	20
degree	2	8

β)

KernelRidge	
<b>Cross Validation Method:</b>	GridSearchCV(cv = 5, scoring = 'r2')
<b>Fixed Hyper-parameters:</b>	–
<b>Tuned Hyper-parameters:</b>	alpha = 0.001, kernel = 'rbf', gamma = 10
<b>CV Score (<math>R^2</math>):</b>	0,98944644
<b>Test Set <math>R^2</math> Score:</b>	0,98210475
<b>Test Set M.A.E. :</b>	0,039324

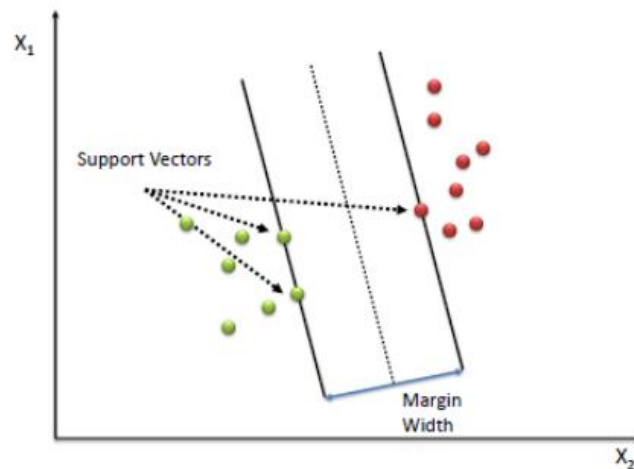
Παρατηρούμε ότι το μοντέλο Kernel Ridge παρουσιάζει σημαντική βελτίωση έναντι του απλού γραμμικού μοντέλου Ridge (Πίνακας 4.2), καθώς έχει μεγαλύτερο συντελεστή προσδιορισμού  $R^2$  και σημαντικά μικρότερο μέσο απόλυτο σφάλμα πρόβλεψης.

## 4.4 – Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines – SVM)

### 4.4.1 – Γενικά

Οι Μηχανές Διανυσμάτων Υποστήριξης (*Support Vector Machines – SVM*) κατασκευάζουν ένα υπερ-επίπεδο ή ένα σύνολο υπερ-επιπέδων σε έναν χώρο υψηλών (ή άπειρων) διαστάσεων, με σκοπό να χρησιμοποιηθεί σε προβλήματα ταξινόμησης ή παλινδρόμησης [4], [32], [41].

Όσον αφορά τα προβλήματα ταξινόμησης, μια Μηχανή Διανυσμάτων Υποστήριξης SVM προσπαθεί να κατασκευάσει το βέλτιστο διαχωριστικό όριο μεταξύ των διαφορετικών κλάσεων, έτσι ώστε να μπορεί στη συνέχεια να ταξινομήσει νέα δείγματα. Αν θεωρήσουμε την απλή περίπτωση ταξινόμησης που φαίνεται στο Σχήμα 4.2, το μοντέλο SVM βρίσκει εκείνο το υπερ-επίπεδο (hyperplane) που διαχωρίζει σωστά όλα τα δείγματα εκπαίδευσης (πράσινες από κόκκινες κουκκίδες) και ταυτόχρονα μεγιστοποιεί την απόσταση (margin) των πλησιέστερων δειγμάτων από αυτό, καθώς γενικά όσο μεγαλύτερο είναι το περιθώριο αυτό τόσο μικρότερη είναι η πιθανότητα να ταξινομηθεί λάθος ένα καινούριο δείγμα [32], [42]. Τα δείγματα που ορίζουν το βέλτιστο υπερ-επίπεδο, δηλαδή αυτά που βρίσκονται πιο κοντά σε αυτό, ονομάζονται *διανύσματα υποστήριξης (support vectors)*.



**Σχήμα 4.2** – Εύρεση του βέλτιστου υπερ-επιπέδου για διαχωρισμό των κλάσεων από μια Μηχανή Διανυσμάτων Υποστήριξης, σε ένα απλό πρόβλημα ταξινόμησης.

Το παραπάνω πρόβλημα ταξινόμησης αποτελεί μια περίπτωση γραμμικά διαχωρίσιμου προβλήματος. Στην πραγματικότητα όμως τα δείγματα δεν είναι σχεδόν ποτέ γραμμικά διαχωρίσιμα, όπως θα γινόταν αν στο Σχήμα 4.2 είχαμε μια πράσινη κουκίδα μεταξύ των κόκκινων. Υπάρχουν δύο τεχνικές για την αντιμετώπιση των μη-γραμμικά διαχωρίσιμων προβλημάτων:

- Η τεχνική *Soft Margin* επιτρέπει σε κάποια δείγματα να ταξινομηθούν λάθος και προσπαθεί να κάνει ένα συμβιβασμό μεταξύ της μεγιστοποίησης του περιθωρίου (margin) του υπερ-επιπέδου και της ελαχιστοποίησης του αριθμού των λάθος ταξινομημένων δειγμάτων. Η

#### 4. Δοκιμές Μοντέλων Παλινδρόμησης

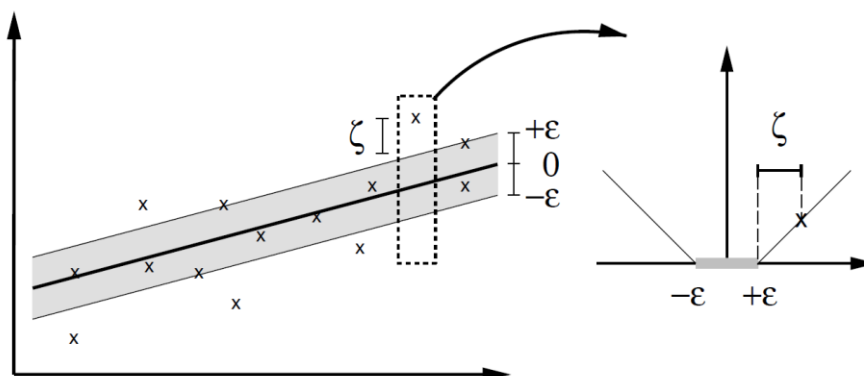
ποινή που επιβάλλεται για τα λάθος ταξινομημένα δείγματα, δηλαδή το πόσο ανεκτικό θα είναι το μοντέλο στη λάθος ταξινόμηση, είναι μια σημαντική παράμετρος του SVM που ονομάζεται *παράμετρος ποινής (penalty parameter)*.

- Υπάρχουν περιπτώσεις όπου η ταξινόμηση δεν μπορεί να γίνει με χρήση ενός γραμμικού υπερ-επιπέδου, αλλά χρειάζεται να σχηματιστεί ένα μη-γραμμικό διαχωριστικό όριο (πχ. μια καμπύλη γραμμή σε ένα δισδιάστατο χώρο). Η τεχνική *Kernel Trick* (βλ. και *Ενότητα 4.3.1*) αποτελεί ένα τρόπο για να μεταφέρουμε τις εισόδους του προβλήματος από τον αρχικό χώρο, όπου το διαχωριστικό όριο είναι μη-γραμμικό, σε ένα χώρο υψηλότερων διαστάσεων όπου ο διαχωρισμός μπορεί να γίνει γραμμικά. Αυτό γίνεται με χρήση μιας κατάλληλης συνάρτησης πυρήνα (kernel function).

Συνδυάζοντας αυτές τις δύο τεχνικές οι Μηχανές Διανυσμάτων Υποστήριξης μπορούν να αντιμετωπίσουν αποτελεσματικά τις μη-γραμμικά διαχωρίσιμες περιπτώσεις ταξινόμησης.

Η μέθοδος των Μηχανών Διανυσμάτων Υποστήριξης μπορεί να επεκταθεί και για χρήση σε προβλήματα παλινδρόμησης (*Support Vector Regression – SVR*) [32], [43]. Όπως και στην περίπτωση της ταξινόμησης, προσπαθούμε να βρούμε ένα βέλτιστο περιθώριο (margin). Τώρα όμως στόχος είναι η πρόβλεψη τιμών και προσπαθούμε να ελαχιστοποιήσουμε μια συνάρτηση απωλειών, η οποία αγνοεί τα σφάλματα πρόβλεψης που βρίσκονται μέσα σε ένα συγκεκριμένο εύρος από την πραγματική τιμή. Αυτό φαίνεται στο *Σχήμα 4.3*: επιλέγουμε ένα εύρος  $\varepsilon$  από κάθε πλευρά του υπερ-επιπέδου, και τα σημεία που βρίσκονται μέσα σε αυτό δεν λαμβάνονται υπόψη από τη συνάρτηση κόστους. Επομένως, το μοντέλο εξαρτάται μόνο από ένα υποσύνολο των δειγμάτων εκπαίδευσης, καθώς λαμβάνονται υπόψη μόνο τα σφάλματα από τα δείγματα που βρίσκονται εκτός του εύρους  $\varepsilon$  και αγνοούνται αυτά που η έξοδός τους βρίσκεται κοντά στην τιμή της πρόβλεψης (το πόσο κοντά καθορίζεται από την επιλογή της παραμέτρου  $\varepsilon$ ). Η συνάρτηση κόστους σε αυτή την περίπτωση ονομάζεται  *$\varepsilon$ -insensitive* και έχει τη μορφή [44]:

$$|\zeta|_{\varepsilon} = \begin{cases} 0 & \text{if } |\zeta| \leq \varepsilon \\ |\zeta| - \varepsilon & \text{otherwise} \end{cases} \quad (4.8)$$



**Σχήμα 4.3** – Ένα απλό παράδειγμα γραμμικής παλινδρόμησης με χρήση Μηχανών Διανυσμάτων Υποστήριξης. [44]

Ένα βασικό πλεονέκτημα των Μηχανών Διανυσμάτων Υποστήριξης είναι ότι χρησιμοποιούν μόνο ένα υποσύνολο των δειγμάτων εκπαίδευσης (τα λεγόμενα διανύσματα υποστήριξης) και επομένως είναι πιο αποδοτικές από άποψη απαιτήσεων μνήμης. Επίσης μπορούν να χρησιμοποιηθούν για ένα μεγάλο εύρος γραμμικών ή μη-γραμμικών προβλημάτων, με χρήση των κατάλληλων συναρτήσεων πυρήνα. Είναι επίσης πολύ αποδοτικές σε χώρους υψηλών διαστάσεων καθώς και σε περιπτώσεις που ο αριθμός των διαστάσεων είναι μεγαλύτερος από τον αριθμό των δειγμάτων. [32]

#### 4.4.2 – Support Vector Regression (SVR)

Το μοντέλο του Scikit-learn που υλοποιεί την Παλινδρόμηση Διανυσμάτων Υποστήριξης (Support Vector Regression) είναι το *svm.SVR* [32]. Οι υπερ-παράμετροι του μοντέλου που πρέπει να ρυθμιστούν είναι οι εξής:

- **kernel**: η συνάρτηση πυρήνα που θα χρησιμοποιηθεί από τον αλγόριθμο. Οι συναρτήσεις που θα δοκιμαστούν είναι οι ‘linear’ (γραμμική) και ‘rbf’ (radial basis function).
- **gamma**: ο συντελεστής πυρήνα (kernel coefficient) για τον πυρήνα ‘rbf’. Η αύξηση του συντελεστή αυτού οδηγεί σε καλύτερη προσαρμογή του μοντέλου στα δεδομένα εκπαίδευσης. Πολύ υψηλές τιμές του μπορεί να οδηγήσουν σε Overfitting.
- **C**: η παράμετρος ποινής (penalty parameter) για τον όρο σφάλματος (βλ. Ενότητα 4.4.1).
- **epsilon**: το εύρος  $\epsilon$  που ορίζει την περιοχή μέσα στην οποία δεν λαμβάνονται υπόψη από τη συνάρτηση κόστους τα σφάλματα πρόβλεψης (βλ. Ενότητα 4.4.1).

Οι επιδόσεις των Μηχανών Διανυσματικής Υποστήριξης και η ακρίβεια των προβλέψεών τους εξαρτάται σε μεγάλο βαθμό από τη σωστή ρύθμιση των υπερ-παραμέτρων του:

Η υπερ-παράμετρος **gamma** καθορίζει το πόσο μεγάλη επίδραση θα έχει το κάθε συγκεκριμένο δείγμα εκπαίδευσης. Όσο υψηλότερη είναι η τιμή του, τόσο πιο κοντά του θα πρέπει να είναι τα άλλα δείγματα για να επηρεαστούν. Η υπερ-παράμετρος gamma μπορεί να θεωρηθεί ως το αντίστροφο της ακτίνας επιρροής των δειγμάτων που έχουν επιλεγεί από το μοντέλο ως διανύσματα υποστήριξης (support vectors). Για πολύ υψηλές τιμές του gamma η σφαίρα επιρροής ενός διανύσματος υποστήριξης περιλαμβάνει μόνο αυτό το ίδιο και δε μπορούμε να αποφύγουμε το Overfitting. [32]

Η υπερ-παράμετρος **C** (παράμετρος ποινής) καθορίζει το συμβιβασμό μεταξύ της πολυπλοκότητας του μοντέλου και του βαθμού στον οποίο αποκλίσεις (σφάλματα) μεγαλύτερες από  $\epsilon$  γίνονται ανεκτά. Όσο πιο μεγάλη γίνεται η τιμή του C τόσο πιο πολύ αυξάνεται η πολυπλοκότητα του μοντέλου αλλά μειώνεται η ανοχή του στα σφάλματα πρόβλεψης. Η υπερ-παράμετρος C είναι αντίστοιχη με την παράμετρο ομαλοποίησης alpha των μοντέλων γραμμικής παλινδρόμησης που εξετάστηκαν προηγουμένως. Η ακριβής σχέση τους εξαρτάται από το εκάστοτε μοντέλο, για παράδειγμα για την παράμετρο alpha του μοντέλου Ridge η σχέση είναι:

$$C = \frac{1}{\text{alpha}} \quad (4.9)$$

#### 4. Δοκιμές Μοντέλων Παλινδρόμησης

Η υπερ-παράμετρος  $\epsilon$  καθορίζει το εύρος της ζώνης ( $\epsilon$ -insensitive zone) μέσα στην οποία τα σφάλματα δεν λαμβάνονται υπόψη. Η τιμή του  $\epsilon$  επηρεάζει τον αριθμό των διανυσμάτων υποστήριξης (δείγματα εκτός της ζώνης) που θα χρησιμοποιηθούν για την κατασκευή της συνάρτησης παλινδρόμησης: όσο πιο μεγάλη η τιμή του  $\epsilon$  τόσο λιγότερα διανύσματα υποστήριξης θα επιλεγθούν.

Για την εύρεση των βέλτιστων τιμών των υπερ-παραμέτρων θα χρησιμοποιηθεί ο αλγόριθμος GridSearchCV με 5-fold CV. Για την απαιτούμενη κανονικοποίηση του συνόλου δεδομένων χρησιμοποιούμε τη μέθοδο *MinMaxScaler* του Scikit-learn (βλ. Ενότητα 3.2.1). Στον Πίνακα 4.8α δίνεται το εύρος των τιμών των υπερ-παραμέτρων που χρησιμοποιήθηκαν για την αναζήτηση του βέλτιστου μοντέλου, και στον Πίνακα 4.8β παρουσιάζονται τα αποτελέσματα του μοντέλου.

**Πίνακας 4.8** – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου SVR που εξετάστηκαν. β) Αξιολόγηση του μοντέλου SVR: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού  $R^2$  και μέσο απόλυτο σφάλμα MAE.

α)

	Minimum Value	Maximum Value
<b>Hyper-parameter</b>		
<i>kernel</i>	'rbf', 'linear'	
<i>gamma</i>	$10^{-3}$	100
<i>C</i>	$10^{-3}$	1000
<i>epsilon</i>	0	1

β)

SVR	
<b>Cross Validation Method:</b>	GridSearchCV(cv = 5, scoring = 'r2')
<b>Fixed Hyper-parameters:</b>	–
<b>Tuned Hyper-parameters:</b>	<i>kernel</i> = 'rbf', <i>gamma</i> = 50, <i>C</i> = 100, <i>epsilon</i> = 0.01
<b>CV Score (<math>R^2</math>):</b>	0,98922228
<b>Test Set <math>R^2</math> Score:</b>	0,98097841
<b>Test Set M.A.E. :</b>	0,037046

## 4.5 – Μοντέλα Κοντινότερων Γειτόνων

### 4.5.1 – Γενικά

Τα μοντέλα *Κοντινότερων Γειτόνων* (*Nearest Neighbors*) μπορούν να χρησιμοποιηθούν για ταξινόμηση και παλινδρόμηση. Η βασική ιδέα πίσω από αυτά τα μοντέλα είναι η εύρεση ενός προκαθορισμένου αριθμού των δειγμάτων εκπαίδευσης που βρίσκονται σε απόσταση πιο κοντά στο νέο δείγμα, και η πρόβλεψη της εξόδου του (label) από τα δείγματα αυτά [32]. Ο αριθμός των δειγμάτων που θα χρησιμοποιηθούν μπορεί να είναι μια σταθερά που καθορίζεται από το χρήστη, όπως στα μοντέλα *k-κοντινότερων γειτόνων* (*k-nearest neighbors*) [45], ή να ποικίλει ανάλογα με το πόσα δείγματα βρίσκονται σε μια καθορισμένη απόσταση από το προς πρόβλεψη δείγμα, όπως στα *ακτινικά μοντέλα γειτόνων* (*radius-based neighbor models*) [46]. Η απόσταση αυτή μπορεί να είναι οποιοδήποτε μετρητικό μέγεθος, όμως η κανονική ευκλείδεια απόσταση είναι η πιο συνηθισμένη επιλογή. Οι μέθοδοι κοντινότερων γειτόνων είναι γνωστές ως μέθοδοι μη-γενίκευσης (non-generalizing / instance-based methods), καθώς δεν προσπαθούν να κατασκευάσουν ένα γενικευμένο εσωτερικό μοντέλο, αλλά απλώς αποθηκεύουν («θυμούνται») όλα τα δεδομένα εκπαίδευσης.

Μια χρήσιμη τεχνική της μεθόδου κοντινότερων γειτόνων, που μπορεί να εφαρμοστεί τόσο στην ταξινόμηση όσο και στην παλινδρόμηση, είναι η απόδοση κάποιου βάρους στη συνεισφορά του κάθε δείγματος-γείτονα, έτσι ώστε οι κοντινότεροι γείτονες να συνεισφέρουν περισσότερο στην πρόβλεψη από ότι οι μακρινότεροι [32], [47]. Για παράδειγμα, μια συνηθισμένη πρακτική είναι η απόδοση σε κάθε γείτονα βάρους ίσου με  $1/d$ , όπου  $d$  είναι η απόστασή του από το προς πρόβλεψη δείγμα.

Υπάρχουν διάφοροι αλγόριθμοι για τον υπολογισμό των κοντινότερων γειτόνων του προς πρόβλεψη δείγματος. Ο πιο απλός από αυτούς είναι ο αλγόριθμος *brute-force*, ο οποίος υπολογίζει τις αποστάσεις όλων των ζευγών δειγμάτων και επιλέγει αυτά που βρίσκονται πιο κοντά. Αυτός ο αναλυτικός υπολογισμός όμως γίνεται μη-βιώσιμος όταν ο αριθμός των δειγμάτων είναι μεγάλος. Για να αντιμετωπιστεί αυτή η αδυναμία του αλγορίθμου *brute-force*, έχουν αναπτυχθεί διάφορες δομές με μορφή δέντρου, οι οποίες έχουν σαν σκοπό να μειώσουν τους απαιτούμενους υπολογισμούς αποστάσεων. Η βασική ιδέα είναι ότι αν ένα σημείο  $A$  βρίσκεται πολύ μακριά από το σημείο  $B$ , και το σημείο  $B$  βρίσκεται κοντά σε ένα άλλο σημείο  $\Gamma$ , τότε γνωρίζουμε ότι και τα σημεία  $A$  και  $\Gamma$  είναι πολύ μακριά μεταξύ τους, χωρίς να χρειάζεται να υπολογίσουμε τη μεταξύ τους απόσταση. Οι αλγόριθμοι που κάνουν χρήση αυτής της ιδέας είναι ο αλγόριθμος *K-D Tree* [48], ο οποίος χρησιμοποιεί τη δομή *K-D Tree*, και ο αλγόριθμος *Ball Tree* [49], ο οποίος χρησιμοποιεί τη δομή *Ball Tree* που αναπτύχθηκε για να αντιμετωπίσει τις αδυναμίες της δομής *K-D Tree* σε υψηλές διαστάσεις. Η επιλογή του κατάλληλου αλγορίθμου εξαρτάται από ένα πλήθος παραγόντων [32], όπως ο αριθμός των δειγμάτων  $N$  και των χαρακτηριστικών (features), η δομή των συνόλων δεδομένων που διαθέτουμε και ο αριθμός  $k$  των γειτόνων που απαιτούνται για ένα σημείο (όταν το  $k$  γίνεται μεγάλο σε σύγκριση με τον αριθμό δειγμάτων  $N$ , η απόδοση των δενδροειδών μειώνεται και ο αλγόριθμος *brute-force* είναι πιο κατάλληλος).

#### 4.5.2 – KNeighbors Regressor

Η παλινδρόμηση που βασίζεται στην τεχνική *k*-κοντινότερων γειτόνων (*k*-nearest neighbors) χρησιμοποιεί τα *k* δείγματα του συνόλου εκπαίδευσης που βρίσκονται πιο κοντά στο προς πρόβλεψη σημείο, όπου *k* είναι ένα αριθμός που ορίζεται από το χρήστη. Η έξοδος (label) του προς πρόβλεψη σημείου υπολογίζεται από τον μέσο όρο των (γνωστών) εξόδων αυτών των δειγμάτων-γειτόνων.

Το μοντέλο του Scikit-learn που υλοποιεί την τεχνική *k*-κοντινότερων γειτόνων είναι το *neighbors.KNeighborsRegressor* [32]. Οι υπερ-παράμετροι του μοντέλου που πρέπει να ρυθμιστούν είναι οι εξής:

- **algorithm**: ο αλγόριθμος που θα χρησιμοποιηθεί για την εύρεση των κοντινότερων γειτόνων. Οι αλγόριθμοι που θα δοκιμαστούν είναι οι ‘brute’, ‘kd\_tree’ και ‘ball\_tree’.
- **leaf\_size**: η παράμετρος αυτή χρησιμοποιείται μόνο στις δένδροειδείς δομές των αλγορίθμων ‘kd\_tree’ και ‘ball\_tree’ και καθορίζει το πόσα δείγματα θα τοποθετηθούν σε κάθε κόμβο (φύλλο) της δένδροειδούς δομής. Στο εσωτερικό ενός φύλλου η αναζήτηση γίνεται με την τεχνική brute-force επειδή είναι πιο αποτελεσματική για μικρό αριθμό δειγμάτων. Γίνεται φανερό ότι αν η τιμή του leaf\_size μεγαλώσει τόσο ώστε να πλησιάζει το μέγεθος του συνόλου εκπαίδευσης, ο αλγόριθμος ουσιαστικά μετατρέπεται σε brute-force. Ωστόσο, η αύξηση του leaf\_size οδηγεί γενικά σε πιο γρήγορη κατασκευή του δέντρου και μείωση της μνήμης που απαιτείται για την αποθήκευσή του.
- **n\_neighbors**: ο αριθμός των κοντινότερων δειγμάτων-γειτόνων που θα χρησιμοποιηθούν για την πρόβλεψη.
- **weights**: η συνάρτηση απόδοσης βάρους που θα χρησιμοποιηθεί για να καθορίσει τη συμβολή κάθε γείτονα στην πρόβλεψη. Οι τιμές που θα εξεταστούν είναι οι ‘uniform’ (όλα τα δείγματα στη γειτονιά έχουν την ίδια επίδραση) και ‘distance’ (η επίδραση κάθε γείτονα είναι αντιστρόφως ανάλογη από την απόστασή του, δηλαδή πιο κοντινά δείγματα έχουν μεγαλύτερη επίδραση στην πρόβλεψη).
- **p**: η παράμετρος του μετρητικού συστήματος Minkowski. Για  $p = 1$  γίνεται χρήση της απόστασης Manhattan, ενώ για  $p = 2$  χρησιμοποιείται ευκλείδεια απόσταση. Για τις υπόλοιπες τιμές χρησιμοποιείται η απόσταση Minkowski.

Για την εύρεση των βέλτιστων τιμών των υπερ-παραμέτρων θα χρησιμοποιηθεί ο αλγόριθμος GridSearchCV με 5-fold CV. Για την απαιτούμενη κανονικοποίηση του συνόλου δεδομένων χρησιμοποιούμε τη μέθοδο *MinMaxScaler* του Scikit-learn (βλ. Ενότητα 3.2.1). Στον Πίνακα 4.9α δίνεται το εύρος των τιμών των υπερ-παραμέτρων που χρησιμοποιήθηκαν για την αναζήτηση του βέλτιστου μοντέλου, και στον Πίνακα 4.9β παρουσιάζονται τα αποτελέσματα του μοντέλου.



**Πίνακας 4.9** – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου *KNeighborsRegressor* που εξετάστηκαν. β) Αξιολόγηση του μοντέλου *KNeighborsRegressor*: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού  $R^2$  και μέσο απόλυτο σφάλμα *MAE*.

α)

	Minimum Value	Maximum Value
<b>Hyper-parameter</b>		
<i>algorithm</i>	'brute', 'kd_tree', 'ball_tree'	
<i>leaf_size</i>	1	60
<i>n_neighbors</i>	4	30
<i>weights</i>	'uniform', 'distance'	
<i>p</i>	1	2

β)

<b>KNeighborsRegressor</b>	
<b>Cross Validation Method:</b>	GridSearchCV(cv = 5, scoring = 'r2')
<b>Fixed Hyper-parameters:</b>	–
<b>Tuned Hyper-parameters:</b>	<i>algorithm</i> = 'ball_tree', <i>leaf_size</i> = 1, <i>n_neighbors</i> = 10, <i>weights</i> = 'distance', <i>p</i> = 1
<b>CV Score (<math>R^2</math>):</b>	0,98822918
<b>Test Set <math>R^2</math> Score:</b>	0,98197127
<b>Test Set M.A.E. :</b>	0,037368

#### 4.5.3 – RadiusNeighbors Regressor

Η παλινδρόμηση που βασίζεται στο ακτινικό μοντέλο γειτόνων (*radius-based neighbors*) χρησιμοποιεί τα  $k$  δείγματα του συνόλου εκπαίδευσης που βρίσκονται μέσα σε μια καθορισμένη ακτίνα  $r$  από το προς πρόβλεψη σημείο, όπου η ακτίνα  $r$  ορίζεται από το χρήστη. Η έξοδος (label) του προς πρόβλεψη σημείου υπολογίζεται και εδώ από τον μέσο όρο των (γνωστών) εξόδων αυτών των δειγμάτων-γειτόνων.

Το μοντέλο του Scikit-learn που υλοποιεί την τεχνική *radius-based neighbors* είναι το *neighbors.RadiusNeighborsRegressor* [32]. Οι υπερ-παραμέτροι του μοντέλου που πρέπει να ρυθμιστούν είναι οι εξής:

- **algorithm**: ο αλγόριθμος που θα χρησιμοποιηθεί για την εύρεση των κοντινότερων γειτόνων. Οι αλγόριθμοι που θα δοκιμαστούν είναι οι 'brute', 'kd\_tree' και 'ball\_tree'.
- **leaf\_size**: η παράμετρος αυτή χρησιμοποιείται μόνο στις δένδροειδείς δομές των αλγορίθμων 'kd\_tree' και 'ball\_tree' και καθορίζει το πόσα δείγματα θα τοποθετηθούν σε κάθε κόμβο (φύλλο) της δένδροειδούς δομής. Για την παράμετρο αυτή ισχύει και εδώ ότι αναφέρθηκε προηγουμένως για το μοντέλο *KNeighborsRegressor*.

#### 4. Δοκιμές Μοντέλων Παλινδρόμησης

- **radius**: η ακτίνα του χώρου μέσα από τον οποίο τα δείγματα που περιέχονται θα χρησιμοποιηθούν για την πρόβλεψη.
- **weights**: η συνάρτηση απόδοσης βάρους που θα χρησιμοποιηθεί για να καθορίσει τη συμβολή κάθε γείτονα στην πρόβλεψη. Οι τιμές που θα εξεταστούν είναι οι ‘uniform’ (όλα τα δείγματα στη γειτονιά έχουν την ίδια επίδραση) και ‘distance’ (η επίδραση κάθε γείτονα είναι αντιστρόφως ανάλογη από την απόστασή του, δηλαδή πιο κοντινά δείγματα έχουν μεγαλύτερη επίδραση στην πρόβλεψη).
- **p**: η παράμετρος του μετρητικού συστήματος Minkowski. Για  $p = 1$  γίνεται χρήση της απόστασης Manhattan, ενώ για  $p = 2$  χρησιμοποιείται ευκλείδεια απόσταση. Για τις υπόλοιπες τιμές χρησιμοποιείται η απόσταση Minkowski.

Για την εύρεση των βέλτιστων τιμών των υπερ-παραμέτρων θα χρησιμοποιηθεί ο αλγόριθμος GridSearchCV με 5-fold CV. Για την απαιτούμενη κανονικοποίηση του συνόλου δεδομένων χρησιμοποιούμε τη μέθοδο *MinMaxScaler* του Scikit-learn (βλ. Ενότητα 3.2.1). Στον Πίνακα 4.10α δίνεται το εύρος των τιμών των υπερ-παραμέτρων που χρησιμοποιήθηκαν για την αναζήτηση του βέλτιστου μοντέλου, και στον Πίνακα 4.10β παρουσιάζονται τα αποτελέσματα του μοντέλου.

**Πίνακας 4.10** – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου *RadiusNeighborsRegressor* που εξετάστηκαν. β) Αξιολόγηση του μοντέλου *RadiusNeighborsRegressor*: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού  $R^2$  και μέσο απόλυτο σφάλμα MAE.

α)

	Minimum Value	Maximum Value
<b>Hyper-parameter</b>		
<i>algorithm</i>	‘brute’, ‘kd_tree’, ‘ball_tree’	
<i>leaf_size</i>	1	60
<i>radius</i>	0.1	20
<i>weights</i>	‘uniform’, ‘distance’	
<i>p</i>	1	2

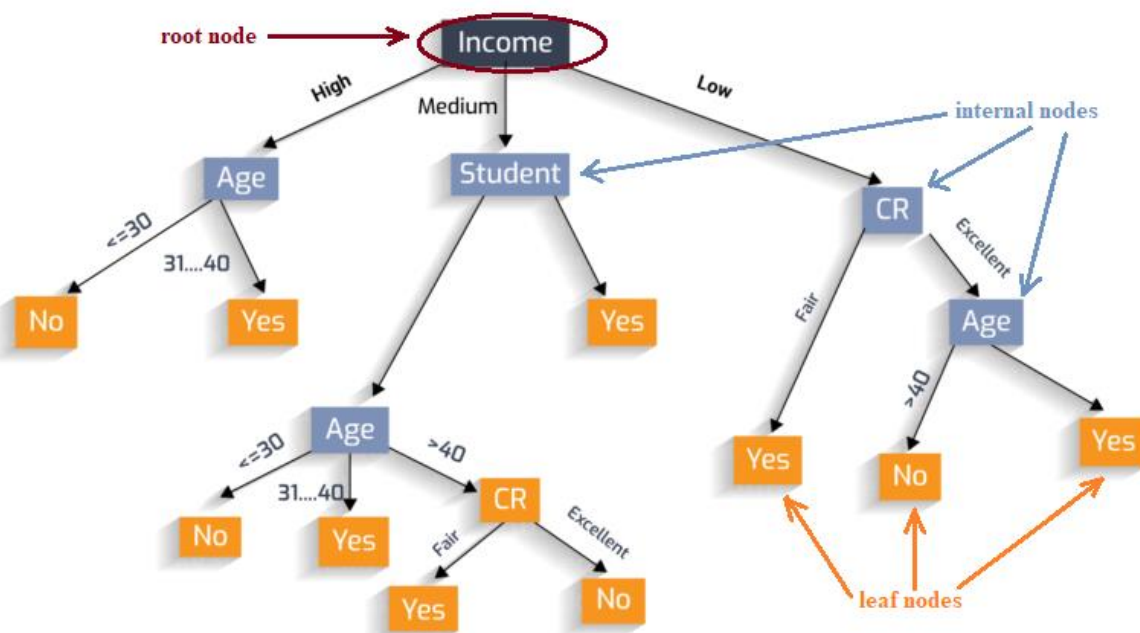
β)

<b>RadiusNeighborsRegressor</b>	
<b>Cross Validation Method:</b>	GridSearchCV(cv = 5, scoring = ‘r2’)
<b>Fixed Hyper-parameters:</b>	–
<b>Tuned Hyper-parameters:</b>	<i>algorithm</i> = ‘ball_tree’, <i>leaf_size</i> = 1, <i>radius</i> = 0.1, <i>weights</i> = ‘distance’, <i>p</i> = 1
<b>CV Score (<math>R^2</math>):</b>	0,98786442
<b>Test Set <math>R^2</math> Score:</b>	0,98136625
<b>Test Set M.A.E. :</b>	0,046482

## 4.6 – Δέντρα Αποφάσεων (Decision Trees)

Τα Δέντρα Αποφάσεων (*Decision Trees*) είναι μια μέθοδος επιτηρούμενης μάθησης που χρησιμοποιείται για ταξινόμηση και παλινδρόμηση. Σκοπός είναι η δημιουργία ενός μοντέλου που μπορεί να προβλέψει την τιμή μιας μεταβλητής μαθαίνοντας απλούς κανόνες επιλογής (πχ. κανόνες if-then-else) από τα χαρακτηριστικά (features) των δειγμάτων με τα οποία εκπαιδεύεται [32]. Στο Σχήμα 4.4 παρουσιάζεται η δομή ενός απλού δέντρου αποφάσεων.

Ένα δέντρο αποφάσεων, όπως υποδηλώνει και το όνομα, έχει δενδροειδή δομή: ξεκινάει από έναν αρχικό κόμβο, που ονομάζεται και *ρίζα (root node)* του δέντρου και περιλαμβάνει το σύνολο των δειγμάτων εκπαίδευσης, και στη συνέχεια διακλαδίζεται σε νέους κόμβους που περιλαμβάνουν υποσύνολα του αρχικού συνόλου εκπαίδευσης. Η κάθε διακλάδωση γίνεται με συγκεκριμένα κριτήρια που συνήθως έχουν να κάνουν με μια απόφαση που παίρνεται αναφορικά με κάποιο χαρακτηριστικό (feature) των δειγμάτων (πχ. αν έχει μια συγκεκριμένη τιμή) [4], [42], [50]. Η διαδικασία αυτή επαναλαμβάνεται για τους νέους κόμβους, οι οποίοι διακλαδίζονται ξανά, έως ότου όλα τα δείγματα σε ένα κόμβο να έχουν την ίδια τιμή εξόδου ή μια περαιτέρω διακλάδωση να μην προσθέτει ουσιαστική αξία στην πρόβλεψη. Οι τερματικοί αυτοί κόμβοι, που δεν διακλαδίζονται περαιτέρω, ονομάζονται και φύλλα (leaf nodes) του δέντρου). Το βάθος (*depth*) του δέντρου είναι το μήκος της μακρύτερης διαδρομής από τη ρίζα σε ένα φύλλο του δέντρου. Για παράδειγμα, το δέντρο του Σχήματος 4.4 έχει βάθος 4.



Σχήμα 4.4 – Η δομή ενός δέντρου αποφάσεων (decision tree).

Τα πλεονεκτήματα των δέντρων αποφάσεων είναι ότι μπορούν να οπτικοποιηθούν και είναι εύκολα ερμηνεύσιμα. Μπορούν επίσης να χειριστούν το ίδιο αποτελεσματικά συνεχείς και διακριτές μεταβλητές, και προσφέρουν μια σαφή ένδειξη των χαρακτηριστικών που είναι πιο σημαντικά για την πρόβλεψη.

Το μοντέλο του Scikit-learn που υλοποιεί το Δέντρο Αποφάσεων για προβλήματα παλινδρόμησης είναι το *tree.DecisionTreeRegressor* [32]. Οι υπερ-παράμετροι του μοντέλου που πρέπει να ρυθμιστούν είναι οι εξής:

- **criterion**: το κριτήριο με το οποίο θα εκτιμηθεί η ποιότητα μιας διακλάδωσης. Οι υποστηριζόμενες τιμές του είναι 'mse' (mean squared error – ελαχιστοποιεί τις L2 απώλειες) και 'mae' (mean absolute error – ελαχιστοποιεί τις L1 απώλειες). Εμείς θα χρησιμοποιήσουμε μόνο το κριτήριο 'mae'.
- **max\_depth**: Το μέγιστο επιτρεπτό βάθος του δέντρου. Όσο μεγαλύτερο είναι το βάθος του δέντρου, τόσο περισσότερες διακλαδώσεις έχει και τόσο περισσότερες πληροφορίες αποτυπώνει για τα δεδομένα. Πολύ μεγάλα βάθη οδηγούν το μοντέλο στο να μάθει πληροφορίες πολύ συγκεκριμένες για το σύνολο εκπαίδευσης και προκαλούν Overfitting ( το μοντέλο δεν μπορεί να γενικεύσει σε νέα δεδομένα).
- **max\_features**: ο αριθμός των χαρακτηριστικών (features) των δειγμάτων που θα πρέπει να εξεταστούν σε κάθε διακλάδωση. Αν έχει την τιμή *None* τότε για κάθε διακλάδωση εξετάζονται όλα τα χαρακτηριστικά (max\_features=n\_features).
- **min\_samples\_split**: ο ελάχιστος αριθμός δειγμάτων που πρέπει να περιέχει ένας εσωτερικός κόμβος του δέντρου για να μπορέσει να διακλαδωθεί (αν είναι δεκαδικός αριθμός τότε είναι το ποσοστό των συνολικών δειγμάτων). Χρησιμοποιείται για να ρυθμίσει το Overfitting: υψηλές τιμές του αποτρέπουν το μοντέλο από το να μάθει σχέσεις πολύ συγκεκριμένες για το σύνολο δειγμάτων με το οποίο εκπαιδεύεται. Πολύ υψηλές τιμές του παρόλα αυτά μπορεί να αποτρέψουν το μοντέλο από το να προσαρμοστεί επαρκώς στα δεδομένα εκπαίδευσης και οδηγούν σε Underfitting.
- **min\_samples\_leaf**: ο ελάχιστος αριθμός δειγμάτων που πρέπει να περιέχονται σε έναν τερματικό κόμβο (φύλλο) του δέντρου (αν είναι δεκαδικός αριθμός τότε είναι το ποσοστό των συνολικών δειγμάτων). Μια διακλάδωση σε οποιοδήποτε βάθος του δέντρου θα εξεταστεί μόνο αν αφήνει τουλάχιστον min\_samples\_leaf δείγματα σε κάθε έναν από τους κόμβους-παιδιά του. Η παράμετρος αυτή χρησιμοποιείται με τον ίδιο τρόπο με την παράμετρο min\_samples\_split για τη ρύθμιση του Overfitting.

Οι υπερ-παράμετροι max\_depth, max\_features, min\_samples\_split και min\_samples\_leaf είναι αυτοί που ρυθμίζουν το μέγεθος και την πολυπλοκότητα του δέντρου, και συνεπώς την κατανάλωση μνήμης, επομένως πρέπει να ρυθμιστούν προσεχτικά.

Για την εύρεση των βέλτιστων τιμών των υπερ-παραμέτρων θα χρησιμοποιηθεί ο αλγόριθμος GridSearchCV με 5-fold CV. Για την απαιτούμενη κανονικοποίηση του συνόλου δεδομένων χρησιμοποιούμε τη μέθοδο *MinMaxScaler* του Scikit-learn (βλ. Ενότητα 3.2.1). Στον Πίνακα 4.11α δίνεται το εύρος των τιμών των υπερ-παραμέτρων που χρησιμοποιήθηκαν για την αναζήτηση του βέλτιστου μοντέλου, και στον Πίνακα 4.11β παρουσιάζονται τα αποτελέσματα του μοντέλου.

#### 4. Δοκιμές Μοντέλων Παλινδρόμησης

**Πίνακας 4.11** – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου *DecisionTreeRegressor* που εξετάστηκαν. β) Αξιολόγηση του μοντέλου *DecisionTreeRegressor*: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού  $R^2$  και μέσο απόλυτο σφάλμα *MAE*.

α)

	Minimum Value	Maximum Value
<b>Hyper-parameter</b>		
<i>criterion</i>	'mae'	
<i>max_depth</i>	1	100
<i>max_features</i>	1	<i>None</i> = $n_{features} = 2$
<i>min_samples_split</i>	0,001	1
<i>min_samples_leaf</i>	0,001	0,5

β)

<b>DecisionTreeRegressor</b>	
<b>Cross Validation Method:</b>	GridSearchCV(cv = 5, scoring = 'r2')
<b>Fixed Hyper-parameters:</b>	<i>criterion</i> = 'mae', <i>random_state</i> = 0
<b>Tuned Hyper-parameters:</b>	<i>max_depth</i> = 10, <i>max_features</i> = <i>None</i> = 2, <i>min_samples_split</i> = 0.0047, <i>min_samples_leaf</i> = 0.0017
<b>CV Score (<math>R^2</math>):</b>	0,98871212
<b>Test Set <math>R^2</math> Score:</b>	0,98146038
<b>Test Set M.A.E. :</b>	0,037619

## 4.7 – Αλγόριθμοι Μάθησης Συνόλων (Ensemble Methods)

Ο στόχος των *Μεθόδων Μάθησης Συνόλων (Ensemble Methods)* είναι να συνδυάσουν πολλά απλά μοντέλα βάσης ώστε να κατασκευάσουν ένα νέο πιο ισχυρό μοντέλο που εκτελεί πιο αξιόπιστες προβλέψεις από ότι το κάθε μοντέλο βάσης χωριστά [32], [51]. Υπάρχουν δύο βασικές κατηγορίες των μεθόδων μάθησης συνόλων:

- Οι *μέθοδοι στάθμισης (averaging methods)*, στις οποίες ο στόχος είναι να κατασκευαστούν πολλά ανεξάρτητα μοντέλα βάσης και στη συνέχεια να συνδυαστούν ώστε να πάρουμε τον μέσο όρο των προβλέψεων όλων αυτών των μοντέλων. Το συνδυαστικό μοντέλο έχει σημαντικά μικρότερη μεταβλητότητα (variance). Στην κατηγορία αυτή ανήκουν οι αλγόριθμοι Random Forests και Extremely Randomized Trees.
- Οι *μέθοδοι ενίσχυσης (boosting methods)* κατασκευάζουν πολλά μοντέλα διαδοχικά, το ένα μετά το άλλο, προσπαθώντας να μειώσουν το συστηματικό σφάλμα (bias) του συνολικού συνδυαστικού μοντέλου. Στην κατηγορία αυτή ανήκουν οι αλγόριθμοι AdaBoost και Gradient Tree Boosting.

### 4.7.1 – Random Forest Regressor

Ο αλγόριθμος παλινδρόμησης *Random Forest* ανήκει στην κατηγορία των μεθόδων στάθμισης (averaging methods) και στηρίζεται στο μοντέλο του Δέντρου Αποφάσεων (Decision Tree): η λειτουργία του συνίσταται στην κατασκευή πολλών δέντρων αποφάσεων με τυχαίο τρόπο κατά την εκπαίδευση. Η τελική πρόβλεψη του μοντέλου είναι ο μέσος όρος των προβλέψεων όλων των επιμέρους δέντρων αποφάσεων που κατασκευάστηκαν [52], [53]. Το κάθε επιμέρους δέντρο αποφάσεων κατασκευάζεται με επιλογή ενός αριθμού *τυχαίων* δειγμάτων από το σύνολο εκπαίδευσης, όπου το κάθε δείγμα μπορεί να χρησιμοποιηθεί και περισσότερες από μία φορές (τεχνική bootstrap aggregating). Επιπλέον, όταν διακλαδίζεται κάθε κόμβος κατά την κατασκευή ενός δέντρου, είναι δυνατό να επιλέξουμε να εξεταστεί ένα *τυχαίο* υποσύνολο των χαρακτηριστικών αντί για όλα τα χαρακτηριστικά. Η εισαγωγή αυτών των δύο παραγόντων τυχαιότητας οδηγεί στην κατασκευή πολλών ποικιλόμορφων δέντρων αποφάσεων, τα οποία όταν συνδυαστούν στο μοντέλο Random Forest επιφέρουν σημαντική μείωση της μεταβλητότητας (variance) [32].

Το μοντέλο του Scikit-learn που υλοποιεί τον αλγόριθμο Random Forest είναι το *ensemble.RandomForestRegressor* [32]. Οι υπερ-παραμέτροι του μοντέλου που πρέπει να ρυθμιστούν είναι οι εξής:

- ***n\_estimators***: ο αριθμός των επιμέρους δέντρων αποφάσεων που θα κατασκευαστούν.
- ***criterion***: το κριτήριο με το οποίο θα εκτιμηθεί η ποιότητα μιας διακλάδωσης. Οι υποστηριζόμενες τιμές του είναι ‘mse’ (mean squared error) και ‘mae’ (mean absolute error). Εμείς θα χρησιμοποιήσουμε μόνο το κριτήριο ‘mae’.
- ***max\_depth***: Το μέγιστο επιτρεπτό βάθος του κάθε δέντρου. Όσο μεγαλύτερο είναι το βάθος του δέντρου, τόσο περισσότερες διακλαδώσεις έχει και τόσο περισσότερες πληροφορίες αποτυπώνει για τα δεδομένα. Πολύ μεγάλα βάθη οδηγούν το μοντέλο στο να

μάθει πληροφορίες πολύ συγκεκριμένες για το σύνολο εκπαίδευσης και προκαλούν Overfitting ( το μοντέλο δεν μπορεί να γενικεύσει σε νέα δεδομένα).

- **max\_features**: ο αριθμός των χαρακτηριστικών (features) των δειγμάτων που θα πρέπει να εξεταστούν σε κάθε διακλάδωση. Τα χαρακτηριστικά που θα εξεταστούν επιλέγονται τυχαία από το σύνολο των χαρακτηριστικών [32]. Σε αυτή την παράμετρο θα δώσουμε την τιμή *None* ώστε για κάθε διακλάδωση να εξετάζονται όλα τα χαρακτηριστικά.
- **min\_samples\_split**: ο ελάχιστος αριθμός δειγμάτων που πρέπει να περιέχει ένας εσωτερικός κόμβος του δέντρου για να μπορέσει να διακλαδωθεί (αν είναι δεκαδικός αριθμός τότε είναι το ποσοστό των συνολικών δειγμάτων). Χρησιμοποιείται για να ρυθμίσει το Overfitting: υψηλές τιμές του αποτρέπουν το μοντέλο από το να μάθει σχέσεις πολύ συγκεκριμένες για το σύνολο δειγμάτων με το οποίο εκπαιδεύεται. Πολύ υψηλές τιμές του παρόλα αυτά μπορεί να αποτρέψουν το μοντέλο από το να προσαρμοστεί επαρκώς στα δεδομένα εκπαίδευσης και οδηγούν σε Underfitting.
- **min\_samples\_leaf**: ο ελάχιστος αριθμός δειγμάτων που πρέπει να περιέχονται σε έναν τερματικό κόμβο (φύλλο) του δέντρου (αν είναι δεκαδικός αριθμός τότε είναι το ποσοστό των συνολικών δειγμάτων). Μια διακλάδωση σε οποιοδήποτε βάθος του δέντρου θα εξεταστεί μόνο αν αφήνει τουλάχιστον min\_samples\_leaf δείγματα σε κάθε έναν από τους κόμβους-παιδιά του. Η παράμετρος αυτή χρησιμοποιείται με τον ίδιο τρόπο με την παράμετρο min\_samples\_split για τη ρύθμιση του Overfitting.

Οι υπερ-παράμετροι criterion, max\_depth, max\_features, min\_samples\_split και min\_samples\_leaf έχουν επίδραση στο κάθε επιμέρους δέντρο του δάσους. Η υπερ-παράμετρος n\_estimators αναφέρεται στο συνολικό συνδυαστικό μοντέλο (Random Forest). Για την εύρεση των βέλτιστων τιμών των υπερ-παραμέτρων θα χρησιμοποιηθεί ο αλγόριθμος GridSearchCV με 5-fold CV. Για την απαιτούμενη κανονικοποίηση του συνόλου δεδομένων χρησιμοποιούμε τη μέθοδο *MinMaxScaler* του Scikit-learn (βλ. Ενότητα 3.2.1). Στον Πίνακα 4.12α δίνεται το εύρος των τιμών των υπερ-παραμέτρων που χρησιμοποιήθηκαν για την αναζήτηση του βέλτιστου μοντέλου, και στον Πίνακα 4.12β παρουσιάζονται τα αποτελέσματα του μοντέλου.

**Πίνακας 4.12** – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου *RandomForestRegressor* που εξετάστηκαν. β) Αξιολόγηση του μοντέλου *RandomForestRegressor*: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού  $R^2$  και μέσο απόλυτο σφάλμα MAE.

α)

	Minimum Value	Maximum Value
<b>Hyper-parameter</b>		
<i>n_estimators</i>	5	200
<i>criterion</i>	'mae'	
<i>max_depth</i>	3	15
<i>max_features</i>	<i>None</i> = $n_{features} = 2$	
<i>min_samples_split</i>	0,001	1
<i>min_samples_leaf</i>	0,0001	0,5

β)

RandomForestRegressor	
<b>Cross Validation Method:</b>	GridSearchCV(cv = 5, scoring = 'r2')
<b>Fixed Hyper-parameters:</b>	<i>criterion = 'mae' , max_features = None = 2 , random_state = 0</i>
<b>Tuned Hyper-parameters:</b>	<i>n_estimators = 100 , max_depth = 10 , min_samples_split = 0.002 , min_samples_leaf = 0.0005</i>
<b>CV Score (<math>R^2</math>):</b>	0,98903021
<b>Test Set <math>R^2</math> Score:</b>	0,98136952
<b>Test Set M.A.E. :</b>	0,037354

Συγκρίνοντας τα αποτελέσματα του παραπάνω πίνακα για το μοντέλο Random Forest με αυτά του απλού δέντρου αποφάσεων (Πίνακας 4.11), παρατηρούμε ότι ο συνδυασμός πολλών δέντρων αποφάσεων σε ένα μοντέλο Random Forest οδηγεί σε ελαφρώς καλύτερη (υψηλότερη) τιμή του συντελεστή προσδιορισμού  $R^2$ , όπως αυτός προκύπτει από το Cross-Validation, και σε χαμηλότερο μέσο απόλυτο σφάλμα πρόβλεψης.

#### 4.7.2 – Extra Trees Regressor

Ο αλγόριθμος *Extra Trees (Extremely Randomized Trees)* ανήκει και αυτός στην κατηγορία των μεθόδων στάθμισης (averaging methods) και λειτουργεί με τον ίδιο τρόπο με τον αλγόριθμο Random Forest, με τη διαφορά ότι εισάγει ένα ακόμη επίπεδο τυχαιότητας στον τρόπο με τον οποίο πραγματοποιούνται οι διακλαδώσεις σε κάθε δέντρο του δάσους: ενώ ο αλγόριθμος Random Forest προσπαθεί να βρει για κάθε ένα από τα (τυχαία επιλεγμένα) χαρακτηριστικά που εξετάζονται κατά τη διακλάδωση το όριο (threshold) που θα οδηγήσει στη βέλτιστη διακλάδωση, ο αλγόριθμος Extra Trees λαμβάνει τυχαία κάποια όρια για κάθε χαρακτηριστικό και στη συνέχεια επιλέγει τα βέλτιστα όρια από αυτά για να κατασκευάσει τον κανόνα διακλάδωσης [32], [54]. Με αυτό τον τρόπο συνήθως επιτυγχάνεται μια περαιτέρω μείωση της μεταβλητότητας (variance) του μοντέλου, με αντάλλαγμα όμως μια μικρή αύξηση του συστηματικού σφάλματος (bias).

Το μοντέλο του Scikit-learn που υλοποιεί τον αλγόριθμο Extra Trees είναι το *ensemble.ExtraTreesRegressor* [32]. Οι υπερ-παράμετροι του μοντέλου που πρέπει να ρυθμιστούν είναι ίδιοι με αυτές του μοντέλου RandomForestRegressor και επιτελούν τις ίδιες ακριβώς λειτουργίες (βλ. Ενότητα 4.7.1).

Όπως στο μοντέλο RandomForestRegressor, οι υπερ-παράμετροι criterion, max\_depth, max\_features, min\_samples\_split και min\_samples\_leaf έχουν επίδραση στο κάθε επιμέρους δέντρο του δάσους, ενώ η υπερ-παράμετρος n\_estimators αναφέρεται στο συνολικό συνδυαστικό μοντέλο. Για την εύρεση των βέλτιστων τιμών των υπερ-παραμέτρων θα χρησιμοποιηθεί ο αλγόριθμος GridSearchCV με 5-fold CV. Για την απαιτούμενη κανονικοποίηση του συνόλου δεδομένων χρησιμοποιούμε τη μέθοδο MinMaxScaler του Scikit-learn (βλ. Ενότητα 3.2.1). Στον Πίνακα 4.13α δίνεται το εύρος των τιμών των υπερ-παραμέτρων



#### 4. Δοκιμές Μοντέλων Παλινδρόμησης

που χρησιμοποιήθηκαν για την αναζήτηση του βέλτιστου μοντέλου, και στον Πίνακα 4.13β παρουσιάζονται τα αποτελέσματα του μοντέλου.

**Πίνακας 4.13** – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου *ExtraTreesRegressor* που εξετάστηκαν. β) Αξιολόγηση του μοντέλου *ExtraTreesRegressor*: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού  $R^2$  και μέσο απόλυτο σφάλμα MAE.

α)

	Minimum Value	Maximum Value
<b>Hyper-parameter</b>		
<i>n_estimators</i>	5	200
<i>criterion</i>	'mae'	
<i>max_depth</i>	9	None = $\infty$
<i>max_features</i>	None = $n_{features} = 2$	
<i>min_samples_split</i>	0,001	1
<i>min_samples_leaf</i>	0,0001	0,5

β)

ExtraTreesRegressor	
<b>Cross Validation Method:</b>	GridSearchCV(cv = 5, scoring = 'r2')
<b>Fixed Hyper-parameters:</b>	<i>criterion</i> = 'mae', <i>max_features</i> = None = 2, <i>random_state</i> = 0
<b>Tuned Hyper-parameters:</b>	<i>n_estimators</i> = 120, <i>max_depth</i> = None, <i>min_samples_split</i> = 0.003, <i>min_samples_leaf</i> = 0.001
<b>CV Score (<math>R^2</math>):</b>	0,98894504
<b>Test Set <math>R^2</math> Score:</b>	0,98251049
<b>Test Set M.A.E. :</b>	0,038221

Συγκρίνοντας το μοντέλο Extra Trees με το μοντέλο του απλού δέντρου αποφάσεων (Πίνακας 4.11) και το μοντέλο Random Forest (Πίνακας 4.12), προκύπτει ότι το μοντέλο Extra Trees παρουσιάζει τον καλύτερο συντελεστή προσδιορισμού κατά την εφαρμογή του στο σύνολο δοκιμής, ακόμη και έναντι του Random Forest, γεγονός που οφείλεται στη μείωση της μεταβλητότητας (variance) λόγω της εισαγωγής του επιπλέον όρου τυχαιότητας, όπως προαναφέρθηκε. Η ταυτόχρονη όμως αύξηση του συστηματικού σφάλματος (bias) αποτυπώνεται στο μέσο απόλυτο σφάλμα, το οποίο έχει την υψηλότερη τιμή για το Extra Trees σε σύγκριση με τα άλλα δύο μοντέλα.

### 4.7.3 – AdaBoost Regressor

Ο αλγόριθμος *AdaBoost* προτάθηκε για πρώτη φορά το 1995 από τους Freund και Schapire [55] και ανήκει στην κατηγορία των μεθόδων ενίσχυσης (boosting methods). Μπορεί να χρησιμοποιηθεί σε προβλήματα ταξινόμησης αλλά και παλινδρόμησης. Η βασική ιδέα του αλγορίθμου *AdaBoost* είναι η κατασκευή μιας ακολουθίας αδύναμων μοντέλων (συνήθως μικρών δέντρων αποφάσεων) με τροποποιημένες κάθε φορά εκδοχές των δεδομένων εκπαίδευσης [32]. Στη συνέχεια οι προβλέψεις από όλα τα μοντέλα συνδυάζονται με κάποια βαρύτητα για να δώσουν την τελική πρόβλεψη.

Ο αλγόριθμος ξεκινά εκπαιδύοντας ένα μοντέλο δέντρου αποφάσεων με το αρχικό σύνολο εκπαίδευσης. Στη συνέχεια εφαρμόζει συντελεστές βαρύτητας  $w_1, w_2, \dots, w_N$  σε κάθε ένα από τα  $N$  δείγματα εκπαίδευσης, και εκπαιδεύει εκ νέου το προηγούμενο μοντέλο με το νέο τροποποιημένο σύνολο εκπαίδευσης. Ο συντελεστής βαρύτητας του κάθε δείγματος επιλέγεται με τέτοιο τρόπο ώστε τα δείγματα που στο προηγούμενο βήμα δεν προβλέφθηκαν σωστά να έχουν αυξημένη βαρύτητα, ενώ για τα δείγματα που η πρόβλεψη έγινε σωστά ο συντελεστής βαρύτητάς τους μειώνεται. Η διαδικασία αυτή επαναλαμβάνεται και καθώς οι επαναλήψεις αυξάνονται, τα δείγματα εκείνα που είναι πιο δύσκολο να προβλεφθούν αποκτούν όλο και μεγαλύτερη επίδραση. Επομένως, κάθε διαδοχικό μοντέλο που εκπαιδεύεται αναγκάζεται να επικεντρωθεί σε εκείνα τα δείγματα που δεν προβλέφθηκαν σωστά από τα προηγούμενα μοντέλα. [42], [56]

Το μοντέλο του Scikit-learn που υλοποιεί τον αλγόριθμο *AdaBoost* για προβλήματα παλινδρόμησης είναι το *ensemble.AdaBoostRegressor* [32]. Οι υπερ-παράμετροι του μοντέλου που πρέπει να ρυθμιστούν είναι οι εξής:

- **base\_estimator**: το απλό μοντέλο παλινδρόμησης (συνήθως ένα δέντρο αποφάσεων) που θα χρησιμοποιηθεί ως η βάση πάνω στην οποία θα κατασκευαστεί το ενισχυμένο σύνολο.
- **n\_estimators**: ο μέγιστος αριθμός μοντέλων που μπορούν να κατασκευαστούν προτού τερματιστεί η διαδικασία ενίσχυσης (δηλαδή ο μέγιστος αριθμός σταδίων ενίσχυσης). Στην περίπτωση επίτευξης απόλυτα ακριβούς πρόβλεψης, η επαναληπτική διαδικασία θα τερματιστεί νωρίτερα.
- **learning\_rate**: η παράμετρος αυτή καθορίζει το ποσοστό κατά το οποίο θα μειωθεί η συμβολή του κάθε επιμέρους μοντέλου στην τελική πρόβλεψη. Όσο μειώνουμε την τιμή αυτής της παραμέτρου θα πρέπει να αυξάνουμε την τιμή του `n_estimators`, κάτι που συνήθως δίνει πιο ακριβές συνολικό μοντέλο αλλά με αυξημένη πολυπλοκότητα (λόγω του μεγαλύτερου αριθμού επιμέρους μοντέλων που πρέπει να εκπαιδευτούν).
- **loss**: η συνάρτηση απωλειών που θα χρησιμοποιηθεί κατά την τροποποίηση των συντελεστών βαρύτητας κάθε δείγματος μετά από κάθε επανάληψη. Οι δυνατές τιμές της είναι: 'linear', 'square' και 'exponential'.

#### 4. Δοκιμές Μοντέλων Παλινδρόμησης

Σαν μοντέλο βάσης θα χρησιμοποιήσουμε το βέλτιστο δέντρο αποφάσεων που υπολογίστηκε στην Ενότητα 4.6, δηλαδή το δέντρο αποφάσεων με υπερ-παραμέτρους: `criterion='mae'`, `max_features=None`, `max_depth=10`, `min_samples_split=0.0047`, `min_samples_leaf=0.0017`.

Για την εύρεση των βέλτιστων τιμών των υπερ-παραμέτρων θα χρησιμοποιηθεί ο αλγόριθμος `GridSearchCV` με *5-fold CV*. Για την απαιτούμενη κανονικοποίηση του συνόλου δεδομένων χρησιμοποιούμε τη μέθοδο *MinMaxScaler* του `Scikit-learn` (βλ. Ενότητα 3.2.1). Στον Πίνακα 4.14α δίνεται το εύρος των τιμών των υπερ-παραμέτρων που χρησιμοποιήθηκαν για την αναζήτηση του βέλτιστου μοντέλου, και στον Πίνακα 4.14β παρουσιάζονται τα αποτελέσματα του μοντέλου.

**Πίνακας 4.14** – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου `AdaBoostRegressor` που εξετάστηκαν. β) Αξιολόγηση του μοντέλου `AdaBoostRegressor`: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού  $R^2$  και μέσο απόλυτο σφάλμα *MAE*.

α)

	Minimum Value	Maximum Value
<b>Hyper-parameter</b>		
<i>n_estimators</i>	50	1000
<i>learning_rate</i>	0,0001	1
<i>loss</i>	'linear', 'square', 'exponential'	

β)

<b>AdaBoostRegressor</b>	
<b>Cross Validation Method:</b>	<code>GridSearchCV(cv = 5, scoring = 'r2')</code>
<b>Fixed Hyper-parameters:</b>	<code>base_estimator = DecisionTreeRegressor (criterion='mae', max_features=None, max_depth=10, min_samples_split=0.0047, min_samples_leaf=0.0017, random_state=0), random_state = 0</code>
<b>Tuned Hyper-parameters:</b>	<code>loss = 'linear', learning_rate = 0.02, n_estimators = 84</code>
<b>CV Score (<math>R^2</math>):</b>	0,98894535
<b>Test Set <math>R^2</math> Score:</b>	0,98115137
<b>Test Set M.A.E. :</b>	0,037528

Συγκρίνοντας το μοντέλο `AdaBoost` με το μοντέλο του απλού δέντρου αποφάσεων (Πίνακας 4.11) παρατηρούμε ότι η μέθοδος της ενίσχυσης επιφέρει μείωση του συστηματικού σφάλματος (bias), η οποία αποτυπώνεται στο χαμηλότερο μέσο απόλυτο σφάλμα πρόβλεψης του μοντέλου `AdaBoost`.

#### 4.7.4 – Gradient Boosting Regressor

Η μέθοδος *Gradient Tree Boosting* είναι μια γενίκευση της μεθόδου ενίσχυσης (boosting) για παραγωγίσιμες συναρτήσεις απωλειών. Η μέθοδος αυτή δουλεύει με παρόμοιο τρόπο με τον αλγόριθμο AdaBoost: χρησιμοποιεί απλά δέντρα αποφάσεων σαν μοντέλα βάσης τα οποία εκπαιδεύει διαδοχικά σε τροποποιημένες κάθε φορά εκδοχές του συνόλου εκπαίδευσης, και προσπαθεί σε κάθε επανάληψη να ελαχιστοποιήσει τη συνάρτηση απωλειών αριθμητικά, ψάχνοντας να βρει την κατεύθυνση όπου η κλίση της συνάρτησης απωλειών (η οποία δίνεται από την παράγωγό της) γίνεται πιο απότομη. Κινούμενη προς αυτή την κατεύθυνση μπορεί να φτάσει πιο γρήγορα στο σημείο του ελαχίστου [57]. Επομένως, η μέθοδος Gradient Tree Boosting αποτελεί ουσιαστικά μια μέθοδο Gradient Descent. Μια μικρή παραλλαγή της μεθόδου αυτής προτάθηκε από τον Friedman το 1999 [58]: σε κάθε επανάληψη του αλγορίθμου χρησιμοποιείται για την εκπαίδευση του εκάστοτε δέντρου ένα μόνο υποσύνολο (subsample) του συνόλου εκπαίδευσης, το οποίο επιλέγεται τυχαία ως ένα ποσοστό των συνολικών δειγμάτων. Η εισαγωγή αυτής της παραμέτρου τυχαιότητας βελτιώνει τις επιδόσεις του μοντέλου και βοηθά στην αποφυγή του φαινομένου Overfitting.

Το μοντέλο του Scikit-learn που υλοποιεί τον αλγόριθμο Gradient Tree Boosting για προβλήματα παλινδρόμησης είναι το *ensemble.GradientBoostingRegressor* [32]. Οι υπερ-παραμέτροι του μοντέλου που πρέπει να ρυθμιστούν είναι οι εξής:

- **loss**: η συνάρτηση απωλειών που πρέπει να ελαχιστοποιηθεί σε κάθε επανάληψη. Οι δυνατές τιμές της είναι: 'ls' (least squares), 'lad' (least absolute deviation), 'huber' (ένας συνδυασμός των δύο προηγούμενων) και 'quantile'. Εμείς θα χρησιμοποιήσουμε μόνο τη συνάρτηση ελαχίστων τετραγώνων ('ls') η οποία είναι η φυσική επιλογή για προβλήματα παλινδρόμησης.
- **n\_estimators**: ο μέγιστος αριθμός μοντέλων μπορούν να κατασκευαστούν προτού τερματιστεί η διαδικασία ενίσχυσης (δηλαδή ο μέγιστος αριθμός σταδίων ενίσχυσης). Η μέθοδος Gradient Boosting έχει γενικά καλή συμπεριφορά ως προς το φαινόμενο Overfitting, επομένως μεγάλες τιμές αυτής της παραμέτρου συνήθως δίνουν καλύτερο μοντέλο.
- **learning\_rate**: η παράμετρος αυτή καθορίζει το ποσοστό κατά το οποίο θα μειωθεί η συμβολή του κάθε επιμέρους μοντέλου στην τελική πρόβλεψη. Όσο μειώνουμε την τιμή αυτής της παραμέτρου θα πρέπει να αυξάνουμε την τιμή του n\_estimators, κάτι που συνήθως δίνει πιο ακριβές συνολικό μοντέλο αλλά με αυξημένη πολυπλοκότητα (λόγω του μεγαλύτερου αριθμού επιμέρους μοντέλων που πρέπει να εκπαιδευτούν).
- **subsample**: το ποσοστό των δειγμάτων εκπαίδευσης που επιλέγονται σε κάθε επανάληψη για την εκπαίδευση των επιμέρους δέντρων (η επιλογή των δειγμάτων γίνεται με τυχαίο τρόπο). Αν είναι 1 τότε χρησιμοποιούνται όλα τα δείγματα. Τιμές μικρότερες του 1 οδηγούν σε μείωση της μεταβλητότητας (variance) και αύξηση του συστηματικού σφάλματος (bias).
- **max\_depth**: Το μέγιστο επιτρεπτό βάθος των επιμέρους δέντρων που κατασκευάζονται. Όσο μεγαλύτερο είναι το βάθος του δέντρου, τόσο περισσότερες διακλαδώσεις έχει και τόσο περισσότερες πληροφορίες αποτυπώνει για τα δεδομένα.

- **max\_features**: ο αριθμός των χαρακτηριστικών (features) των δειγμάτων που θα πρέπει να εξεταστούν σε κάθε διακλάδωση. Αν έχει την τιμή *None* τότε για κάθε διακλάδωση εξετάζονται όλα τα χαρακτηριστικά ( $max\_features=n\_features$ ).
- **min\_samples\_split**: ο ελάχιστος αριθμός δειγμάτων που πρέπει να περιέχει ένας εσωτερικός κόμβος του δέντρου για να μπορέσει να διακλαδωθεί (αν είναι δεκαδικός αριθμός τότε είναι το ποσοστό των συνολικών δειγμάτων).
- **min\_samples\_leaf**: ο ελάχιστος αριθμός δειγμάτων που πρέπει να περιέχονται σε έναν τερματικό κόμβο (φύλλο) του δέντρου (αν είναι δεκαδικός αριθμός τότε είναι το ποσοστό των συνολικών δειγμάτων). Μια διακλάδωση σε οποιοδήποτε βάθος του δέντρου θα εξεταστεί μόνο αν αφήνει τουλάχιστον  $min\_samples\_leaf$  δείγματα σε κάθε έναν από τους κόμβους-παιδιά του.

Οι παράμετροι *max\_depth*, *max\_features*, *min\_samples\_split* και *min\_samples\_leaf* είναι αυτοί που ρυθμίζουν το μέγεθος και την πολυπλοκότητα του κάθε επιμέρους δέντρου του μοντέλου. Οι παράμετροι *n\_estimators*, *learning\_rate* και *subsample* είναι αυτές που αφορούν τη διαδικασία ενίσχυσης (boosting) του μοντέλου.

Για την εύρεση των βέλτιστων τιμών των υπερ-παραμέτρων θα χρησιμοποιηθεί ο αλγόριθμος GridSearchCV με *5-fold CV*. Λόγω του μεγάλου πλήθους των υπερ-παραμέτρων που χρειάζονται ρύθμιση σε αυτό το μοντέλο, η διαδικασία που θα ακολουθήσουμε είναι η εξής:

1. Αρχικά ρυθμίζουμε το *learning\_rate* σε μια σχετικά μεγάλη τιμή και υπολογίζουμε τη βέλτιστη τιμή του *n\_estimators*, η οποία για υψηλό *learning\_rate* είναι σχετικά μικρή. Αυτό το κάνουμε για να έχουμε μικρότερο αριθμό επιμέρους δέντρων στο μοντέλο, άρα μικρότερο χρόνο εκπαίδευσης ώστε να μπορέσουμε να βρούμε πιο γρήγορα τις βέλτιστες τιμές για όλες τις υπόλοιπες υπερ-παραμέτρους.
2. Για αυτές τις τιμές του *learning\_rate* και του *n\_estimators* υπολογίζουμε τις βέλτιστες τιμές των υπερ-παραμέτρων που σχετίζονται με τη δομή των επιμέρους δέντρων του μοντέλου (*max\_depth*, *max\_features*, *min\_samples\_split*, *min\_samples\_leaf*).
3. Στη συνέχεια προχωράμε στη βέλτιστη ρύθμιση των υπερ-παραμέτρων που σχετίζονται με την ενίσχυση: μειώνουμε το *learning\_rate* και αυξάνουμε το *n\_estimators* μέχρι να βρούμε το βέλτιστο συνδυασμό τους. Υπολογίζουμε επίσης το βέλτιστο *subsample*.

Για την απαιτούμενη κανονικοποίηση του συνόλου δεδομένων χρησιμοποιούμε τη μέθοδο *MinMaxScaler* του Scikit-learn (βλ. Ενότητα 3.2.1). Στον Πίνακα 4.15α δίνεται το εύρος των τιμών των υπερ-παραμέτρων που χρησιμοποιήθηκαν για την αναζήτηση του βέλτιστου μοντέλου, και στον Πίνακα 4.15β παρουσιάζονται τα αποτελέσματα του μοντέλου.

#### 4. Δοκιμές Μοντέλων Παλινδρόμησης

**Πίνακας 4.15** – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου *GradientBoostingRegressor* που εξετάστηκαν. β) Αξιολόγηση του μοντέλου *GradientBoostingRegressor*: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού  $R^2$  και μέσο απόλυτο σφάλμα MAE.

α)

	Minimum Value	Maximum Value
<b>Hyper-parameter</b>		
<i>n_estimators</i>	5	9000
<i>learning_rate</i>	0,001	0,5
<i>subsample</i>	0,6	1
<i>max_depth</i>	3	15
<i>max_features</i>	'sqrt', None	
<i>min_samples_split</i>	0,002	0,01
<i>min_samples_leaf</i>	0,001	0,01

β)

<b>GradientBoostingRegressor</b>	
<b>Cross Validation Method:</b>	GridSearchCV(cv = 5, scoring = 'r2')
<b>Fixed Hyper-parameters:</b>	<i>loss</i> = 'ls', <i>random_state</i> = 0
<b>Tuned Hyper-parameters:</b>	<i>learning_rate</i> = 0.2 , <i>n_estimators</i> = 30 , <i>subsample</i> = 0.8 , <i>max_depth</i> = 5 , <i>max_features</i> = None = 2 , <i>min_samples_split</i> = 0.004 , <i>min_samples_leaf</i> = 0.0088
<b>CV Score (<math>R^2</math>):</b>	0,98910647
<b>Test Set <math>R^2</math> Score:</b>	0,98203823
<b>Test Set M.A.E. :</b>	0,040233

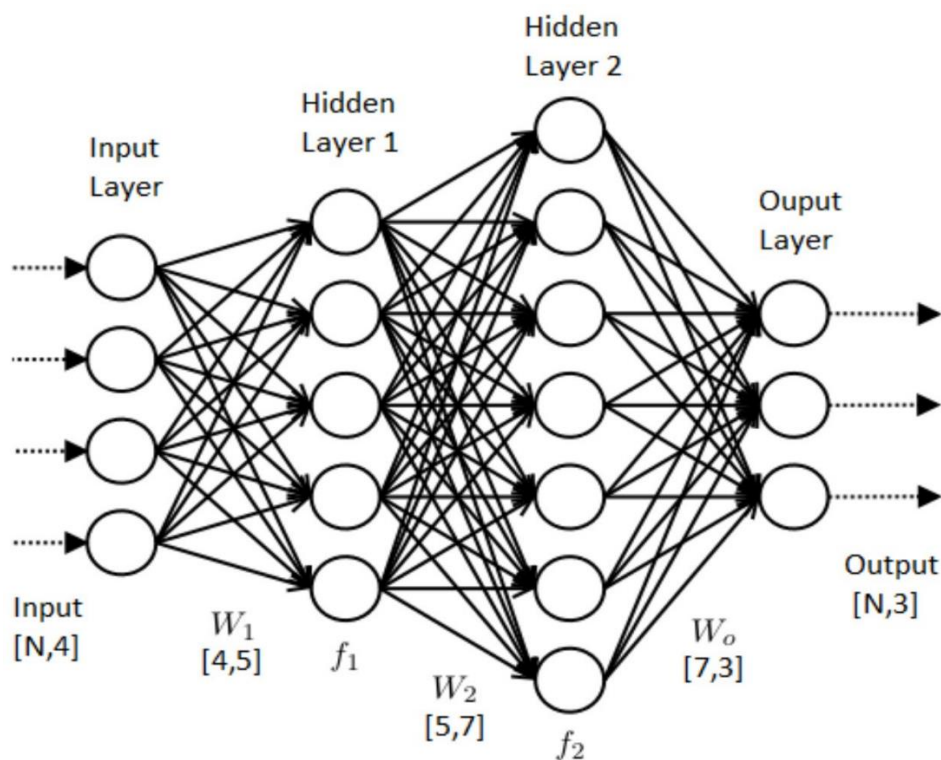
Παρατηρούμε ότι η μέθοδος Gradient Boosting, σε σύγκριση με τις υπόλοιπες μεθόδους συνόλων (Πίνακες 4.12 – 4.14), παρουσιάζει τον καλύτερο (υψηλότερο) συντελεστή προσδιορισμού, όπως αυτός προκύπτει από το Cross-Validation, αλλά και το υψηλότερο μέσο απόλυτο σφάλμα πρόβλεψης.

## 4.8 – Τεχνητά Νευρωνικά Δίκτυα

### 4.8.1 – Γενικά

Τα *Τεχνητά Νευρωνικά Δίκτυα* (*Artificial Neural Networks*) είναι υπολογιστικά μοντέλα που η λειτουργία τους είναι εμπνευσμένη από τον τρόπο που είναι οργανωμένο το βιολογικό νευρωνικό δίκτυο του εγκεφάλου. Τα μοντέλα αυτά έχουν ευρεία εφαρμογή, μεταξύ άλλων, και σε προβλήματα μηχανικής μάθησης, όπως προβλήματα ταξινόμησης και παλινδρόμησης.

Στο *Σχήμα 4.5* φαίνεται η δομή ενός τεχνητού νευρωνικού δικτύου. Η λειτουργία του νευρωνικού δικτύου βασίζεται σε ένα πλήθος διασυνδεδεμένων μονάδων που ονομάζονται *τεχνητοί νευρώνες* (*neurons*), και οι οποίοι είναι εμπνευσμένοι από μια απλοποιημένη εκδοχή των νευρώνων ενός βιολογικού εγκεφάλου. Κάθε διασύνδεση μεταφέρει ένα σήμα (συνήθως έναν πραγματικό αριθμό) από τον ένα νευρώνα στον άλλο, παίζοντας το ρόλο των συνάψεων του εγκεφάλου. Όταν ένας τεχνητός νευρώνας δέχεται ένα σήμα, το επεξεργάζεται και στη συνέχεια μπορεί να το προωθήσει στους νευρώνες που είναι συνδεδεμένοι μαζί του. Τυπικά, οι νευρώνες είναι οργανωμένοι σε διάφορα *επίπεδα* (*layers*): το πρώτο είναι το επίπεδο εισόδου (*input layer*) το οποίο δέχεται τις εισόδους του προβλήματος (δηλαδή τα χαρακτηριστικά των δειγμάτων). Στη συνέχεια βρίσκονται ένα ή περισσότερα επίπεδα που ονομάζονται *κρυφά επίπεδα* (*hidden layers*) και τα οποία μετασχηματίζουν το σήμα που δέχονται. Τέλος, έχουμε το επίπεδο εξόδου (*output layer*) το οποίο μετασχηματίζει τις τιμές που πήρε από το τελευταίο κρυφό επίπεδο και δημιουργεί τις εξόδους (*labels*) του προβλήματος. [21], [42]



*Σχήμα 4.5* – Η δομή ενός τεχνητού νευρωνικού δικτύου (*artificial neural network*).

Ανάλογα με το επίπεδο στο οποίο ανήκουν, οι τεχνητοί νευρώνες χωρίζονται σε τρεις κατηγορίες: τους νευρώνες εισόδου (επίπεδο εισόδου), τους υπολογιστικούς ή κρυφούς νευρώνες (κρυφά επίπεδα) και τους νευρώνες εξόδου (επίπεδο εξόδου). Οι *νευρώνες εισόδου* δεν επιτελούν κανένα μετασχηματισμό και αντιπροσωπεύουν απλώς τα χαρακτηριστικά (features) του δείγματος. Οι *υπολογιστικοί ή κρυφοί νευρώνες* πολλαπλασιάζουν κάθε είσοδό τους με το αντίστοιχο **συναπτικό βάρος** της  $w_i$  και υπολογίζουν το ολικό άθροισμα των γινομένων αυτών. Κάθε διασύνδεση έχει το δικό της συναπτικό βάρος το οποίο καθορίζει την ισχύ του σήματος που διέρχεται από τη διασύνδεση (τα βάρη αυτά συνήθως δε μένουν σταθερά αλλά αναπροσαρμόζονται κατά την εκπαίδευση του μοντέλου). Το άθροισμα των εισόδων του νευρώνα (με τα βάρη τους) διοχετεύεται στη **συνάρτηση ενεργοποίησης (activation function)** την οποία υλοποιεί εσωτερικά ο νευρώνας, και η τιμή της αποτελεί την έξοδο του νευρώνα για τις τρέχουσες εισόδους και βάρη. Τέλος, οι *νευρώνες εξόδου* δέχονται τις εξόδους του τελευταίου κρυφού επιπέδου και τις μετασχηματίζουν για να παραχθούν οι τελικές αριθμητικές εξοδοί του μοντέλου.

#### 4.8.2 – Multi-Layer Perceptron (MLP)

Το μοντέλο *Multi-Layer Perceptron (MLP)* είναι ένα τεχνητό νευρωνικό δίκτυο πολλών επιπέδων. Αποτελείται από ένα επίπεδο εισόδου, ένα επίπεδο εξόδου και ένα ή περισσότερα κρυφά επίπεδα. Λειτουργεί με τον τρόπο που περιγράφηκε στην προηγούμενη ενότητα και χρησιμοποιεί μη-γραμμικές συναρτήσεις ενεργοποίησης (activation functions) στους νευρώνες [32]. Όταν το MLP χρησιμοποιείται σε προβλήματα παλινδρόμησης, οι νευρώνες εξόδου δεν έχουν συνάρτηση ενεργοποίησης, και η συνάρτηση απωλειών που χρησιμοποιείται είναι η συνάρτηση τετραγωνικού σφάλματος:

$$L(\hat{y}, y, W) = \frac{1}{2} \|\hat{y} - y\|_2^2 + \frac{a}{2} \|W\|_2^2 \quad (4.10)$$

όπου  $a$  είναι η παράμετρος ομαλοποίησης (L2 κανονικοποίηση) που επιτελεί τον ίδιο ρόλο όπως στα μοντέλα παλινδρόμησης Ridge και Lasso: επιβάλλει ποινές στα υψηλά βάρη με στόχο την αποφυγή του Overfitting. Τα αρχικά συναπτικά βάρη των διασυνδέσεων μεταξύ των νευρώνων λαμβάνονται στην τύχη, και το μοντέλο MLP ελαχιστοποιεί τη συνάρτηση απωλειών αναπροσαρμόζοντας αυτά τα βάρη: αφού υπολογιστεί η τιμή της συνάρτησης απωλειών, ο αλγόριθμος διατρέχει τα επίπεδα προς τα πίσω, από το επίπεδο εξόδου προς το επίπεδο εισόδου, και δίνει σε κάθε συναπτικό βάρος μια νέα τιμή με στόχο τη μείωση του σφάλματος. Η τεχνική αυτή ονομάζεται **backpropagation** [59], [60]. Στην περίπτωση που η βελτιστοποίηση των βαρών γίνεται με τη μέθοδο gradient descent, υπολογίζεται η κλίση (παράγωγος) της συνάρτησης απωλειών ως προς τα βάρη [32]. Ο αλγόριθμος σταματάει τη διαδικασία βελτιστοποίησης όταν φτάσει έναν προκαθορισμένο μέγιστο αριθμό επαναλήψεων, ή όταν η περαιτέρω μείωση των απωλειών είναι πολύ μικρή.



Το μοντέλο του Scikit-learn που υλοποιεί το μοντέλο Multi-Layer Perceptron για προβλήματα παλινδρόμησης είναι το *neural\_network.MLPRegressor* [32]. Οι υπερ-παράμετροι του μοντέλου που πρέπει να ρυθμιστούν είναι οι εξής:

- **hidden\_layer\_sizes**: μια λίστα που το *i*-στο στοιχείο της είναι ο αριθμός των νευρώνων που θα υπάρχουν στο *i*-στο κρυφό επίπεδο. Ουσιαστικά, η παράμετρος αυτή καθορίζει τον αριθμό των κρυφών επιπέδων (ανάλογα με το πόσα στοιχεία περιλαμβάνει η λίστα) και τον αριθμό των νευρώνων σε κάθε ένα από αυτά.
- **activation**: η συνάρτηση ενεργοποίησης των νευρώνων που βρίσκονται στα κρυφά επίπεδα. Οι τιμές που θα δοκιμαστούν είναι οι 'tanh' για την υπερβολοειδή εφαπτομένη  $f(x) = \tanh(x)$ , και 'relu' για τη συνάρτηση  $f(x) = \max(0, x)$  (rectified linear unit function).
- **alpha**: η παράμετρος ομαλοποίησης (regularization term) για την L2 κανονικοποίηση. Χρησιμοποιείται για την αντιμετώπιση του φαινομένου Overfitting.
- **max\_iter**: ο μέγιστος αριθμός επαναλήψεων. Ο αλγόριθμος θα επαναλαμβάνει τη διαδικασία βελτιστοποίησης των συναπτικών βαρών μέχρι τη σύγκλιση, ή μέχρι να φτάσει αυτόν τον αριθμό.
- **solver**: η μέθοδος με την οποία θα πραγματοποιηθεί η βελτιστοποίηση των συναπτικών βαρών. Οι τιμές που θα δοκιμαστούν είναι οι 'sgd' (μέθοδος stochastic gradient descent) και 'adam' (μια βελτιστοποίηση της μεθόδου stochastic gradient descent [61]).
- **learning\_rate**: ο βαθμός στον οποίο κάθε επανάληψη του αλγορίθμου (αναπροσαρμογή των βαρών) συνεισφέρει στο συνολικό αποτέλεσμα. Χρησιμοποιείται μόνο όταν solver='sgd'. Οι τιμές που θα δοκιμαστούν είναι οι 'constant' (σταθερό ποσοστό συνεισφοράς), 'invscaling' (σταδιακή μείωση του ποσοστού συνεισφοράς) και 'adaptive' (το ποσοστό συνεισφοράς μειώνεται μόνο όταν δύο διαδοχικές επαναλήψεις δεν επιφέρουν ουσιαστική μείωση των απωλειών) [32].
- **momentum**: η παράμετρος αυτή παίρνει τιμές μεταξύ 0 και 1 και χρησιμοποιείται μόνο όταν solver='sgd'.

Για την εύρεση των βέλτιστων τιμών των υπερ-παραμέτρων θα χρησιμοποιηθεί ο αλγόριθμος GridSearchCV με 5-fold CV. Το μοντέλο MLPRegressor είναι εξαιρετικά ευαίσθητο στην μορφή των χαρακτηριστικών (features) των δειγμάτων, και προτείνεται αυτά να κανονικοποιούνται προτού χρησιμοποιηθούν [32]. Για την απαιτούμενη κανονικοποίηση του συνόλου δεδομένων χρησιμοποιούμε τη μέθοδο *MinMaxScaler* του Scikit-learn (βλ. Ενότητα 3.2.1). Στον Πίνακα 4.16α δίνεται το εύρος των τιμών των υπερ-παραμέτρων που χρησιμοποιήθηκαν για την αναζήτηση του βέλτιστου μοντέλου, και στον Πίνακα 4.16β παρουσιάζονται τα αποτελέσματα του μοντέλου.

#### 4. Δοκιμές Μοντέλων Παλινδρόμησης

**Πίνακας 4.16** – α) Εύρος τιμών των υπερ-παραμέτρων του μοντέλου *MLPRegressor* που εξετάστηκαν. β) Αξιολόγηση του μοντέλου *MLPRegressor*: βέλτιστες τιμές των υπερ-παραμέτρων, συντελεστής προσδιορισμού  $R^2$  και μέσο απόλυτο σφάλμα *MAE*.

α)

	Minimum Value	Maximum Value
<b>Hyper-parameter</b>		
<i>hidden_layer_sizes</i>	(2,2,2)	(100,100,100)
<i>activation</i>	'tanh', 'relu'	
<i>alpha</i>	$10^{-6}$	$10^{-1}$
<i>max_iter</i>	50	1000
<i>solver</i>	'adam', 'sgd'	
<i>learning_rate</i>	'constant', 'adaptive', 'invscaling'	
<i>momentum</i>	0,5	0,9

β)

MLPRegressor	
<b>Cross Validation Method:</b>	GridSearchCV(cv = 5, scoring = 'r2')
<b>Fixed Hyper-parameters:</b>	<i>random_state</i> = 0
<b>Tuned Hyper-parameters:</b>	<i>solver</i> = 'adam', <i>hidden_layer_sizes</i> = (55,55,55), <i>activation</i> = 'tanh', <i>alpha</i> = 0.018, <i>max_iter</i> = 100
<b>CV Score (<math>R^2</math>):</b>	0,98894312
<b>Test Set <math>R^2</math> Score:</b>	0,9817047
<b>Test Set M.A.E. :</b>	0,044038

## 4.9 – Σύγκριση Μοντέλων

Στον Πίνακα 4.17 της επόμενης σελίδας παρουσιάζονται συγκεντρωτικά τα αποτελέσματα για όλα τα μοντέλα που εξετάστηκαν στις ενότητες που προηγήθηκαν. Στον πίνακα φαίνονται, για κάθε μοντέλο, η μέθοδος Cross-Validation που χρησιμοποιήθηκε, οι βέλτιστες τιμές των υπερ-παραμέτρων του που προέκυψαν, καθώς και τα μεγέθη αξιολόγησης του μοντέλου: ο συντελεστής προσδιορισμού  $R^2$  που προκύπτει από το Cross-Validation, η αντίστοιχη τιμή του κατά την εφαρμογή στο σύνολο δοκιμής (test set) και η τιμή του μέσου απόλυτου σφάλματος της πρόβλεψης στο σύνολο δοκιμής.

Από τον πίνακα παρατηρούμε ότι όλα τα απλά γραμμικά μοντέλα έχουν παρόμοιες επιδόσεις μεταξύ τους, με ελαφρώς καλύτερες τιμές του συντελεστή προσδιορισμού  $R^2$  για τα μοντέλα Ridge, Lasso και ElasticNet. Γενικά όμως, οι επιδόσεις των γραμμικών μοντέλων είναι αρκετά χαμηλότερες σε σύγκριση με τα υπόλοιπα μοντέλα που δοκιμάστηκαν: για τα γραμμικά μοντέλα ο συντελεστής προσδιορισμού είναι της τάξεως του 0,88 και το μέσο απόλυτο σφάλμα της τάξεως του 0,225, σε αντιδιαστολή με τα πιο πολύπλοκα μοντέλα όπου ο συντελεστής προσδιορισμού είναι της τάξεως του 0,98 και το μέσο απόλυτο σφάλμα της τάξεως του 0,04. Μεταξύ των μοντέλων που βασίζονται στις μεθόδους συνόλων (ensemble methods) την καλύτερη συνολική συμπεριφορά την παρουσιάζει το μοντέλο Random Forest, το οποίο χαρακτηρίζεται από υψηλό συντελεστή προσδιορισμού και χαμηλό μέσο απόλυτο σφάλμα πρόβλεψης.

Παρατηρούμε επίσης ότι για όλα τα μοντέλα, η τιμή του συντελεστή προσδιορισμού, όπως αυτός προκύπτει από το Cross-Validation του GridSearchCV, είναι ελαφρώς υψηλότερη από την τιμή του όταν το μοντέλο εφαρμόζεται στο σύνολο δοκιμής. Αυτό είναι αναμενόμενο, γιατί η πρώτη τιμή αναφέρεται στα δεδομένα με τα οποία εκπαιδεύεται το μοντέλο και αποτελεί μια αισιόδοξη εκτίμηση της πραγματικής τιμής του συντελεστή προσδιορισμού όταν το μοντέλο κληθεί να εκτελέσει πρόβλεψη σε καινούρια δεδομένα, όπως αυτά του συνόλου δοκιμής. Παρόλα αυτά, η χρήση της τεχνικής Cross-Validation κατά την εκπαίδευση συμβάλει ώστε οι δύο τιμές να είναι αρκετά κοντά.

Το μοντέλο που παρουσιάζει τον καλύτερο (υψηλότερο) συντελεστή προσδιορισμού  $R^2$ , όπως αυτός προκύπτει από την τεχνική Cross-Validation, είναι ο παλινδρομητής Kernel Ridge. Το χαμηλότερο όμως μέσο απόλυτο σφάλμα πρόβλεψης το παρουσιάζει το μοντέλο παλινδρόμησης Διανυσμάτων Υποστήριξης (Support Vector Regression – SVR). Το βασικό κριτήριο που θα χρησιμοποιήσουμε για την επιλογή των βέλτιστων μοντέλων είναι το μέσο απόλυτο σφάλμα, όπως αναλύθηκε στην Ενότητα 3.3. Από αυτή την άποψη, τα καλύτερα μοντέλα είναι αυτά που δίνουν τις χαμηλότερες τιμές του μέσου απόλυτου σφάλματος, επομένως από τον Πίνακα 4.17 βλέπουμε ότι τα μοντέλα αυτά είναι τα **SVR (Support Vector Regression)**, **K-Neighbors** και **Random Forest**.

*Πίνακας 4.17 (επόμενη σελίδα) – Συγκεντρωτικός πίνακας με τα αποτελέσματα για όλα τα μοντέλα παλινδρόμησης που δοκιμάστηκαν.*

Regressor	Cross-Validation Method	Fixed Parameters	Tuned Parameters	R2 Score	Test Set R2 Score	Mean Absolute Error
The model used for regression	The cross-validation method used for tuning the model's hyper-parameters	The model's parameters with a fixed value during the tuning	The optimal values for the model's hyper-parameters that are tuned with cross-validation	The R2 Score of the optimal model, calculated with cross-validation	The Score of the optimal model for predictions on the Test Set	The Mean Absolute Error of the prediction on the Test Set
<b>Generalized Linear Models</b>						
Ordinary Least Squares (Linear Regression)	GridSearchCV( cv=5, scoring=default=r2)			0.88945248	0.883054	0.2253
Ridge Regression	RidgeCV( cv=GCV, scoring=default=r2)		alpha = 0.01	0.88967542	0.883054	0.2253
Lasso	LassoCV( cv = 3)		alpha = 10^-7	0.88967542	0.883054	0.2253
Elastic-Net	ElasticNetCV( cv = 3)		alpha = 10^-7, l1_ratio = 1	0.88967542	0.883054	0.2253
Bayesian Ridge Regression	GridSearchCV( cv=5, scoring=default=r2)		a1 = 10^-10, a2 = 0.01, l1 = 0.01, l2 = 10^-10	0.88945249	0.883054	0.2253
Stochastic Gradient Descent (SGD) Regressor	GridSearchCV( cv=5, scoring=default=r2)	loss = 'squared_loss', random_state = 0	max_iter = 100, alpha = 10^-5, l1_ratio = 0, penalty = 'l1'	0.8894593	0.883059	0.22587
<b>Support Vector Machines (SVM)</b>						
Support Vector Regression (SVR)	GridSearchCV( cv=5, scoring=default=r2)		kernel = 'rbf', gamma = 50, C = 100, epsilon = 0.01	0.98922228	0.98097841	0.037046
<b>Kernel Ridge Regression</b>						
Kernel Ridge	GridSearchCV( cv=5, scoring=default=r2)		kernel = 'rbf', gamma = 10, alpha = 0.001	0.98944644	0.98210475	0.039324
<b>Nearest Neighbors</b>						
Kneighbors Regressor	GridSearchCV( cv=5, scoring=default=r2)		algorithm = 'ball_tree', leaf_size = 1, n_neighbors = 10, p = 1, weights = 'distance'	0.9882918	0.98197127	0.037368
RadiuNeighbors Regressor	GridSearchCV( cv=5, scoring=default=r2)		algorithm = 'ball_tree', leaf_size = 1, radius = 0.1, p = 1, weights = 'distance'	0.98786442	0.98136625	0.046482
<b>Decision Trees</b>						
Decision Tree Regressor	GridSearchCV( cv=5, scoring=default=r2)	criterion = 'mae', max_features = None(=2), random_state = 0	max_depth = 10, min_samples_split = 0.0047, min_samples_leaf = 0.0017	0.98871212	0.98146038	0.037619
<b>Ensemble Methods</b>						
Random Forest Regressor	GridSearchCV( cv=5, scoring=default=r2)	criterion = 'mae', max_features = None(=2), random_state = 0	max_depth = 10, min_samples_split = 0.002, min_samples_leaf = 0.0005, n_estimators = 100	0.98903021	0.98136952	0.037354
Extra Trees Regressor (Extremely Randomized Trees)	GridSearchCV( cv=5, scoring=default=r2)	criterion = 'mae', max_features = None(=2), random_state = 0	max_depth = None, min_samples_split = 0.003, min_samples_leaf = 0.001, n_estimators = 120	0.98894504	0.98251049	0.038221
AdaBoost Regressor	GridSearchCV( cv=5, scoring=default=r2)	base_estimator = DecisionTreeRegressor (criterion='mae', max_features=None, max_depth=10, min_samples_split=0.0047, min_samples_leaf=0.0017, random_state=0), random_state = 0	loss = 'linear', learning_rate = 0.02, n_estimators = 84	0.98894535	0.98115137	0.037528
Gradient Boosting Regressor	GridSearchCV( cv=5, scoring=default=r2)	loss = 'ls', random_state = 0	learning_rate = 0.2, n_estimators = 30, subsample = 0.8, max_depth = 5, max_features = None, min_samples_split = 0.004, min_samples_leaf = 0.0088	0.98910647	0.98203823	0.040233
<b>Neural Networks</b>						
Multi-Layer Perceptron Regressor	GridSearchCV( cv=5, scoring=default=r2)	random_state = 0	solver = 'adam', hidden_layer_sizes = (55,55,55), activation = 'tanh', alpha = 0.018, max_iter = 100	0.98894312	0.9817047	0.044038

## ΚΕΦΑΛΑΙΟ 5 – ΚΑΜΠΥΛΕΣ ΕΚΠΑΙΔΕΥΣΗΣ ΒΕΛΤΙΣΤΩΝ ΜΟΝΤΕΛΩΝ

---

### 5.1 – Εισαγωγή

Στο προηγούμενο κεφάλαιο εκπαιδεύτηκαν διάφορα μοντέλα παλινδρόμησης, με σκοπό την εύρεση εκείνων που είναι πιο κατάλληλα για το πρόβλημά μας, δηλαδή την πρόβλεψη της ισχύος ενός αιολικού πάρκου. Προέκυψε τελικά (βλ. και *Ενότητα 4.9*) ότι τα καλύτερα μοντέλα για αυτή την περίπτωση, δηλαδή αυτά που παρουσιάζουν το χαμηλότερο μέσο απόλυτο σφάλμα πρόβλεψης (mean absolute error), είναι τα **SVR (Support Vector Regression)**, **K-Neighbors** και **Random Forest**.

Σε αυτό το κεφάλαιο θα εξεταστούν τα μοντέλα αυτά πιο διεξοδικά, με σκοπό να προσδιορίσουμε την επίδραση που έχουν οι υπερ-παράμετροι του κάθε μοντέλου στο μέσο απόλυτο σφάλμα της πρόβλεψης, αλλά και στον χρόνο εκπαίδευσης (training time) του μοντέλου. Ο *χρόνος εκπαίδευσης* των μοντέλων εξαρτάται γενικά σε μεγάλο βαθμό από το πόσο ισχυρό είναι το υπολογιστικό σύστημα στο οποίο γίνεται η εκπαίδευση, παρόλα αυτά υπάρχουν και αντικειμενικές διαφορές από μοντέλο σε μοντέλο, καθώς πιο πολύπλοκα μοντέλα (για παράδειγμα ένα νευρωνικό δίκτυο με μεγάλο αριθμό νευρώνων ή ένα μοντέλο Random Forest με μεγάλο αριθμό επιμέρους δέντρων) χρειάζονται περισσότερο χρόνο για να εκπαιδευτούν. Για εφαρμογές μεγάλης κλίμακας όπου ο χρόνος αποτελεί σημαντικό παράγοντα, οι πιο γρήγοροι αλγόριθμοι έχουν πλεονέκτημα [62].

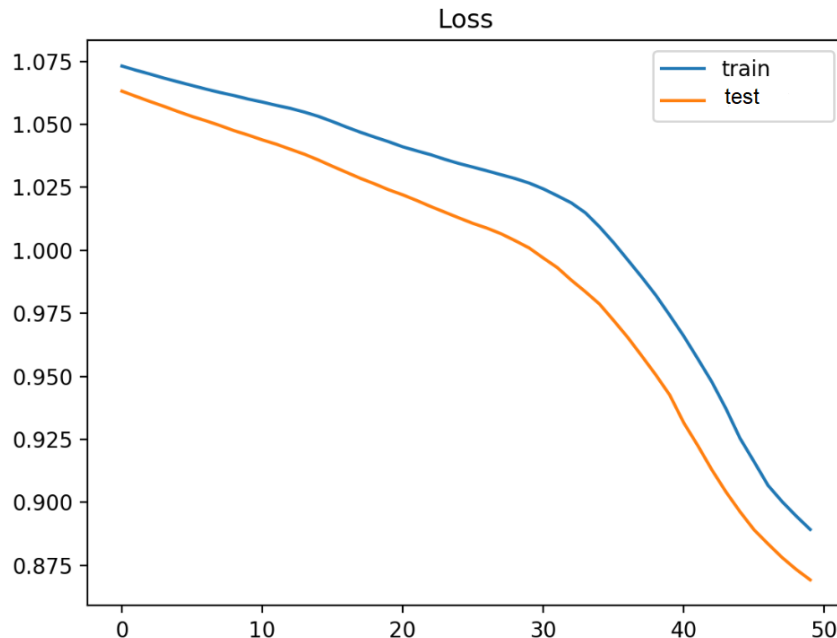
Για τον σκοπό αυτό, θα κατασκευαστούν για κάθε μοντέλο δύο ειδών γραφικές παραστάσεις:

- Οι *καμπύλες εκπαίδευσης* του μοντέλου: οι καμπύλες εκπαίδευσης δείχνουν πώς μεταβάλλεται η συμπεριφορά του μοντέλου όταν μεταβάλλεται η τιμή μιας υπερ-παραμέτρου του μοντέλου [63], [64]. Όπως προαναφέρθηκε, το κριτήριο που χρησιμοποιούμε σε αυτή την εργασία για την εκτίμηση ενός μοντέλου είναι το μέσο απόλυτο σφάλμα πρόβλεψης, επομένως η κάθε καμπύλη εκπαίδευσης θα απεικονίζει τη μεταβολή του μέσου απόλυτου σφάλματος με την τιμή μιας υπερ-παραμέτρου του μοντέλου. Επίσης, η εκτίμηση θα γίνει τόσο για το σύνολο εκπαίδευσης (training set) όσο και για το σύνολο δοκιμής (test set), επομένως για κάθε υπερ-πάρμετρο θα έχουμε δύο καμπύλες: η καμπύλη που αναφέρεται στο σύνολο εκπαίδευσης δείχνει το πόσο καλά εκπαιδεύεται το μοντέλο, ενώ η καμπύλη που αναφέρεται στο σύνολο δοκιμής δίνει μια ιδέα της ικανότητας του μοντέλου να γενικεύει σε νέα δεδομένα. Η σύγκριση των δύο αυτών καμπυλών μπορεί επίσης να μας δώσει στοιχεία για τη συμπεριφορά του μοντέλου ως προς τα φαινόμενα **Overfitting** (όταν το μοντέλο έχει εξειδικευτεί υπερβολικά στο σύνολο εκπαίδευσης, με αποτέλεσμα να μην μπορεί να γενικεύσει αποτελεσματικά σε νέα δεδομένα) και **Underfitting** (όταν το μοντέλο δεν μπορεί να πετύχει χαμηλό σφάλμα κατά την εκπαίδευσή του, δηλαδή δεν μπορεί να προσαρμοστεί στο σύνολο εκπαίδευσης) [65]. Το φαινόμενο **Underfitting** μπορεί να ανιχνευτεί από την καμπύλη που αναφέρεται στο σύνολο εκπαίδευσης, όταν το

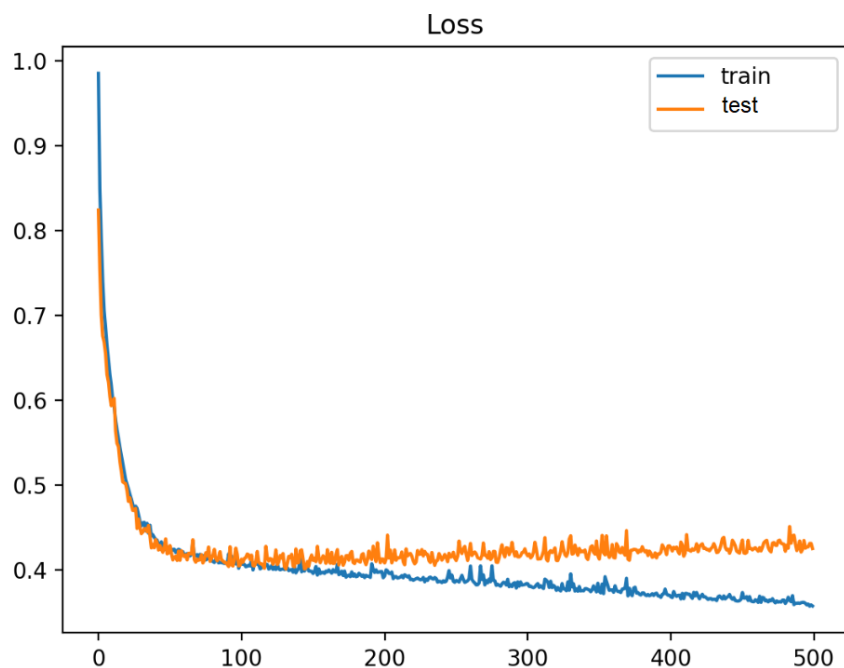
## 5. Καμπύλες Εκπαίδευσης Βέλτιστων Μοντέλων

σφάλμα είναι πολύ υψηλό ή δεν παρουσιάζει καμία μεταβολή κατά την εκπαίδευση (Σχήμα 5.1-α). Το φαινόμενο Overfitting ανιχνεύεται όταν η μείωση του σφάλματος που αναφέρεται στο σύνολο εκπαίδευσης συνοδεύεται από αύξηση του σφάλματος στο σύνολο δοκιμής, δηλαδή όταν οι δύο καμπύλες εκπαίδευσης απομακρύνονται μεταξύ τους (Σχήμα 5.1-β).

- Οι καμπύλες του χρόνου εκπαίδευσης, που απεικονίζουν το πώς μεταβάλλεται ο χρόνος εκπαίδευσης του μοντέλου με την τιμή των υπερ-παραμέτρων του.



(α)



(β)

**Σχήμα 5.1** – α) Φαινόμενο Underfitting: για χαμηλές τιμές της προς ρύθμιση υπερ-παραμέτρου, το σφάλμα είναι πολύ υψηλό και το μοντέλο δεν μπορεί να «μάθει» το σύνολο εκπαίδευσης [65]. β) Φαινόμενο Overfitting: το σημείο στο οποίο οι δύο καμπύλες αρχίζουν να απομακρύνονται μεταξύ τους σηματοδοτεί την έναρξη του Overfitting [65].

## 5.2 – Καμπύλες Εκπαίδευσης Βέλτιστων Μοντέλων

### 5.2.1 – Support Vector Regressor (SVR)

Το μοντέλο παλινδρόμησης Διανυσμάτων Υποστήριξης (Support Vector Regression – SVR) εξετάστηκε στην *Ενότητα 4.4*, από όπου προέκυψε ότι οι βέλτιστες τιμές των υπερ-παραμέτρων του είναι:

- **kernel** = 'rbf'
- **gamma** = 50
- **C** = 100
- **epsilon** = 0,01

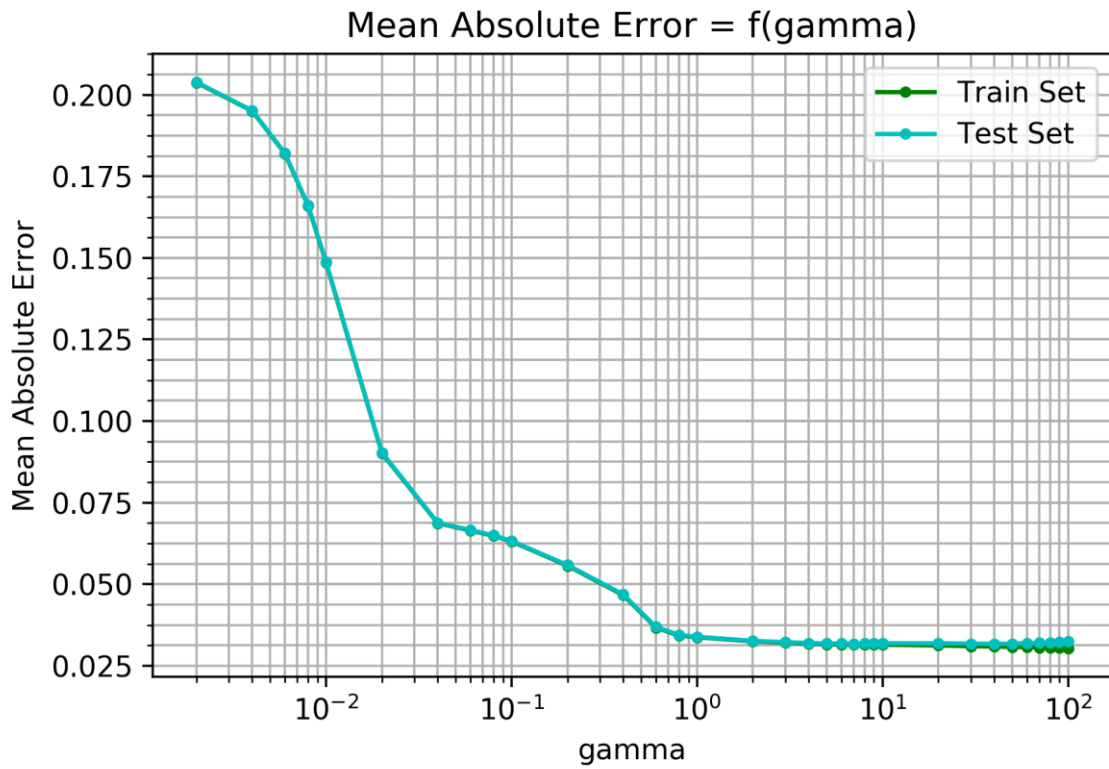
Η αναζήτηση των βέλτιστων υπερ-παραμέτρων έγινε με χρήση του αλγορίθμου GridSearchCV, με 5-fold CV, και με κριτήριο τη μεγιστοποίηση του συντελεστή προσδιορισμού  $R^2$ . Στη συνέχεια, με χρήση των καμπυλών εκπαίδευσης, θα εξεταστεί η επίδραση που έχει η κάθε υπερ-παραμέτρος στο μέσο απόλυτο σφάλμα πρόβλεψης (mean absolute error) και στο χρόνο εκπαίδευσης του μοντέλου, και θα διαπιστωθεί αν οι παραπάνω τιμές είναι οι βέλτιστες και ως προς αυτά τα κριτήρια.

#### gamma

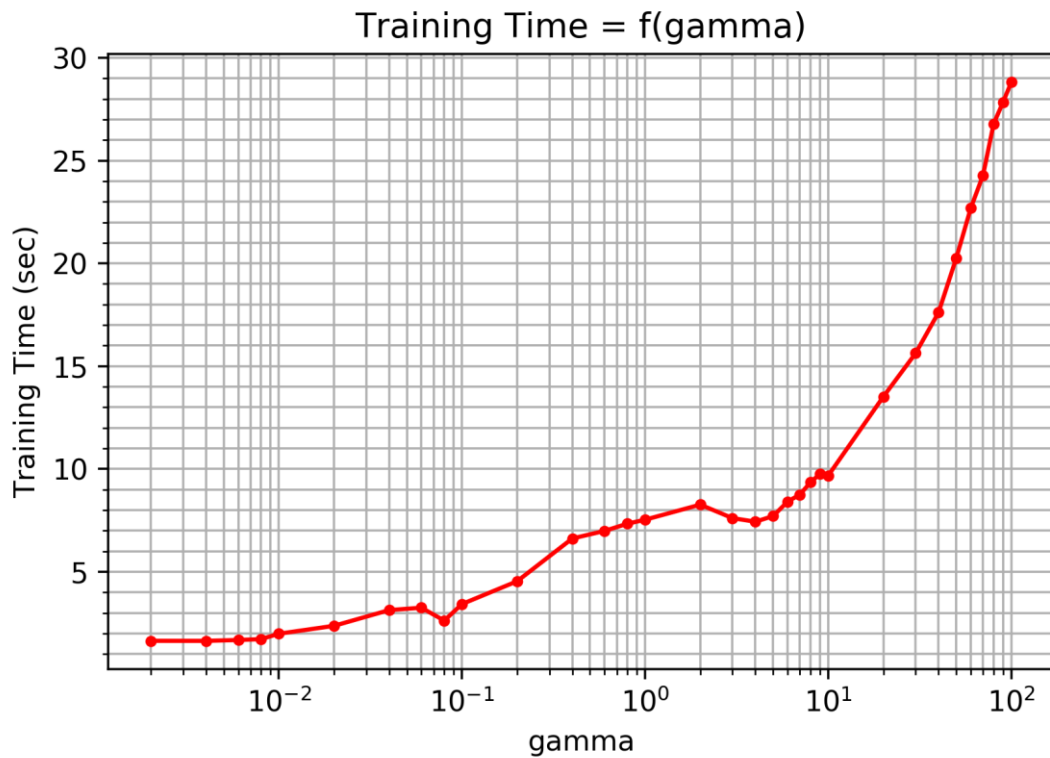
Στο *Σχήμα 5.2-α* παρουσιάζονται οι καμπύλες εκπαίδευσης που αφορούν τα σύνολα εκπαίδευσης και δοκιμής ως προς την τιμή της υπερ-παραμέτρου gamma. Από το σχήμα αυτό παρατηρούμε ότι η αύξηση του gamma προκαλεί μείωση του σφάλματος, μέχρι ένα σημείο ισορροπίας, περίπου για  $\text{gamma} = 4$ , πέρα από το οποίο η περαιτέρω αύξησή του δεν έχει επίδραση στο σφάλμα. Βλέπουμε επίσης ότι για όλο το εύρος των τιμών του gamma που εξετάστηκαν, οι καμπύλες για τα σύνολα εκπαίδευσης και δοκιμής σχεδόν ταυτίζονται, επομένως δεν έχουμε εμφάνιση Overfitting. Για τις μεγάλες μόνο τιμές του gamma, της τάξεως του 100, μπορούμε να παρατηρήσουμε ότι οι 2 καμπύλες αρχίζουν ελαφρώς να αποκλίνουν, επομένως περαιτέρω αύξηση του gamma πιθανώς να οδηγήσει σε Overfitting.

Στο *Σχήμα 5.2-β* παρουσιάζεται η μεταβολή του χρόνου εκπαίδευσης του μοντέλου ως προς την υπερ-παραμέτρο gamma. Παρατηρούμε ότι γενικά ο χρόνος εκπαίδευσης του μοντέλου αυξάνεται με την αύξηση του gamma, και μάλιστα πολύ πιο απότομα για τιμές του gamma πάνω από 10.

Από τα δύο αυτά σχήματα μπορούμε να συμπεράνουμε ότι η τιμή **gamma = 50** που επιλέχθηκε είναι αποδεκτή ως βέλτιστη, καθώς οδηγεί στο ελάχιστο μέσο απόλυτο σφάλμα με ικανοποιητικό χρόνο εκπαίδευσης. Στην περίπτωση που ο χρόνος εκπαίδευσης ήταν σημαντικό κριτήριο, θα μπορούσαμε ενδεχομένως να επιλέξουμε και λίγο μικρότερο gamma, για παράδειγμα 10, το οποίο δίνει ίδια τιμή του μέσου απόλυτου σφάλματος αλλά ελαφρώς μικρότερο χρόνο εκπαίδευσης.



(α)



(β)

Σχήμα 5.2 – α) Καμπύλες εκπαίδευσης του μοντέλου SVR ως προς την τιμή της υπερ-παραμέτρου  $\text{gamma}$ . β) Χρόνος εκπαίδευσης του μοντέλου SVR ως προς την τιμή της υπερ-παραμέτρου  $\text{gamma}$ .

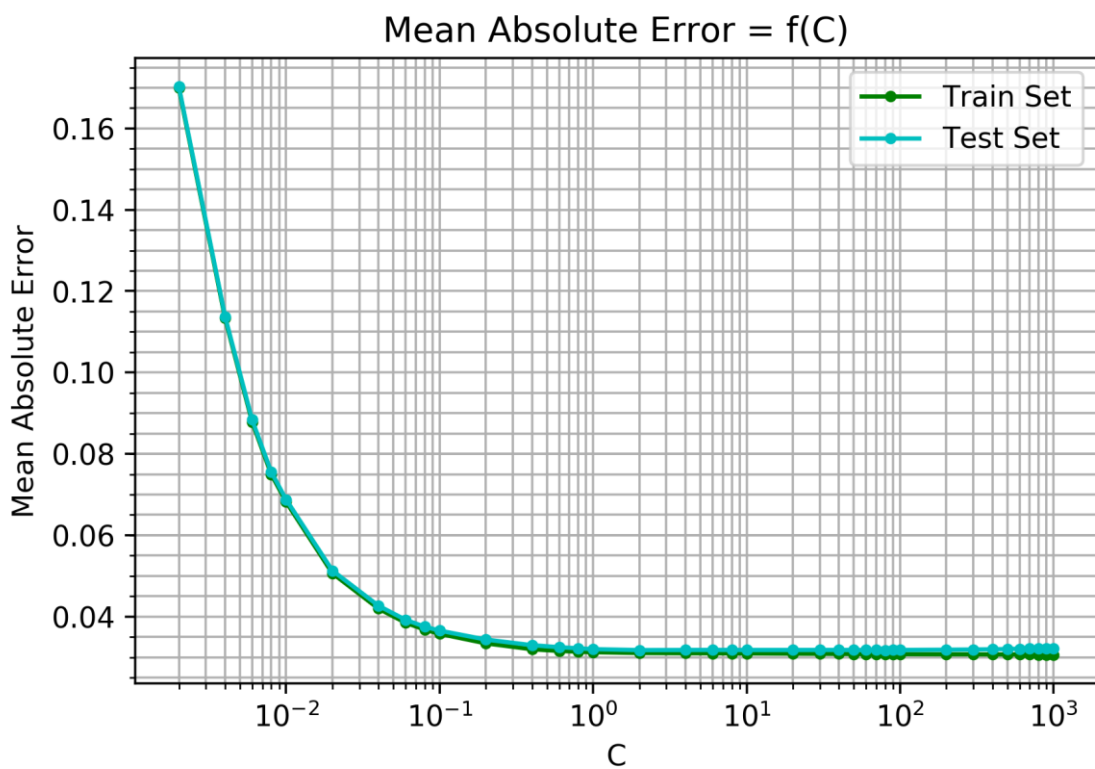


C

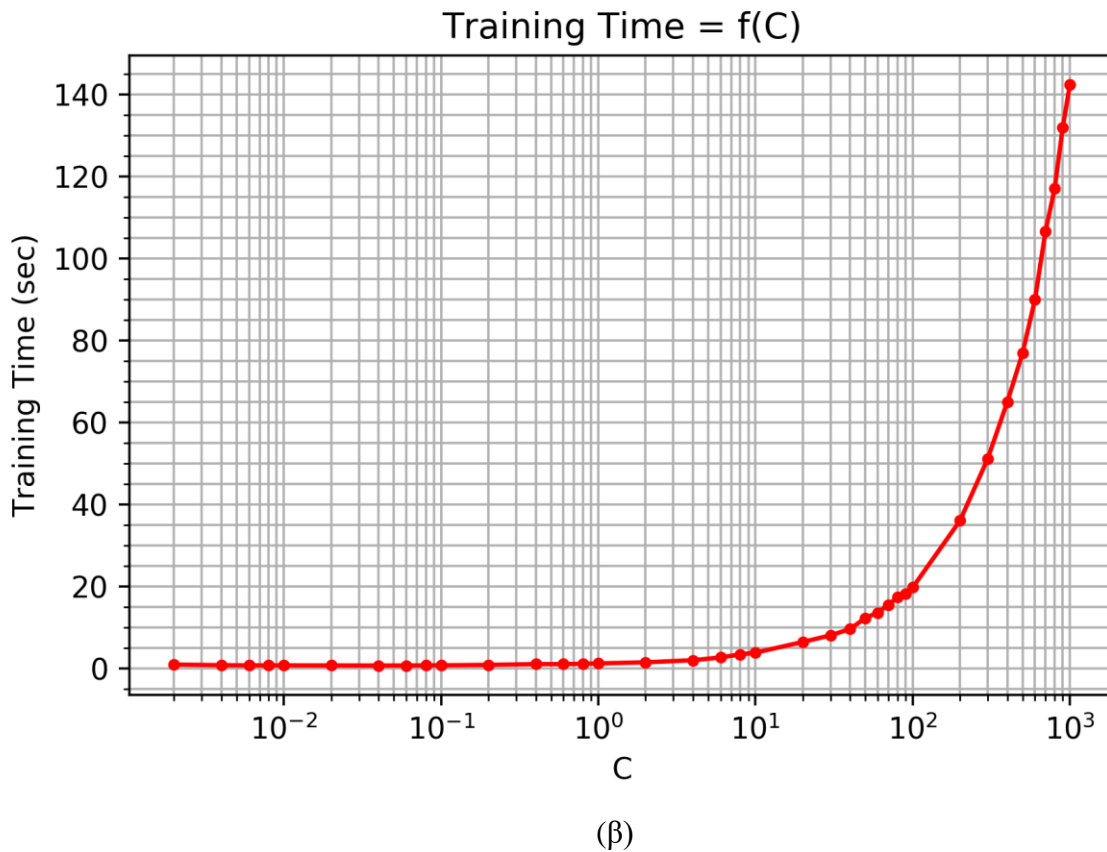
Στο Σχήμα 5.3-α παρουσιάζονται οι καμπύλες εκπαίδευσης που αφορούν τα σύνολα εκπαίδευσης και δοκιμής ως προς την τιμή της υπερ-παραμέτρου  $C$ . Από το σχήμα αυτό παρατηρούμε ότι η αύξηση του  $C$  προκαλεί μείωση του σφάλματος, μέχρι ένα σημείο ισορροπίας, περίπου για  $C = 10$ , πέρα από το οποίο η περαιτέρω αύξηση δεν έχει επίδραση στο σφάλμα. Βλέπουμε επίσης ότι για όλο το εύρος των τιμών του  $C$  που εξετάστηκαν, οι καμπύλες για τα σύνολα εκπαίδευσης και δοκιμής σχεδόν ταυτίζονται, επομένως δεν έχουμε εμφάνιση Overfitting.

Στο Σχήμα 5.3-β παρουσιάζεται η μεταβολή του χρόνου εκπαίδευσης του μοντέλου ως προς την υπερ-παραμέτρο  $C$ . Παρατηρούμε ότι για τιμές περίπου μέχρι  $C = 1$  ο χρόνος εκπαίδευσης είναι πολύ μικρός και δε μεταβάλλεται, για μεγαλύτερες όμως τιμές του  $C$  ο χρόνος εκπαίδευσης αυξάνεται σχεδόν εκθετικά.

Από τα δύο αυτά σχήματα μπορούμε να συμπεράνουμε ότι η τιμή  $C = 100$  που επιλέχθηκε είναι αποδεκτή ως βέλτιστη, καθώς οδηγεί στο ελάχιστο μέσο απόλυτο σφάλμα με ικανοποιητικό χρόνο εκπαίδευσης. Και πάλι στην περίπτωση που ο χρόνος εκπαίδευσης ήταν σημαντικό κριτήριο, θα μπορούσαμε να επιλέξουμε μικρότερο  $C$ , για παράδειγμα 10, το οποίο δίνει περίπου ίδια τιμή του μέσου απόλυτου σφάλματος, αλλά ελαφρώς μικρότερο χρόνο εκπαίδευσης.



(α)



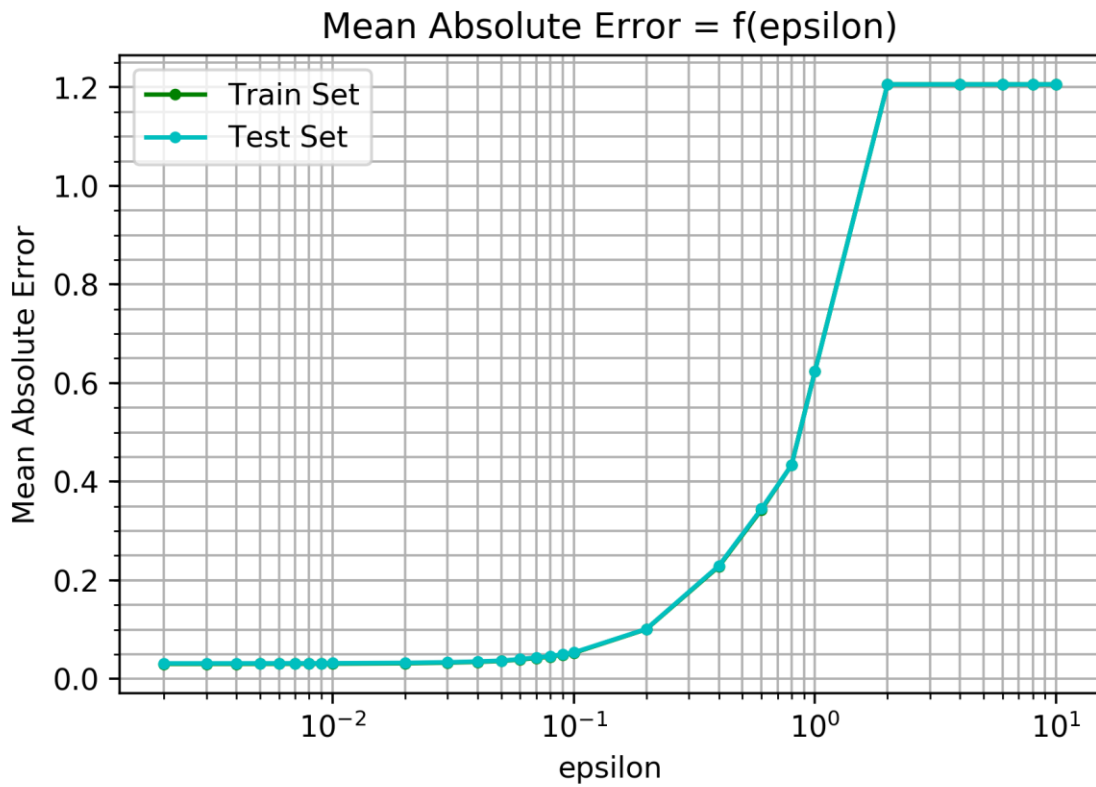
**Σχήμα 5.3** – α) Καμπύλες εκπαίδευσης του μοντέλου SVR ως προς την τιμή της υπερ-παραμέτρου C. β) Χρόνος εκπαίδευσης του μοντέλου SVR ως προς την τιμή της υπερ-παραμέτρου C.

### epsilon

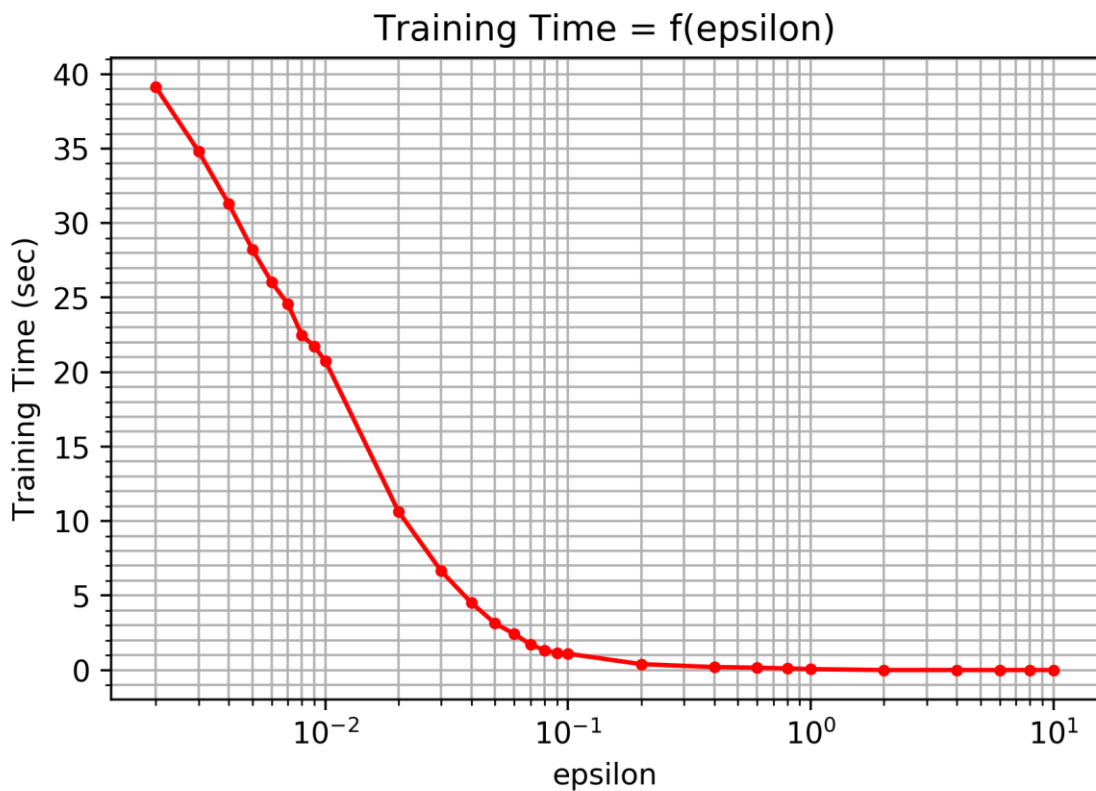
Στο Σχήμα 5.4-α παρουσιάζονται οι καμπύλες εκπαίδευσης που αφορούν τα σύνολα εκπαίδευσης και δοκιμής ως προς την τιμή της υπερ-παραμέτρου epsilon. Από το σχήμα αυτό παρατηρούμε ότι η αύξηση του epsilon πάνω από την τιμή 0,1 προκαλεί εκθετική αύξηση του σφάλματος, κάτι που είναι αναμενόμενο καθώς το epsilon στο μοντέλο SVR καθορίζει το εύρος της περιοχής μέσα στην οποία δε λαμβάνονται υπόψη τα σφάλματα, και μεγάλες τιμές του οδηγούν σε μοντέλο με μεγάλη ανεκτικότητα στα λάθη. Παρατηρούμε επίσης ότι για όλο το εύρος των τιμών του epsilon που εξετάστηκαν, οι καμπύλες για τα σύνολα εκπαίδευσης και δοκιμής ταυτίζονται απόλυτα, επομένως δεν έχουμε εμφάνιση Overfitting.

Στο Σχήμα 5.4-β παρουσιάζεται η μεταβολή του χρόνου εκπαίδευσης του μοντέλου ως προς την υπερ-παραμέτρο epsilon. Παρατηρούμε ότι με την αύξηση του epsilon ο χρόνος εκπαίδευσης μειώνεται, μέχρι την τιμή epsilon = 0,1 πέρα από την οποία παραμένει σχεδόν σταθερός σε μια χαμηλή τιμή.

Από τα δύο αυτά σχήματα μπορούμε να συμπεράνουμε ότι η τιμή **epsilon = 0,01** που επιλέχθηκε είναι αποδεκτή ως βέλτιστη, καθώς οδηγεί σε ελαχιστοποίηση του μέσου απόλυτου σφάλματος με ικανοποιητικό χρόνο εκπαίδευσης.



(α)



(β)

Σχήμα 5.4 – α) Καμπύλες εκπαίδευσης του μοντέλου SVR ως προς την τιμή της υπερ-παραμέτρου  $\epsilon$ . β) Χρόνος εκπαίδευσης του μοντέλου SVR ως προς την τιμή της υπερ-παραμέτρου  $\epsilon$ .

### 5.2.2 – KNeighbors Regressor

Το μοντέλο παλινδρόμησης που βασίζεται στην τεχνική k-κοντινότερων γειτόνων (KNeighbors Regressor) εξετάστηκε στην *Ενότητα 4.5.2*, από όπου προέκυψε ότι οι βέλτιστες τιμές των υπερ-παραμέτρων του είναι:

- ***algorithm*** = 'ball\_tree'
- ***leaf\_size*** = 1
- ***n\_neighbors*** = 10
- ***weights*** = 'distance'
- ***p*** = 1

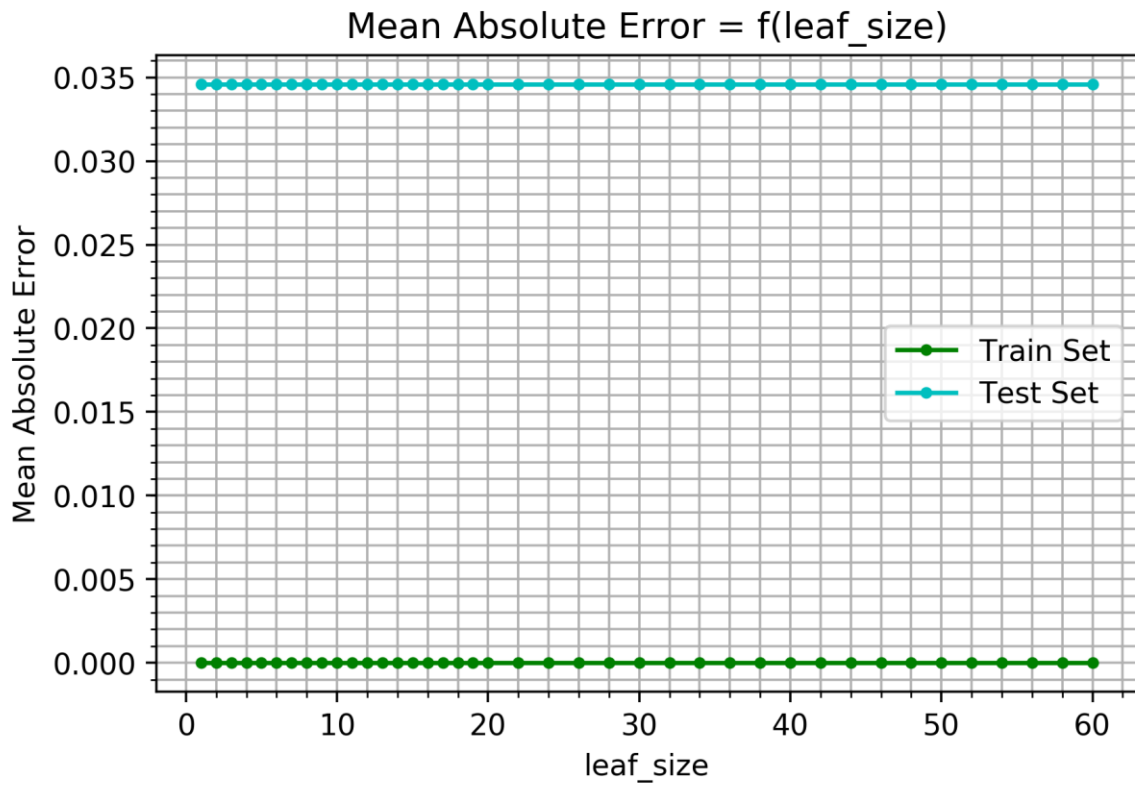
Η αναζήτηση των βέλτιστων υπερ-παραμέτρων έγινε με χρήση του αλγορίθμου GridSearchCV, με 5-fold CV, και με κριτήριο τη μεγιστοποίηση του συντελεστή προσδιορισμού  $R^2$ . Στη συνέχεια, με χρήση των καμπυλών εκπαίδευσης, θα εξεταστεί η επίδραση που έχει η κάθε υπερ-παραμέτρος στο μέσο απόλυτο σφάλμα πρόβλεψης (mean absolute error) και στο χρόνο εκπαίδευσης του μοντέλου, και θα διαπιστωθεί αν οι παραπάνω τιμές είναι οι βέλτιστες και ως προς αυτά τα κριτήρια.

#### *leaf\_size*

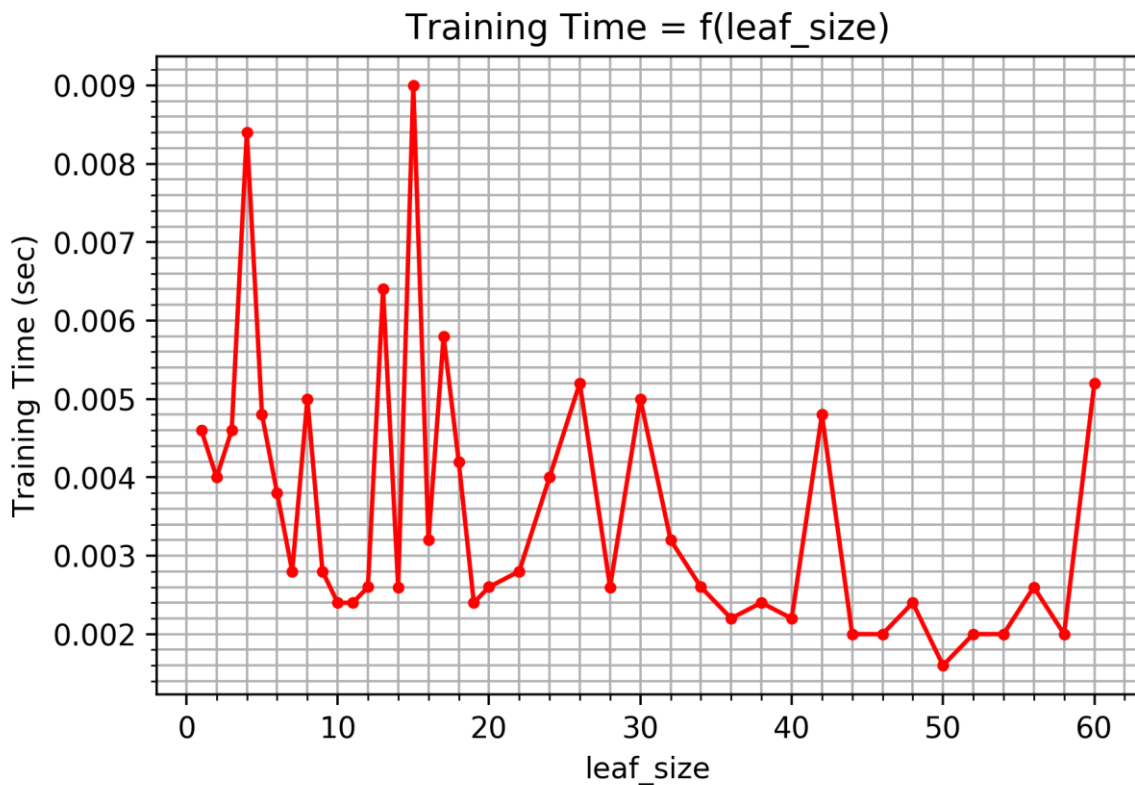
Στο *Σχήμα 5.5-α* παρουσιάζονται οι καμπύλες εκπαίδευσης που αφορούν τα σύνολα εκπαίδευσης και δοκιμής ως προς την τιμή της υπερ-παραμέτρου leaf\_size. Από το σχήμα αυτό παρατηρούμε ότι η μεταβολή του leaf\_size δεν έχει ουσιαστική επίδραση στο μέσο απόλυτο σφάλμα, τουλάχιστον για το εύρος τιμών που δοκιμάστηκαν. Βλέπουμε επίσης ότι το σφάλμα για το σύνολο δοκιμής είναι σταθερά σε μια υψηλότερη τιμή από το σφάλμα του συνόλου εκπαίδευσης: αυτή η διαφορά ονομάζεται *χάσμα γενίκευσης (generalization gap)* και αποτελεί συνηθισμένο φαινόμενο στα μοντέλα μηχανικής μάθησης, καθώς το μοντέλο προσαρμόζεται στα δεδομένα με τα οποία εκπαιδεύεται και σχεδόν πάντα η απόδοσή του είναι μειωμένη όταν προσπαθεί να γενικεύσει σε νέα δεδομένα. Παρόλα αυτά, παρατηρούμε ότι για όλο το εύρος των τιμών που εξετάστηκαν, το χάσμα αυτό μεταξύ των δύο καμπυλών παραμένει σταθερό, επομένως συμπεραίνουμε ότι δεν έχουμε εμφάνιση του φαινομένου Overfitting.

Στο *Σχήμα 5.5-β* παρουσιάζεται η μεταβολή του χρόνου εκπαίδευσης του μοντέλου ως προς την υπερ-παραμέτρο leaf\_size. Παρατηρούμε ότι για όλο εύρος των τιμών που εξετάστηκαν ο χρόνος εκπαίδευσης παρουσιάζει κάποιες μικρές διακυμάνσεις (της τάξεως των msec), αλλά πρακτικά παραμένει σταθερός σε μια χαμηλή τιμή.

Από τα δύο αυτά σχήματα μπορούμε να συμπεράνουμε ότι η τιμή ***leaf\_size*** = 1 που επιλέχθηκε είναι αποδεκτή ως βέλτιστη, καθώς οδηγεί στο ελάχιστο μέσο απόλυτο σφάλμα και σε γρήγορη εκπαίδευση. Πρακτικά θα μπορούσαμε να επιλέξουμε και μεγαλύτερες τιμές του leaf\_size, πχ 50, οι οποίες θα οδηγούσαν σε απλούστερη μορφή της δενδροειδούς δομής που χρησιμοποιείται από τον αλγόριθμο 'ball\_tree', χωρίς να παρουσιάζεται αύξηση του μέσου απόλυτου σφάλματος, όπως φαίνεται στο *Σχήμα 5.5-α*.



(α)



(β)

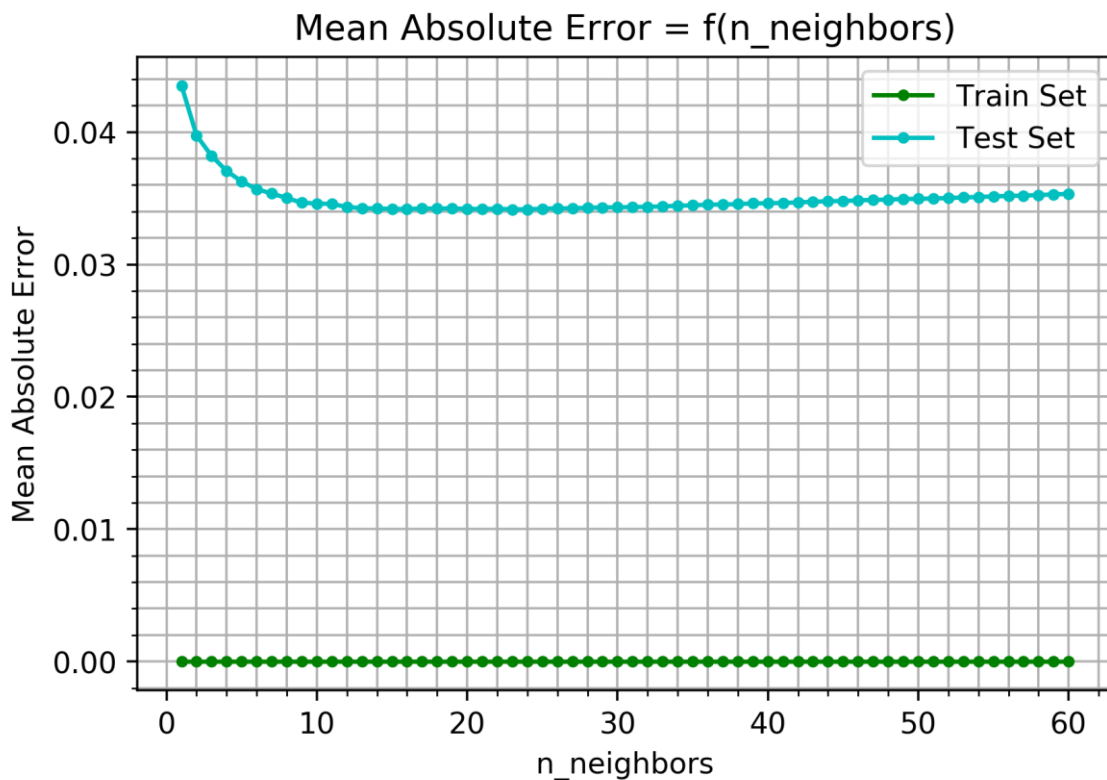
Σχήμα 5.5 – α) Καμπύλες εκπαίδευσης του μοντέλου *KNeighborsRegressor* ως προς την τιμή της υπερ-παραμέτρου *leaf\_size*. β) Χρόνος εκπαίδευσης του μοντέλου *KNeighborsRegressor* ως προς την τιμή της υπερ-παραμέτρου *leaf\_size*.

***n neighbors***

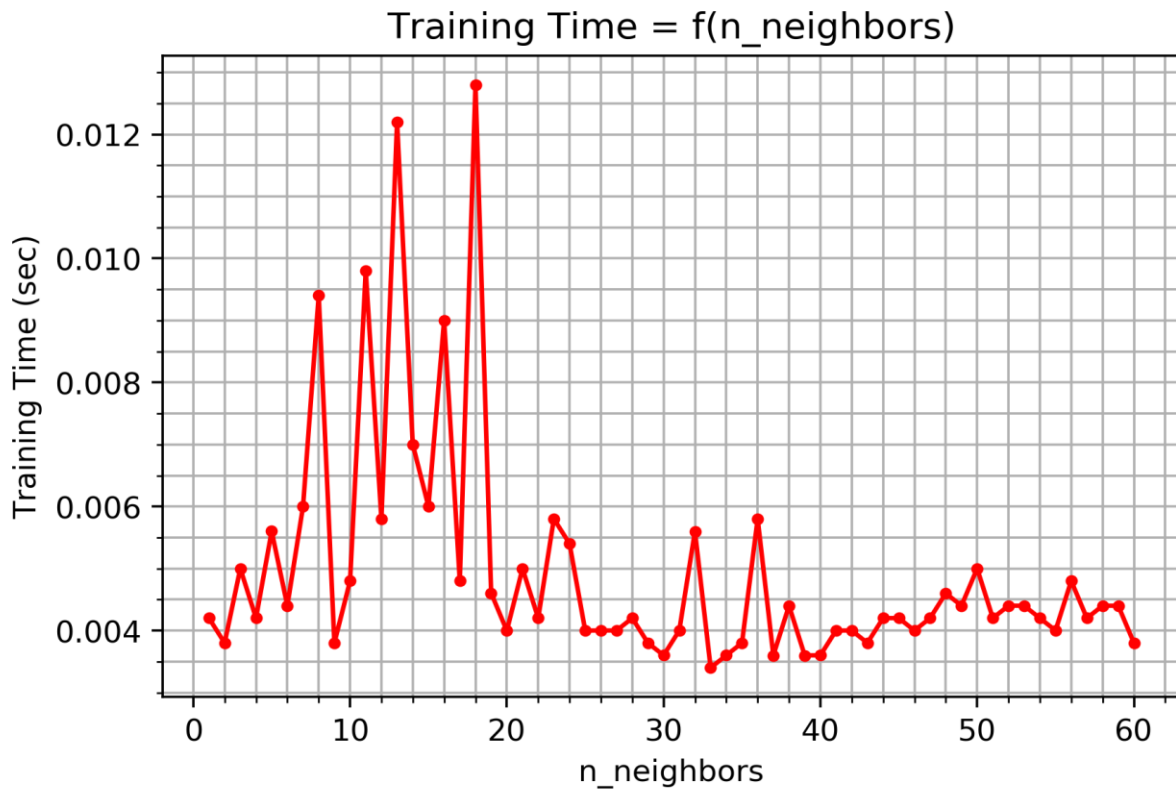
Στο Σχήμα 5.6-α παρουσιάζονται οι καμπύλες εκπαίδευσης που αφορούν τα σύνολα εκπαίδευσης και δοκιμής ως προς την τιμή της υπερ-παραμέτρου  $n\_neighbors$ . Από το σχήμα αυτό παρατηρούμε ότι η μεταβολή του  $n\_neighbors$  έχει επίδραση μόνο στο μέσο απόλυτο σφάλμα του συνόλου δοκιμής, και συγκεκριμένα: με την αύξηση της τιμής του  $n\_neighbors$  το σφάλμα αρχικά μειώνεται μέχρι την τιμή  $n\_neighbors = 10$ , πέρα από την οποία παραμένει σταθερό στην ελάχιστη τιμή του. Αν όμως η τιμή του  $n\_neighbors$  αυξηθεί πάνω από την τιμή 32 (περίπου), τότε το σφάλμα πρόβλεψης του συνόλου δοκιμής αρχίζει να αυξάνεται, κάτι που είναι ενδεικτικό της εμφάνισης Overfitting στο μοντέλο. Στο σχήμα μπορούμε επίσης να παρατηρήσουμε την ύπαρξη του χάσματος γενίκευσης (generalization gap) μεταξύ των δύο καμπυλών, σε όλο το εύρος τιμών της παραμέτρου.

Στο Σχήμα 5.6-β παρουσιάζεται η μεταβολή του χρόνου εκπαίδευσης του μοντέλου ως προς την υπερ-παραμέτρο  $n\_neighbors$ . Και εδώ παρατηρούμε ότι για όλο εύρος των τιμών που εξετάστηκαν ο χρόνος εκπαίδευσης παρουσιάζει κάποιες μικρές διακυμάνσεις (της τάξεως των msec), αλλά πρακτικά παραμένει σταθερός σε μια χαμηλή τιμή. Οι διακυμάνσεις είναι σχετικά πιο έντονες στο εύρος τιμών του  $n\_neighbors$  από 6 έως 20.

Από τα δύο αυτά σχήματα μπορούμε να συμπεράνουμε ότι η τιμή  **$n\_neighbors = 10$**  που επιλέχθηκε είναι αποδεκτή ως βέλτιστη, καθώς οδηγεί στο ελάχιστο μέσο απόλυτο σφάλμα και σε γρήγορη εκπαίδευση. Πρακτικά θα μπορούσαμε να επιλέξουμε και λίγο μεγαλύτερες τιμές του, όχι όμως πάνω από 30, καθώς τότε θα είχαμε την εμφάνιση Overfitting.



(α)



(β)

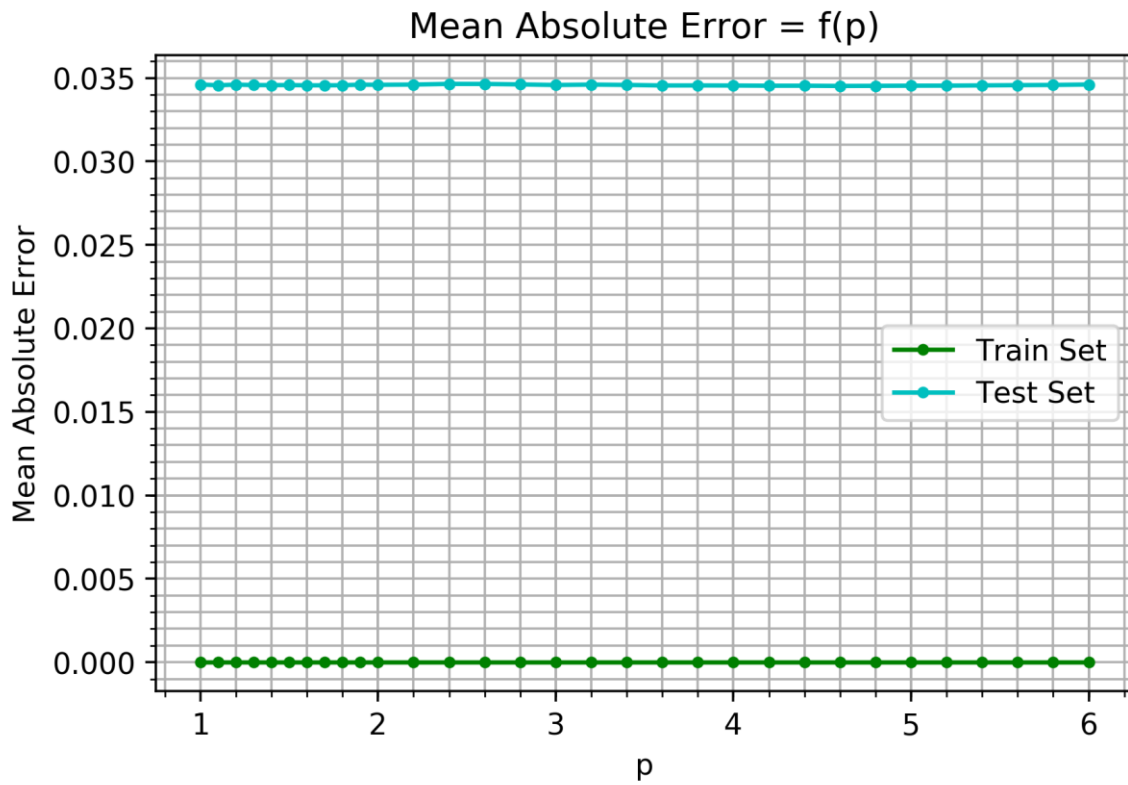
**Σχήμα 5.6** – α) Καμπύλες εκπαίδευσης του μοντέλου *KNeighborsRegressor* ως προς την τιμή της υπερ-παραμέτρου *n\_neighbors*. β) Χρόνος εκπαίδευσης του μοντέλου *KNeighborsRegressor* ως προς την τιμή της υπερ-παραμέτρου *n\_neighbors*.

### ***p***

Στο *Σχήμα 5.7-α* παρουσιάζονται οι καμπύλες εκπαίδευσης που αφορούν τα σύνολα εκπαίδευσης και δοκιμής ως προς την τιμή της υπερ-παραμέτρου *p*. Η συμπεριφορά εδώ είναι αντίστοιχη με την υπερ-παραμέτρο *leaf\_size*: η μεταβολή του *p* δεν έχει ουσιαστική επίδραση στο μέσο απόλυτο σφάλμα, ενώ είναι και πάλι φανερή η ύπαρξη του χάσματος γενίκευσης μεταξύ των δύο καμπυλών. Το χάσμα αυτό παραμένει σταθερό σε όλο το εύρος των τιμών που εξετάστηκαν, επομένως δεν έχουμε εμφάνιση του φαινομένου *Overfitting*.

Στο *Σχήμα 5.7-β* παρουσιάζεται η μεταβολή του χρόνου εκπαίδευσης του μοντέλου ως προς την υπερ-παραμέτρο *p*. Και πάλι παρατηρούμε ότι για όλο εύρος των τιμών που εξετάστηκαν ο χρόνος εκπαίδευσης παρουσιάζει μικρές διακυμάνσεις (της τάξεως των msec), αλλά πρακτικά παραμένει σταθερός σε μια χαμηλή τιμή.

Από τα δύο αυτά σχήματα μπορούμε να συμπεράνουμε ότι η τιμή  $p = 1$  που επιλέχθηκε είναι αποδεκτή ως βέλτιστη, καθώς οδηγεί στο ελάχιστο μέσο απόλυτο σφάλμα και σε γρήγορη εκπαίδευση. Πρακτικά, οποιαδήποτε τιμή του στο εύρος από 1 έως 6 φαίνεται ότι θα είχε το ίδιο αποτέλεσμα στην ποιότητα του μοντέλου, τουλάχιστον ως προς την ελαχιστοποίηση του μέσου απόλυτου σφάλματος.



(α)



(β)

Σχήμα 5.7 – α) Καμπύλες εκπαίδευσης του μοντέλου *KNeighborsRegressor* ως προς την τιμή της υπερ-παραμέτρου  $p$ . β) Χρόνος εκπαίδευσης του μοντέλου *KNeighborsRegressor* ως προς την τιμή της υπερ-παραμέτρου  $p$ .



### 5.2.3 – Random Forest Regressor

Το μοντέλο παλινδρόμησης Random Forest εξετάστηκε στην *Ενότητα 4.7.1*, από όπου προέκυψε ότι οι βέλτιστες τιμές των υπερ-παραμέτρων του είναι:

- ***n\_estimators*** = 100
- ***criterion*** = 'mae'
- ***max\_depth*** = 10
- ***max\_features*** = 2
- ***min\_samples\_split*** = 0,002
- ***min\_samples\_leaf*** = 0,0005

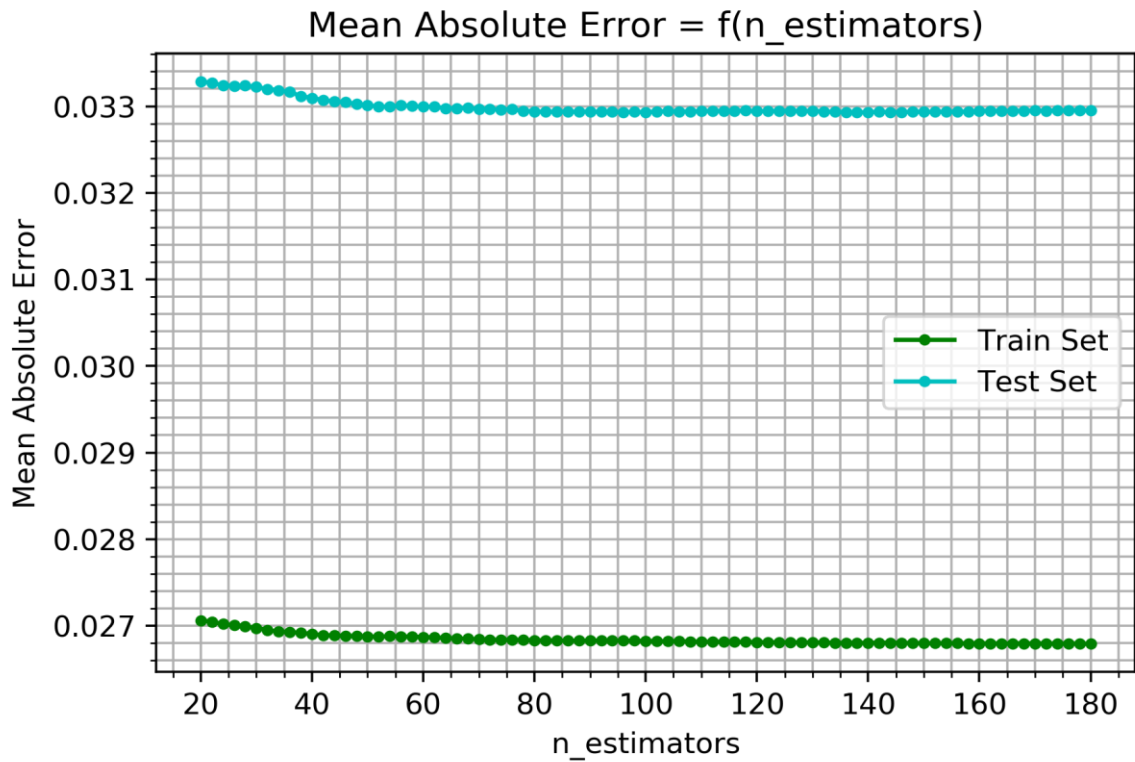
Η αναζήτηση των βέλτιστων υπερ-παραμέτρων έγινε με χρήση του αλγορίθμου GridSearchCV, με 5-fold CV και με κριτήριο τη μεγιστοποίηση του συντελεστή προσδιορισμού  $R^2$ . Στη συνέχεια, με χρήση των καμπυλών εκπαίδευσης, θα εξεταστεί η επίδραση που έχει η κάθε υπερ-παραμέτρος στο μέσο απόλυτο σφάλμα πρόβλεψης (mean absolute error) και στο χρόνο εκπαίδευσης του μοντέλου, και θα διαπιστωθεί αν οι παραπάνω τιμές είναι οι βέλτιστες και ως προς αυτά τα κριτήρια.

#### ***n\_estimators***

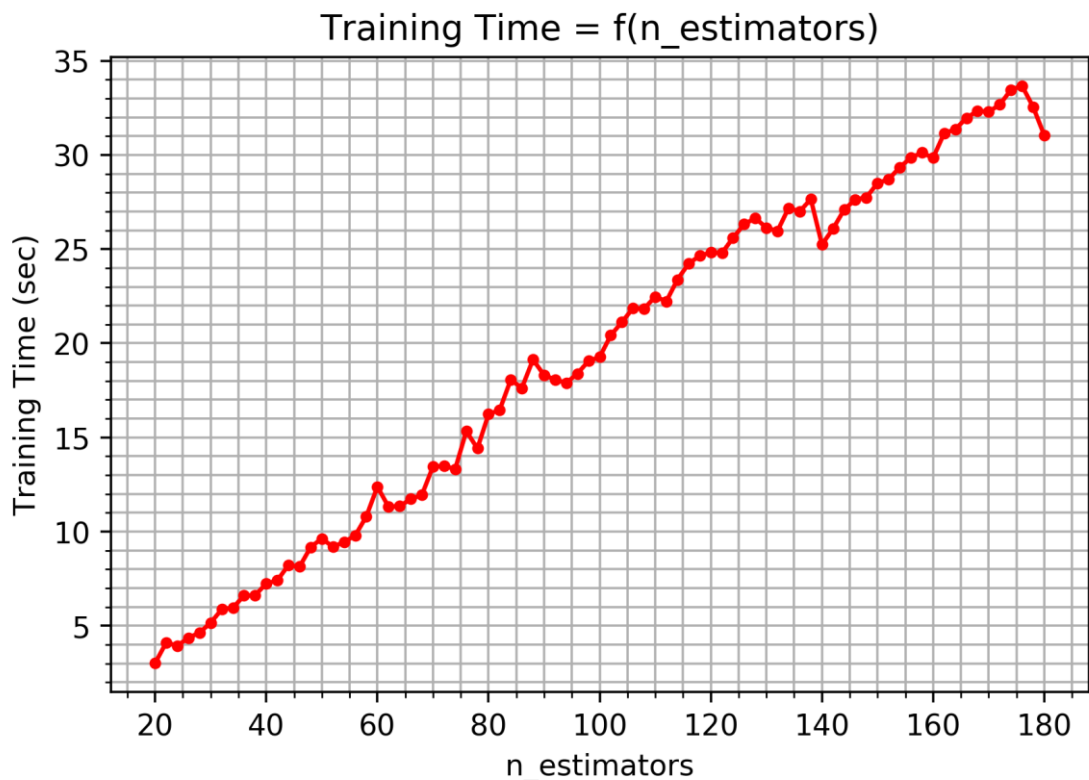
Στο *Σχήμα 5.8-α* παρουσιάζονται οι καμπύλες εκπαίδευσης που αφορούν τα σύνολα εκπαίδευσης και δοκιμής ως προς την τιμή της υπερ-παραμέτρου *n\_estimators*. Από το σχήμα αυτό παρατηρούμε ότι η αύξηση του *n\_estimators* προκαλεί μείωση του σφάλματος, μέχρι ένα σημείο ισορροπίας, περίπου για *n\_estimators* = 80, πέρα από το οποίο η περεταίρω αύξησή του δεν έχει σημαντική επίδραση στο σφάλμα. Είναι εμφανές και εδώ το χάσμα γενίκευσης (generalization gap) μεταξύ των δύο καμπυλών, που οφείλεται στη γενική ιδιότητα των μοντέλων μηχανικής μάθησης να έχουν χαμηλότερη απόδοση όταν γενικεύουν (εκτελούν προβλέψεις) σε νέα δεδομένα. Βλέπουμε όμως ότι για όλο το εύρος των τιμών του *n\_estimators* που εξετάστηκαν, το χάσμα γενίκευσης παραμένει σχεδόν σταθερό, επομένως δεν έχουμε εμφάνιση Overfitting.

Στο *Σχήμα 5.8-β* παρουσιάζεται η μεταβολή του χρόνου εκπαίδευσης του μοντέλου ως προς την υπερ-παραμέτρο *n\_estimators*. Παρατηρούμε ότι ο χρόνος εκπαίδευσης του μοντέλου αυξάνεται σταθερά με την αύξηση του *n\_estimators*, κάτι που είναι αναμενόμενο καθώς υψηλότερες τιμές του *n\_estimators* σημαίνουν ότι πρέπει να κατασκευαστούν περισσότερα επιμέρους δέντρα και άρα αυξημένη πολυπλοκότητα του μοντέλου Random Forest.

Από τα δύο αυτά σχήματα μπορούμε να συμπεράνουμε ότι η τιμή ***n\_estimators* = 100** που επιλέχθηκε είναι αποδεκτή ως βέλτιστη, καθώς οδηγεί στην ελαχιστοποίηση του μέσου απόλυτου σφάλματος με ικανοποιητικό χρόνο εκπαίδευσης. Στην περίπτωση που ο χρόνος εκπαίδευσης θα ήταν σημαντικό κριτήριο, θα μπορούσαμε ενδεχομένως να επιλέξουμε και λίγο μικρότερη τιμή, για παράδειγμα 80, η οποία δίνει σχεδόν ίδια τιμή του μέσου απόλυτου σφάλματος αλλά ελαφρώς μικρότερο χρόνο εκπαίδευσης.



(α)



(β)

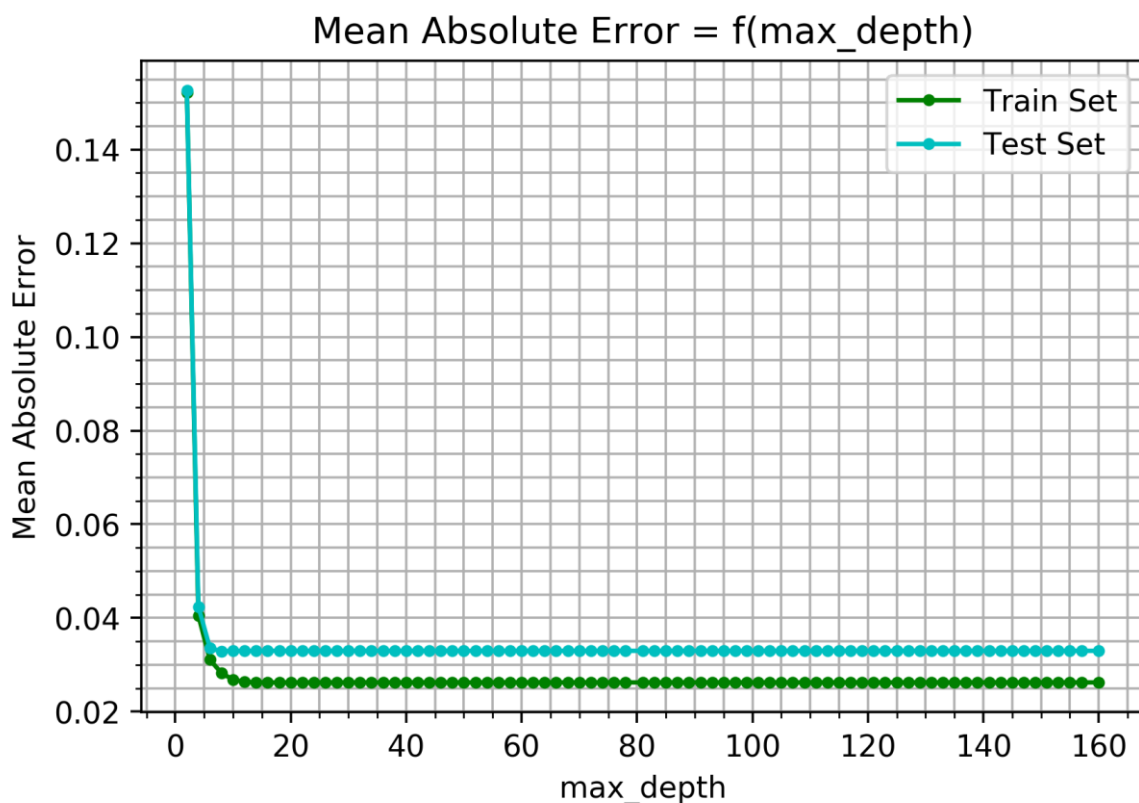
**Σχήμα 5.8** – α) Καμπύλες εκπαίδευσης του μοντέλου *RandomForestRegressor* ως προς την τιμή της υπερ-παραμέτρου *n\_estimators*. β) Χρόνος εκπαίδευσης του μοντέλου *RandomForestRegressor* ως προς την τιμή της υπερ-παραμέτρου *n\_estimators*.

**max\_depth**

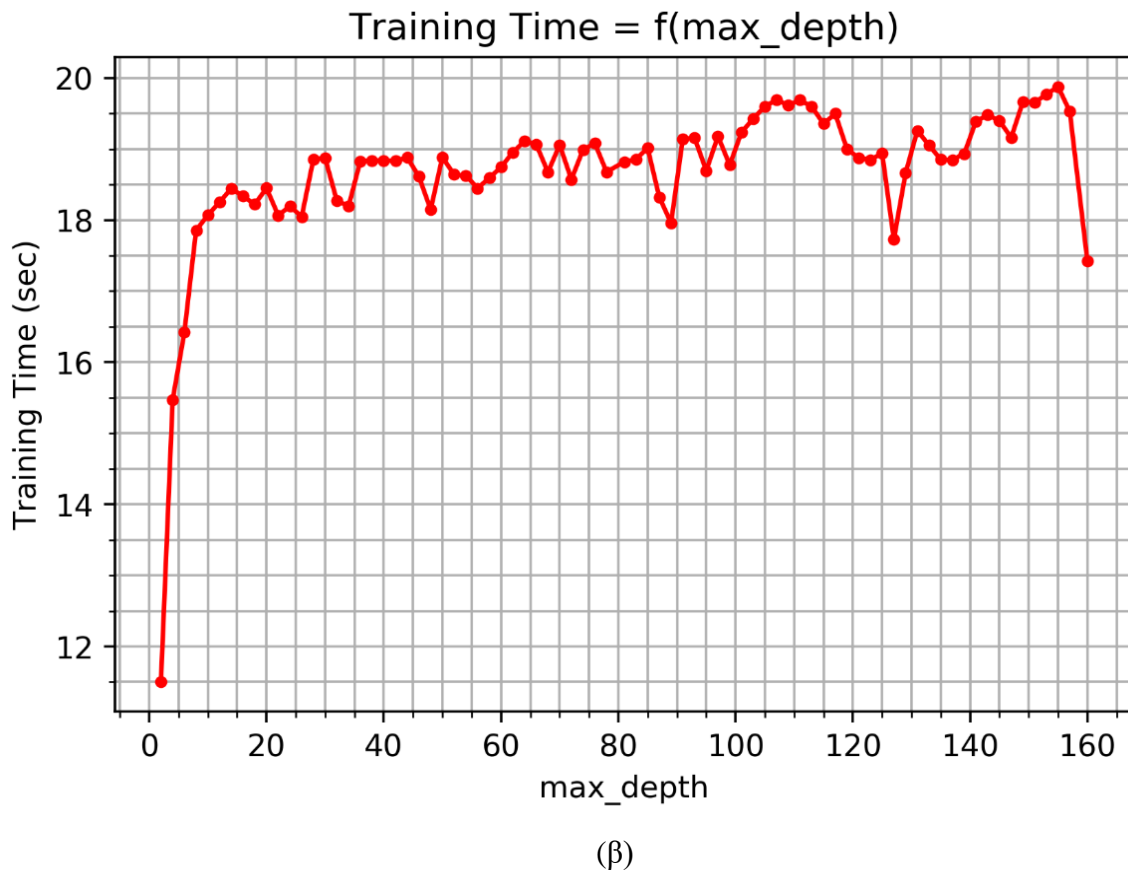
Στο Σχήμα 5.9-α παρουσιάζονται οι καμπύλες εκπαίδευσης που αφορούν τα σύνολα εκπαίδευσης και δοκιμής ως προς την τιμή της υπερ-παραμέτρου `max_depth`. Από το σχήμα αυτό παρατηρούμε ότι η αύξηση του `max_depth` προκαλεί πολύ γρήγορη ελαχιστοποίηση του σφάλματος, μέχρι την τιμή `max_depth = 10`, πέρα από την οποία η περαιτέρω αύξηση του βάθους των επιμέρους δέντρων δεν έχει επίδραση στο σφάλμα. Βλέπουμε επίσης ότι για όλο το εύρος των τιμών του `max_depth` που εξετάστηκαν, οι καμπύλες για τα σύνολα εκπαίδευσης και δοκιμής διατηρούν μια σταθερή διαφορά (που είναι το χάσμα γενίκευσης), επομένως δεν έχουμε εμφάνιση Overfitting στο μοντέλο.

Στο Σχήμα 5.9-β παρουσιάζεται η μεταβολή του χρόνου εκπαίδευσης του μοντέλου ως προς την υπερ-παραμέτρο `max_depth`. Παρατηρούμε ότι με την αύξηση του `max_depth` ο χρόνος εκπαίδευσης αρχικά αυξάνεται πολύ γρήγορα, μέχρι την τιμή `max_depth = 15`, πέρα από την οποία παραμένει σχετικά σταθερός με μικρές διακυμάνσεις (οι οποίες γίνονται πιο έντονες στις υψηλές τιμές του `max_depth`).

Από τα δύο αυτά σχήματα μπορούμε να συμπεράνουμε ότι η τιμή **`max_depth = 10`** που επιλέχθηκε είναι αποδεκτή ως βέλτιστη, καθώς οδηγεί στο ελάχιστο μέσο απόλυτο σφάλμα με αρκετά χαμηλό χρόνο εκπαίδευσης. Βλέπουμε στο Σχήμα 5.9-β ότι υπάρχουν κάποιες τιμές του `max_depth` όπως το 160, οι οποίες δίνουν την ίδια τιμή σφάλματος αλλά με μικρότερο χρόνο εκπαίδευσης.



(α)



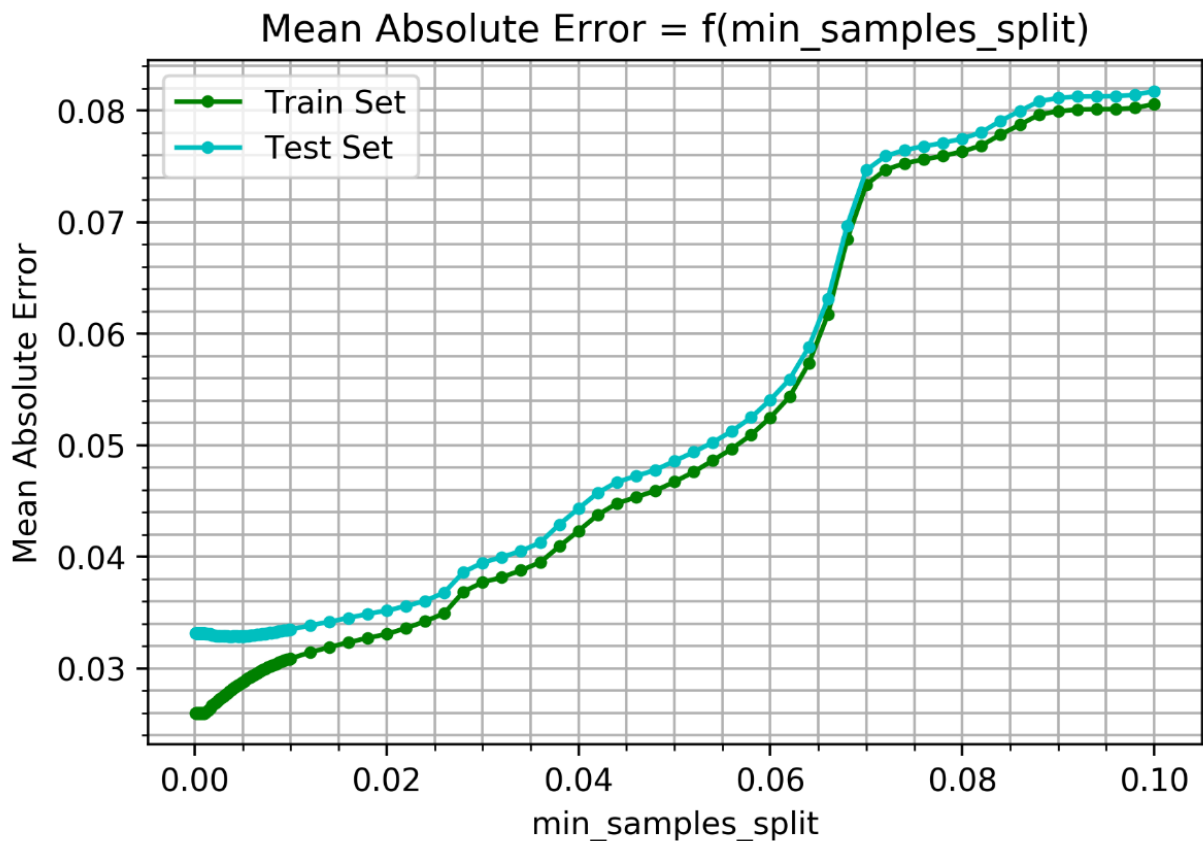
**Σχήμα 5.9** – α) Καμπύλες εκπαίδευσης του μοντέλου *RandomForestRegressor* ως προς την τιμή της υπερ-παραμέτρου *max\_depth*. β) Χρόνος εκπαίδευσης του μοντέλου *RandomForestRegressor* ως προς την τιμή της υπερ-παραμέτρου *max\_depth*.

### *min samples split*

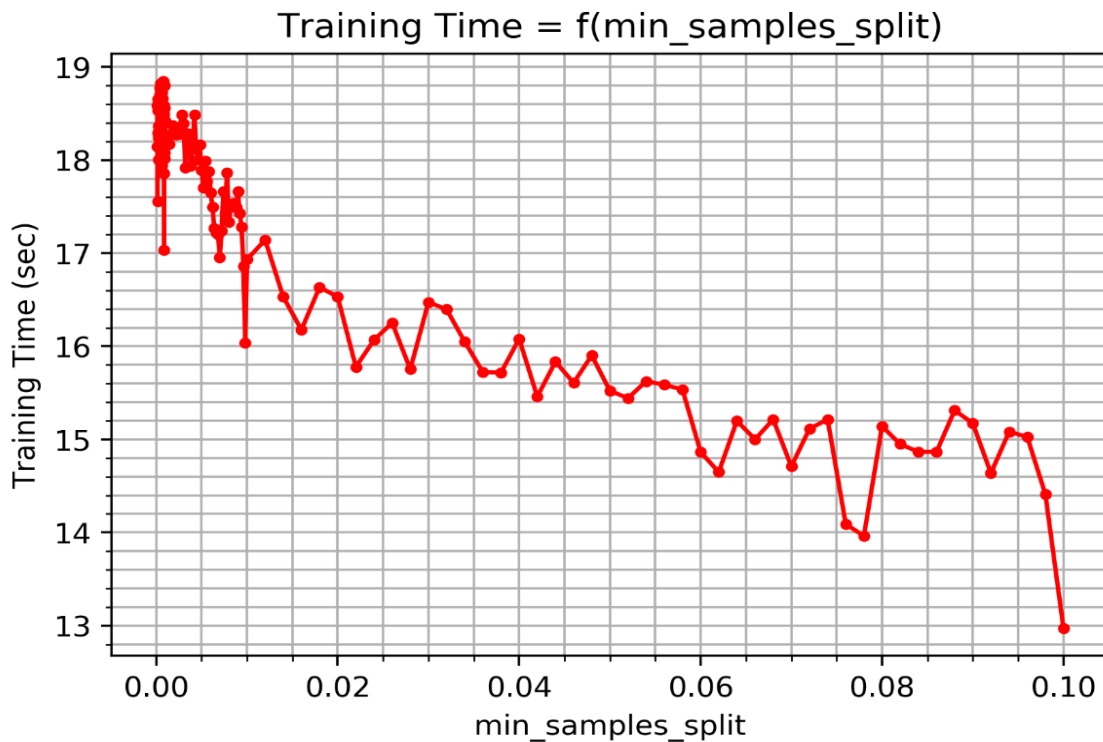
Στο Σχήμα 5.10-α παρουσιάζονται οι καμπύλες εκπαίδευσης που αφορούν τα σύνολα εκπαίδευσης και δοκιμής ως προς την τιμή της υπερ-παραμέτρου *min\_samples\_split*. Από το σχήμα αυτό παρατηρούμε ότι η αύξηση του *min\_samples\_split* προκαλεί γενικά αύξηση του σφάλματος, και μάλιστα σε μεγάλες τιμές του *min\_samples\_split* (πάνω από 0,07) το σφάλμα παίρνει υψηλές τιμές. Αυτό είναι ένα δείγμα φαινομένου *Underfitting*, το οποίο πιθανώς να γίνεται πιο έντονο αν αυξήσουμε το *min\_samples\_split* σε τιμές πάνω από το εύρος του σχήματος. Γενικά, οι υψηλές τιμές αυτής της παραμέτρου οδηγούν στην κατασκευή πολύ απλών επιμέρους δέντρων, τα οποία δεν μπορούν να προσαρμοστούν καλά στα δεδομένα εκπαίδευσης. Τέλος είναι φανερό στο σχήμα και το χάσμα γενίκευσης μεταξύ των δύο καμπυλών, το οποίο μάλιστα αυξάνεται για πολύ χαμηλές τιμές του *min\_samples\_split* (κάτω από 0,002), κάτι το οποίο μπορεί να υποδηλώνει την εμφάνιση *Overfitting*, καθώς σε τόσο χαμηλές τιμές της παραμέτρου το μοντέλο μαθαίνει χαρακτηριστικά που είναι πολύ συγκεκριμένα για το συγκεκριμένο σύνολο εκπαίδευσης.

Στο Σχήμα 5.10-β παρουσιάζεται η μεταβολή του χρόνου εκπαίδευσης του μοντέλου ως προς την υπερ-παράμετρο `min_samples_split`. Παρατηρούμε ότι γενικά υπάρχει η τάση να μειώνεται ο χρόνος εκπαίδευσης με την αύξηση του `min_samples_split`, κάτι που είναι αναμενόμενο καθώς υψηλές τιμές του `min_samples_split` οδηγούν στη δημιουργία απλούστερων δέντρων, τα οποία χρειάζονται λιγότερο χρόνο για να κατασκευαστούν.

Από τα δύο αυτά σχήματα μπορούμε να συμπεράνουμε ότι η τιμή `min_samples_split = 0,002` που επιλέχθηκε είναι αποδεκτή ως βέλτιστη, καθώς οδηγεί στο ελάχιστο μέσο απόλυτο σφάλμα αλλά με σχετικά υψηλό χρόνο εκπαίδευσης. Όπως βλέπουμε από τα Σχήματα 5.10, η μείωση του χρόνου εκπαίδευσης απαιτεί αύξηση της τιμής του `min_samples_split`, η οποία όμως συνεπάγεται και αύξηση του μέσου απόλυτου σφάλματος. Επομένως, αν ο χρόνος ήταν σημαντικός παράγοντας, τότε θα έπρεπε αυξήσουμε την τιμή της παραμέτρου αυτής, θυσιάζοντας έτσι ένα ποσοστό της ακρίβειας του μοντέλου.



(α)



(β)

**Σχήμα 5.10** – α) Καμπύλες εκπαίδευσης του μοντέλου *RandomForestRegressor* ως προς την τιμή της υπερ-παραμέτρου *min\_samples\_split*. β) Χρόνος εκπαίδευσης του μοντέλου *RandomForestRegressor* ως προς την τιμή της υπερ-παραμέτρου *min\_samples\_split*.

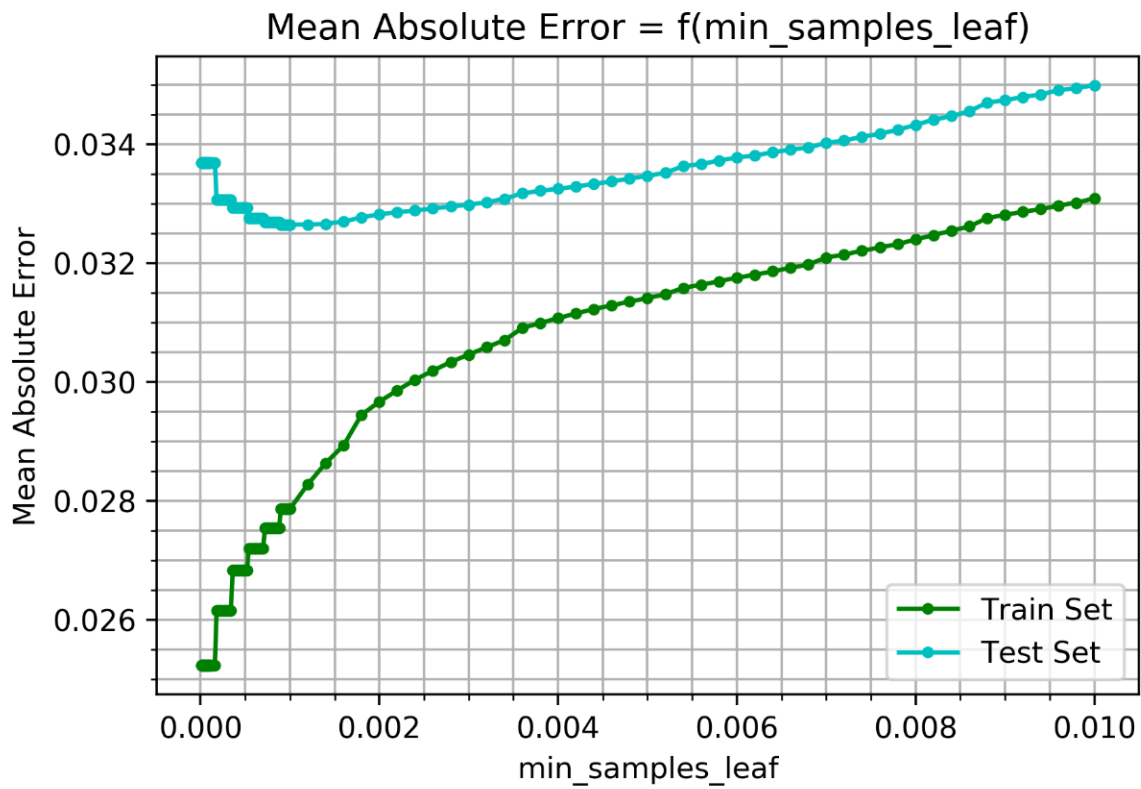
### **min samples leaf**

Στο Σχήμα 5.11-α παρουσιάζονται οι καμπύλες εκπαίδευσης που αφορούν τα σύνολα εκπαίδευσης και δοκιμής ως προς την τιμή της υπερ-παραμέτρου *min\_samples\_leaf*. Η συμπεριφορά εδώ είναι αντίστοιχη με την περίπτωση της παραμέτρου *min\_samples\_split*: η αύξηση του *min\_samples\_leaf* προκαλεί γενικά αύξηση του σφάλματος, και αν αυξήσουμε το *min\_samples\_leaf* σε τιμές πάνω από το εύρος του σχήματος πιθανώς να έχουμε Underfitting. Παρατηρούμε και εδώ το χάσμα γενίκευσης μεταξύ των δύο καμπυλών, το οποίο μάλιστα αυξάνεται για πολύ χαμηλές τιμές του *min\_samples\_leaf* (κάτω από 0,005) και έχουμε φαινόμενο Overfitting, καθώς σε τόσο χαμηλές τιμές της παραμέτρου το μοντέλο μαθαίνει χαρακτηριστικά που είναι πολύ συγκεκριμένα για το συγκεκριμένο σύνολο εκπαίδευσης.

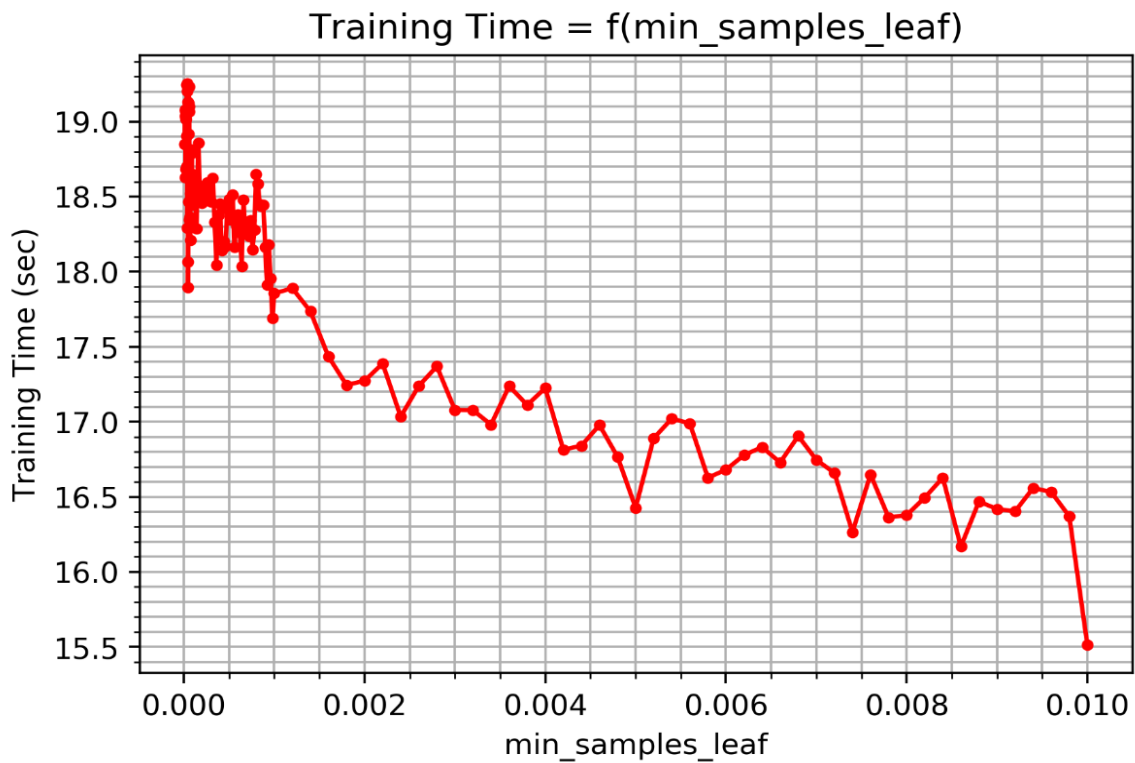
Στο Σχήμα 5.11-β παρουσιάζεται η μεταβολή του χρόνου εκπαίδευσης ως προς το *min\_samples\_leaf*, όπου βλέπουμε ότι υπάρχει η τάση να μειώνεται ο χρόνος εκπαίδευσης με την αύξηση του *min\_samples\_leaf*, καθώς υψηλές τιμές του οδηγούν στη δημιουργία απλούστερων δέντρων, τα οποία χρειάζονται λιγότερο χρόνο για να κατασκευαστούν.

Από τα δύο αυτά σχήματα μπορούμε να συμπεράνουμε ότι η τιμή ***min\_samples\_leaf* = 0,0005** που επιλέχθηκε είναι οριακά αποδεκτή ως βέλτιστη, καθώς οδηγεί στο ελάχιστο μέσο απόλυτο σφάλμα, ενώ περαιτέρω μείωσή της θα οδηγούσε στην εμφάνιση Overfitting. Για τον χρόνο εκπαίδευσης ισχύει ότι και στην περίπτωση του

`min_samples_split`, καθώς η προσπάθεια μείωσής του θα οδηγήσει στην αύξηση του σφάλματος πρόβλεψης.



(α)



(β)

**Σχήμα 5.11** – α) Καμπύλες εκπαίδευσης του μοντέλου `RandomForestRegressor` ως προς την τιμή της υπερ-παραμέτρου `min_samples_leaf`. β) Χρόνος εκπαίδευσης του μοντέλου `RandomForestRegressor` ως προς την τιμή της υπερ-παραμέτρου `min_samples_leaf`.

### 5.3 – Επιλογή Τελικού Βέλτιστου Μοντέλου

Από την ανάλυση που προηγήθηκε στις προηγούμενες ενότητες προκύπτει ότι τα μοντέλα *SVR*, *KNeighbors* και *Random Forest*, με τις βέλτιστες τιμές των υπερ-παραμέτρων τους, παρουσιάζουν όλα πολύ καλή συμπεριφορά, τόσο ως προς την ελαχιστοποίηση του μέσου απόλυτου σφάλματος πρόβλεψης όσο και ως προς την αντιμετώπιση φαινομένων *Overfitting* και *Underfitting*.

Στον Πίνακα 5.1 δίνονται συγκεντρωτικά τα αποτελέσματα αυτού του κεφαλαίου για τα βέλτιστα μοντέλα *SVR*, *KNeighbors* και *Random Forest* (δηλαδή για τις βέλτιστες τιμές των υπερ-παραμέτρων τους). Από τον πίνακα αυτόν βλέπουμε ότι το μοντέλο *KNeighbors* έχει μακράν τον καλύτερο χρόνο εκπαίδευσης, όμως το μικρότερο μέσο απόλυτο σφάλμα, που είναι και το βασικό κριτήριο που μας ενδιαφέρει για την πρόβλεψη, το έχει το μοντέλο *SVR*.

Το γεγονός αυτό, σε συνδυασμό με τα γενικότερα πλεονεκτήματα των Μηχανών Διανυσμάτων Υποστήριξης, μας οδηγεί να επιλέξουμε ως βέλτιστο μοντέλο για το πρόβλημά μας το μοντέλο **Παλινδρόμησης Διανυσμάτων Υποστήριξης (Support Vector Regressor)**, με τιμές των υπερ-παραμέτρων του:

- *kernel* = 'rbf'
- *gamma* = 50
- *C* = 100
- *epsilon* = 0,01

Το μοντέλο αυτό θα χρησιμοποιηθεί στο επόμενο κεφάλαιο για την πρόβλεψη της ισχύος των 5 αιολικών πάρκων της Ρόδου, και τον υπολογισμό στη συνέχεια της απορριπτόμενης ισχύος κάθε πάρκου.

*Πίνακας 5.1 – Προσεγγιστικές τιμές για το μέσο απόλυτο σφάλμα του συνόλου δοκιμής και τον χρόνο εκπαίδευσης των βέλτιστων μοντέλων SVR, KNeighbors και Random Forest, όπως προέκυψαν από την ανάλυση αυτού του κεφαλαίου.*

	<b>Mean Absolute Error (test set)</b>	<b>Training Time (sec)</b>
<b>SVR</b>	0,0321	20
<b>KNeighbors Regressor</b>	0,0345	0,0045
<b>Random Forest Regressor</b>	0,0329	19



### 6.1 – Εισαγωγή

Στο κεφάλαιο αυτό πραγματοποιείται η εκτίμηση (πρόβλεψη) της ετήσιας δυνατότητας παραγωγής ισχύος του κάθε πάρκου, καθώς και της ετήσιας απορριπτόμενης ισχύος από αυτό. Για όλες τις προβλέψεις χρησιμοποιείται το βέλτιστο μοντέλο SVR (*Support Vector Regression*) το οποίο επιλέχθηκε στο προηγούμενο κεφάλαιο (βλ. *Ενότητα 5.3*). Τα ετήσια μεγέθη παραγόμενης και απορριπτόμενης ισχύος υπολογίζονται για τα έτη 2017 και 2018, για τα οποία διαθέτουμε ανεμολογικά δεδομένα και μετρήσεις ισχύος.

Για κάθε πάρκο, η διαδικασία που ακολουθείται έχει ως εξής: αρχικά εκπαιδεύεται το βέλτιστο μοντέλο SVR. Για την εκπαίδευσή του χρησιμοποιούνται δείγματα και από τα δύο έτη (2017 και 2018) για τα οποία το Set Point είναι 1 (βλ. *Ενότητα 3.1.1*) και η μετρημένη ισχύς έχει θετική τιμή. Αρνητικές τιμές ισχύος στη χρονοσειρά σημαίνουν ότι για τη συγκεκριμένη χρονική στιγμή δεν πραγματοποιήθηκε μέτρηση, επομένως αυτά τα δείγματα μπορούν να αγνοηθούν κατά τη δημιουργία του συνόλου εκπαίδευσης (training set). Το μοντέλο SVR εκπαιδεύεται με αυτά τα δεδομένα, και στη συνέχεια χρησιμοποιείται για να εκτιμήσει (προβλέψει) την παραγόμενη ισχύ του πάρκου για τις χρονικές στιγμές όπου το Set Point είναι μικρότερο του 1. Στη συνέχεια, για κάθε μία από αυτές τις χρονικές στιγμές υπολογίζεται η απορριπτόμενη ισχύς ως η διαφορά της μετρημένης στον υποσταθμό ισχύος (αυτή που χρησιμοποιείται από το δίκτυο) από την τιμή της παραγόμενης από το πάρκο ισχύος, η οποία προέκυψε από την πρόβλεψη για τη συγκεκριμένη χρονική στιγμή. Για τις περιπτώσεις που η διαφορά αυτή προκύπτει αρνητική, σημαίνει ότι η ισχύς που προβλέφθηκε από το μοντέλο είναι μικρότερη από την ισχύ που πραγματικά χρησιμοποιήθηκε από το δίκτυο εκείνη τη στιγμή, επομένως θεωρούμε ότι η απορριπτόμενη ισχύς είναι μηδενική. Για όλες τις χρονικές στιγμές όπου Set Point = 1 η απορριπτόμενη ισχύς είναι επίσης μηδενική, καθώς όλη η ισχύς που παράγεται από το πάρκο χρησιμοποιείται από το δίκτυο. Σε αυτό το σημείο γνωρίζουμε πλέον την παραγόμενη από το πάρκο ισχύ και την απορριπτόμενη ισχύ για κάθε χρονική στιγμή, επομένως αθροίζοντας αυτά τα μεγέθη για τη διάρκεια ενός έτους προκύπτουν η ετήσια παραγόμενη ισχύς και η ετήσια απορριπτόμενη ισχύς από το πάρκο για το έτος αυτό. Τονίζεται ότι για τις εκτιμήσεις ισχύος δε λαμβάνεται υπόψη το σφάλμα του μοντέλου, το οποίο θεωρείται αμελητέο.

Από τα δεδομένα που προέκυψαν από τις προβλέψεις κατασκευάζονται επίσης, για κάθε πάρκο, οι γραφικές παραστάσεις που δείχνουν το πώς μεταβάλλονται η παραγόμενη και η απορριπτόμενη ισχύς του πάρκου κατά τη διάρκεια του έτους, για τα έτη 2017 και 2018. Οι καμπύλες αυτές χρησιμοποιούνται για την εξαγωγή συμπερασμάτων για το κάθε πάρκο, όσον αφορά τη συμπεριφορά του κατά τη διάρκεια του έτους.

Στο *Παράρτημα Α* δίνεται ενδεικτικά ο κώδικας που χρησιμοποιήθηκε για την παραπάνω διαδικασία. Ο κώδικας αναφέρεται στο 1<sup>ο</sup> πάρκο (Aeiforiki), για το έτος 2018. Η διαδικασία που ακολουθείται για το έτος 2017, καθώς και για τα υπόλοιπα αιολικά πάρκα, είναι αντίστοιχη.

## 6.2 – Αποτελέσματα για τα Πάρκα

### 6.2.1 – Αιολικό Πάρκο 1 (Aeiforiki)

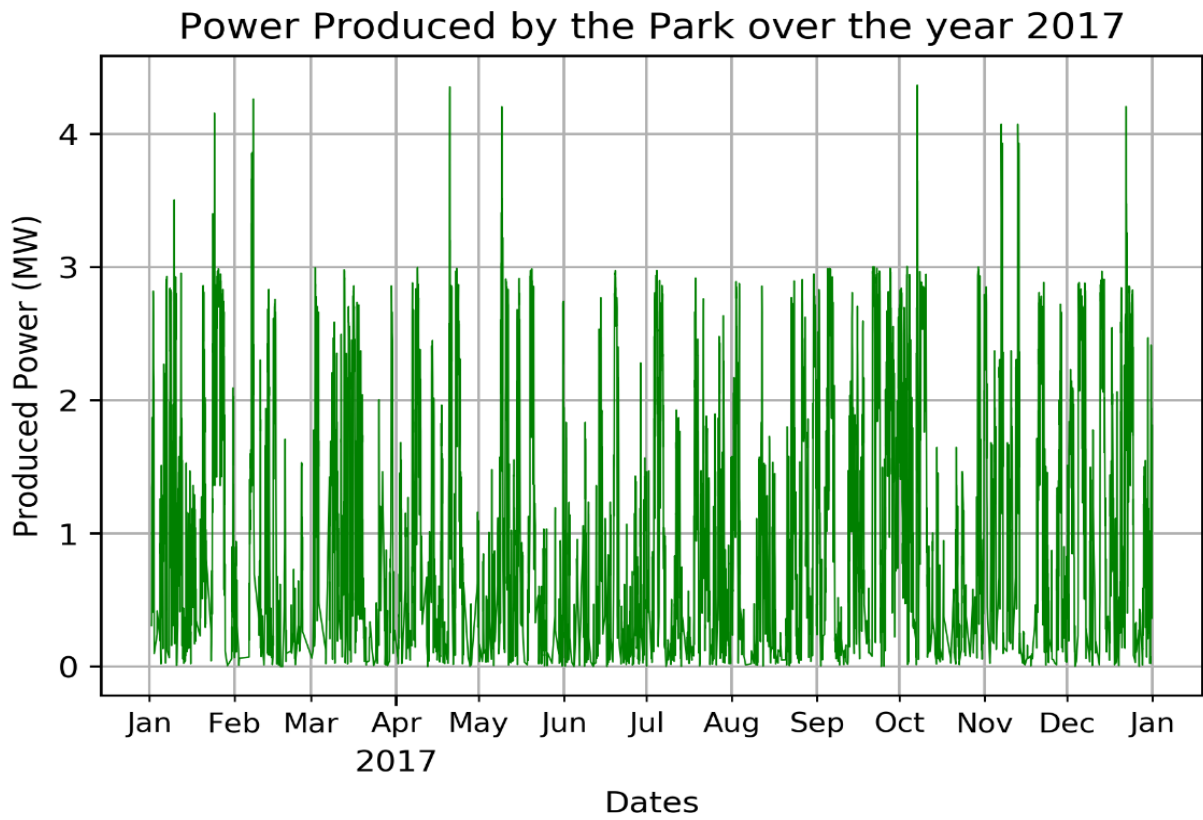
Το αιολικό πάρκο 1 (Aeiforiki) έχει εγκατεστημένη (ονομαστική) ισχύ **4 MW**. Στον Πίνακα 6.1 δίνονται οι τιμές που υπολογίστηκαν για την ετήσια παραγόμενη από το πάρκο ισχύ και την ετήσια απορριπτόμενη ισχύ, για τα έτη 2017 και 2018 καθώς και συνολικά για τα δύο αυτά έτη. Δίνεται επίσης η τιμή της ετήσιας ισχύος που διοχετεύεται στο δίκτυο, η οποία προκύπτει απλά από το άθροισμα όλων των τιμών της μετρημένης στον υποσταθμό ισχύος που αναφέρονται στο συγκεκριμένο έτος. Υπενθυμίζεται ότι οι τιμές της παραγόμενης και της απορριπτόμενης ισχύος στηρίζονται σε προβλέψεις που πραγματοποιήθηκαν με το μοντέλο SVR, επομένως αποτελούν *εκτιμήσεις* του αντίστοιχου μεγέθους και όχι πραγματικές τιμές. Για τις εκτιμήσεις αυτές, όπως προαναφέρθηκε, δε λαμβάνεται υπόψη το σφάλμα του μοντέλου.

Από τον Πίνακα 6.1 προκύπτει ότι και για τα δύο έτη η απορριπτόμενη ισχύς αποτελεί ένα σημαντικό ποσοστό της συνολικής παραγόμενης ισχύος του πάρκου (40% της συνολικής παραγόμενης ισχύος για το 2017 και 41% για το 2018). Παρατηρούμε επίσης ότι το 2018 το πάρκο παρήγαγε περισσότερη ισχύ, γεγονός που πιθανώς οφείλεται σε ευνοϊκότερες συνθήκες ανέμου για το συγκεκριμένο έτος.

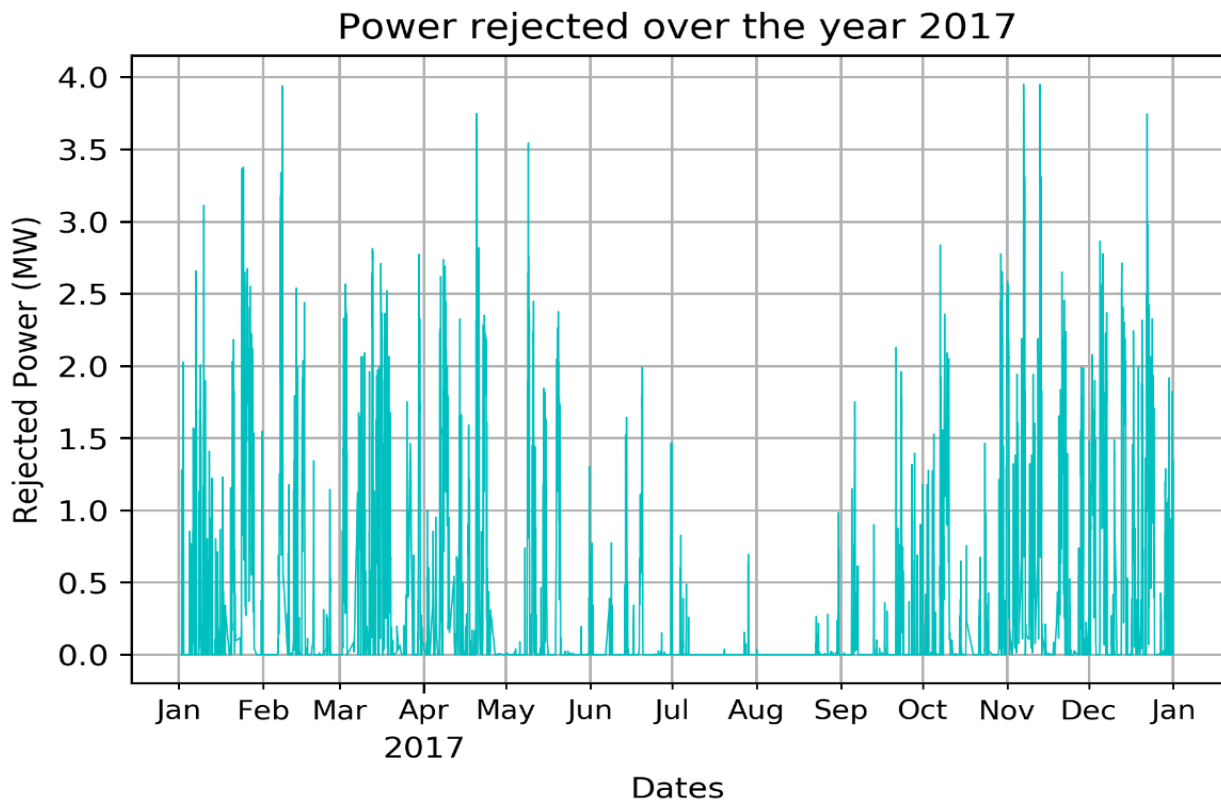
Στα Σχήματα 6.1 και 6.2 απεικονίζεται, για τα έτη 2017 και 2018 αντίστοιχα, η μεταβολή της παραγόμενης και της απορριπτόμενης ισχύος κατά τη διάρκεια του έτους. Αρχικά παρατηρούμε και στα δύο σχήματα τις μεγάλες διακυμάνσεις των μεγεθών αυτών κατά τη διάρκεια του έτους, γεγονός που οφείλεται στον τυχαίο και απρόβλεπτο χαρακτήρα του ανέμου. Σαν γενική συμπεριφορά μπορούμε να παρατηρήσουμε (Σχήματα 6.1-β και 6.2-β) ότι και για τα δύο έτη, η απορριπτόμενη ισχύς ελαχιστοποιείται τους καλοκαιρινούς μήνες (Ιούνιο, Ιούλιο και Αύγουστο). Το γεγονός αυτό οφείλεται στην αύξηση της ζήτησης ηλεκτρικής ενέργειας που παρατηρείται γενικά το καλοκαίρι, κυρίως λόγω των αυξημένων αναγκών κλιματισμού των χώρων.

**Πίνακας 6.1** – Αποτελέσματα προβλέψεων για το 1ο Αιολικό Πάρκο (Aeiforiki): στον πίνακα φαίνονται, για τα έτη 2017 και 2018, οι εκτιμήσεις για την συνολική (ετήσια) παραγόμενη από το πάρκο ισχύ και την απορριπτόμενη ισχύ, όπως προέκυψαν από το βέλτιστο μοντέλο SVR. Δίνεται επίσης η ετήσια ισχύς που διοχετεύεται στο δίκτυο.

Αιολικό Πάρκο 1 (Aeiforiki)			
	Παραγόμενη Ισχύς (MW)	Ισχύς που διοχετεύεται στο δίκτυο (MW)	Απορριπτόμενη Ισχύς (MW)
2017	6145,27	3795,90	2435,50
2018	7240,67	4297,52	2985,07
Συνολικά	13385,94	8093,42	5420,57

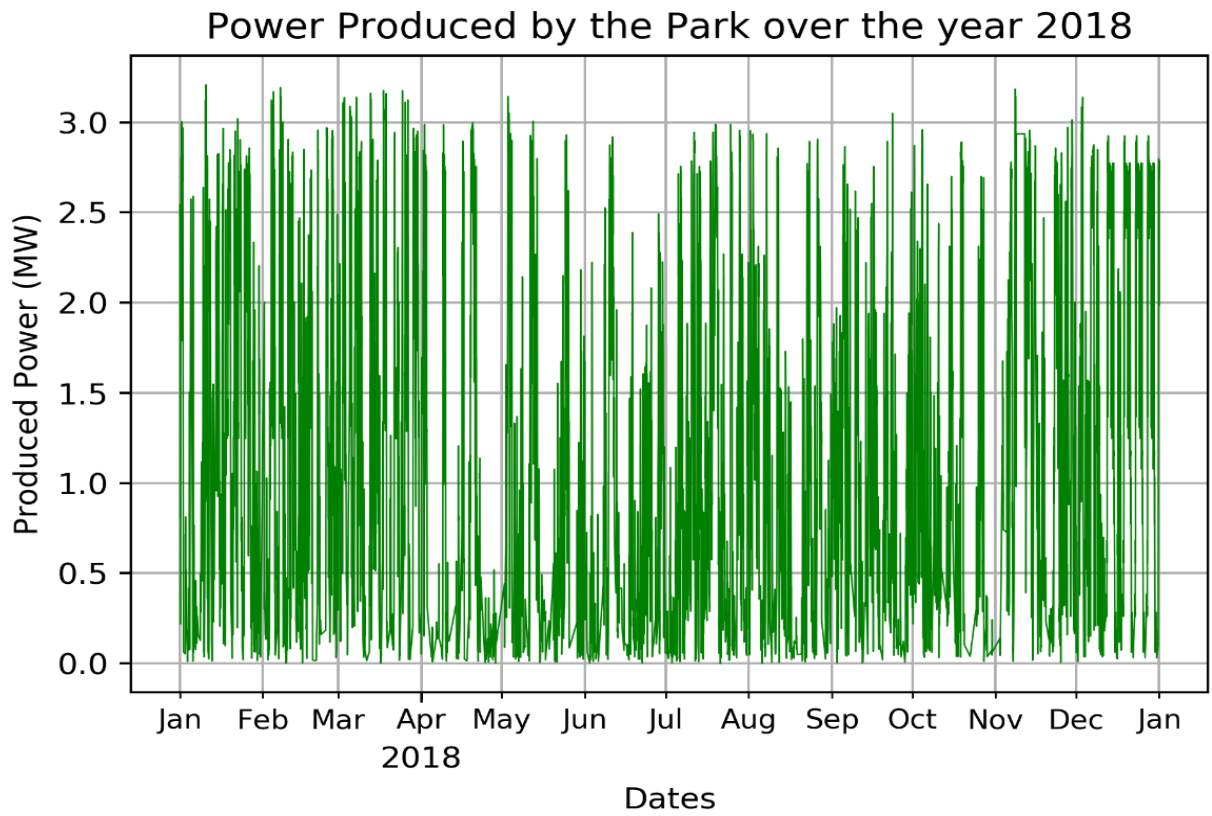


(α)

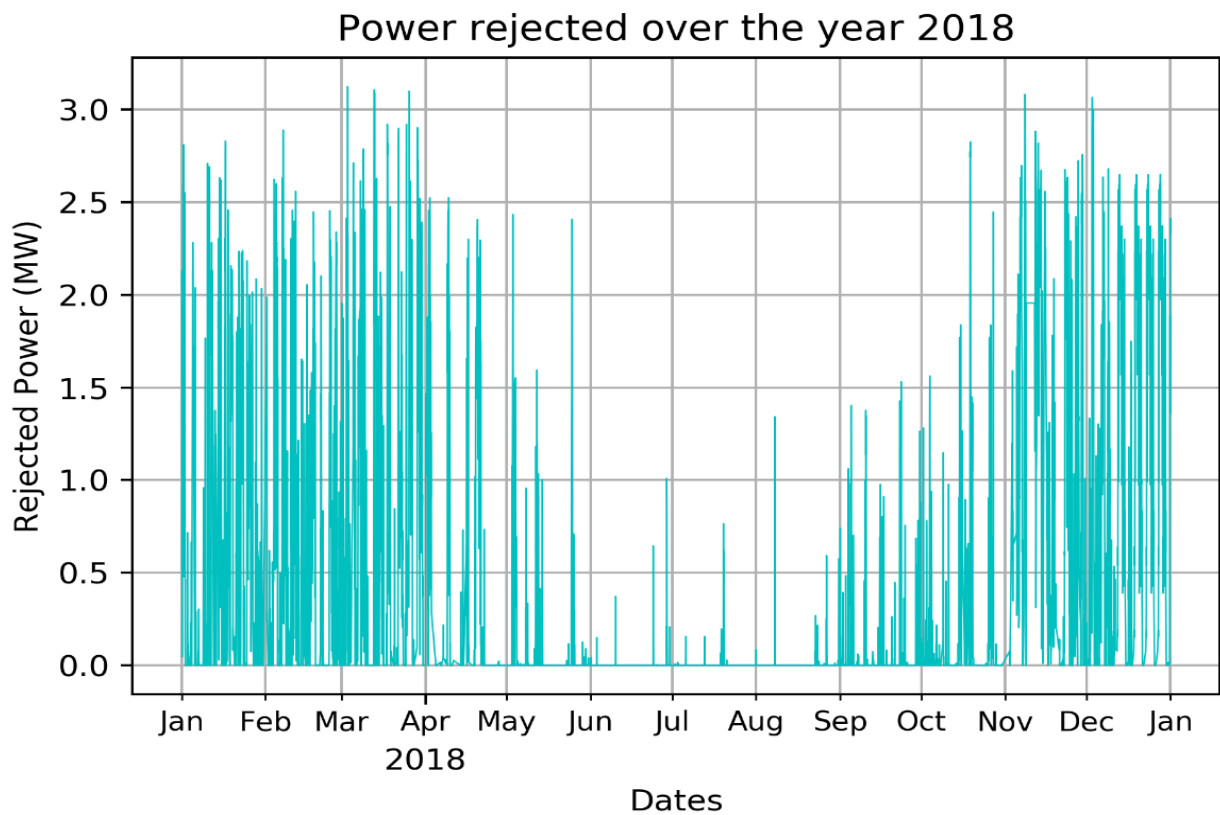


(β)

Σχήμα 6.1 – Διαγράμματα του αιολικού πάρκου 1 (Aeiforiki) για το έτος 2017: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς.



(α)



(β)

Σχήμα 6.2 – Διαγράμματα του αιολικού πάρκου 1 (Aeiforiki) για το έτος 2018: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς.

## 6.2.2 – Αιολικό Πάρκο 2 (Dei)

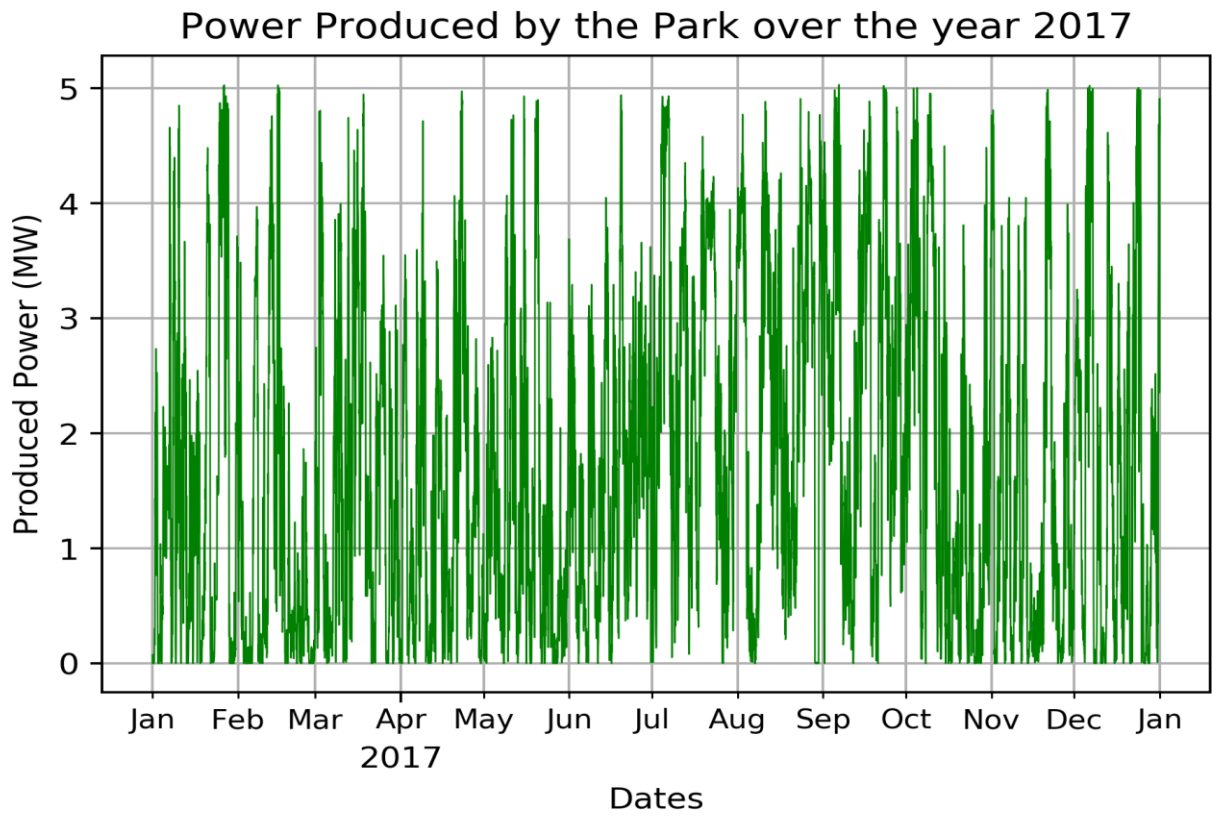
Το αιολικό πάρκο 2 (Dei) έχει εγκατεστημένη (ονομαστική) ισχύ **5,25 MW**. Στον Πίνακα 6.2 δίνονται οι τιμές που υπολογίστηκαν για την ετήσια παραγόμενη από το πάρκο ισχύ και την ετήσια απορριπτόμενη ισχύ, για τα έτη 2017 και 2018 καθώς και συνολικά για τα δύο αυτά έτη. Δίνεται επίσης η τιμή της ετήσιας ισχύος που διοχετεύεται στο δίκτυο, η οποία προκύπτει απλά από το άθροισμα όλων των τιμών της μετρημένης στον υποσταθμό ισχύος που αναφέρονται στο συγκεκριμένο έτος. Υπενθυμίζεται ότι οι τιμές της παραγόμενης και της απορριπτόμενης ισχύος στηρίζονται σε προβλέψεις που πραγματοποιήθηκαν με το μοντέλο SVR, επομένως αποτελούν *εκτιμήσεις* του αντίστοιχου μεγέθους και όχι πραγματικές τιμές. Για τις εκτιμήσεις αυτές δε λαμβάνεται υπόψη το σφάλμα του μοντέλου.

Από τον Πίνακα 6.2 προκύπτει ότι και για αυτό το πάρκο, και για τα δύο έτη η απορριπτόμενη ισχύς αποτελεί ένα αρκετά υψηλό ποσοστό της συνολικής παραγόμενης ισχύος (31% της συνολικής παραγόμενης ισχύος για το 2017 και 35% για το 2018). Παρατηρούμε επίσης ότι το 2018 το πάρκο παρήγαγε περισσότερη ισχύ, γεγονός που πιθανώς οφείλεται σε ευνοϊκότερες συνθήκες ανέμου για το συγκεκριμένο έτος.

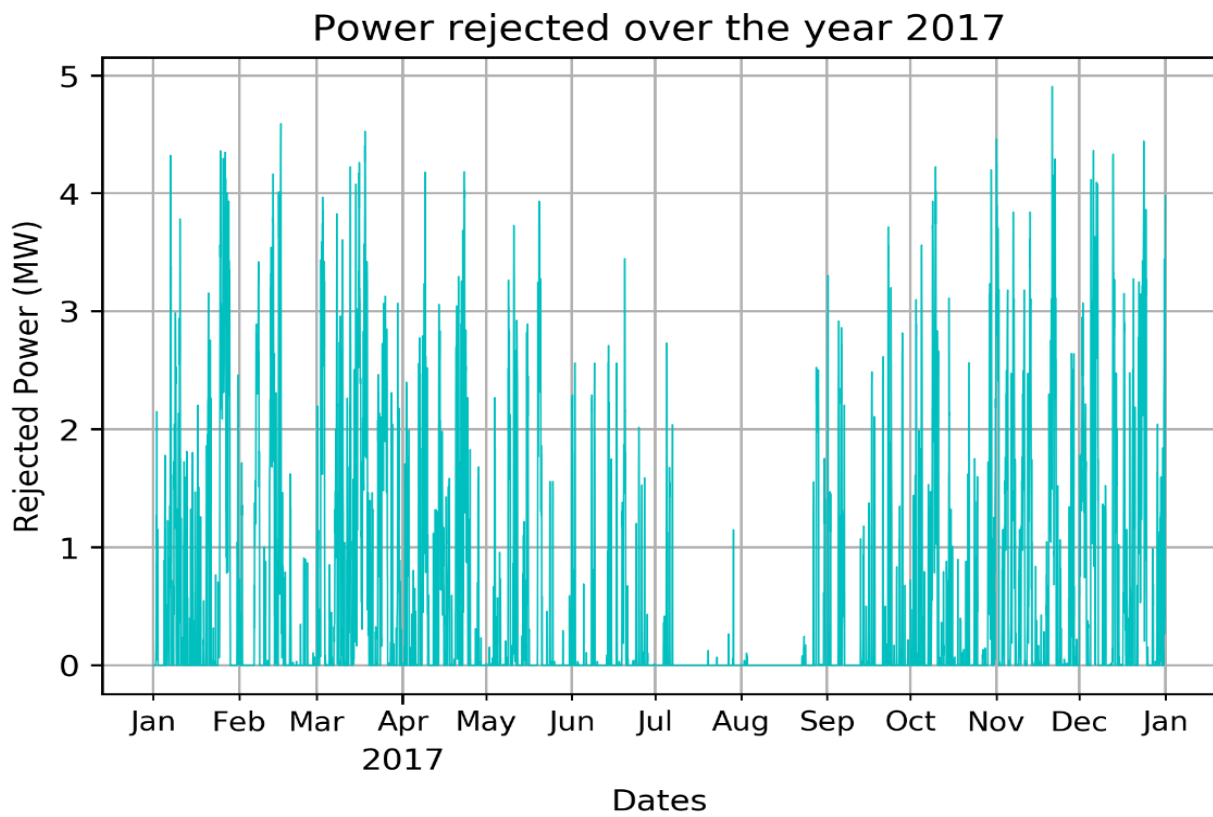
Στα Σχήματα 6.3 και 6.4 απεικονίζεται, για τα έτη 2017 και 2018 αντίστοιχα, η μεταβολή της παραγόμενης και της απορριπτόμενης ισχύος κατά τη διάρκεια του έτους. Αρχικά παρατηρούμε και εδώ τις μεγάλες διακυμάνσεις των μεγεθών αυτών κατά τη διάρκεια του έτους, γεγονός που οφείλεται στον τυχαίο και απρόβλεπτο χαρακτήρα του ανέμου. Σαν γενική συμπεριφορά μπορούμε και εδώ να παρατηρήσουμε (Σχήματα 6.3-β και 6.4-β) ότι και για τα δύο έτη, η απορριπτόμενη ισχύς ελαχιστοποιείται τους καλοκαιρινούς μήνες (Ιούνιο, Ιούλιο και Αύγουστο). Το γεγονός αυτό είναι ενδεικτικό της αύξησης της ζήτησης ηλεκτρικής ενέργειας που παρατηρείται το καλοκαίρι, κυρίως λόγω των αυξημένων αναγκών κλιματισμού των χώρων. Από τα Σχήματα 6.3-α και 6.4-α μπορούμε επίσης να δούμε ότι η ισχύς που παρήγαγε το πάρκο ήταν γενικά αυξημένη τους μήνες Ιούλιο ως Σεπτέμβριο, γεγονός που οφείλεται σε ισχυρότερους ανέμους τους μήνες αυτούς.

**Πίνακας 6.2** – Αποτελέσματα προβλέψεων για το 2ο Αιολικό Πάρκο (Dei): στον πίνακα φαίνονται, για τα έτη 2017 και 2018, οι εκτιμήσεις για την συνολική (ετήσια) παραγόμενη από το πάρκο ισχύ και την απορριπτόμενη ισχύ, όπως προέκυψαν από το βέλτιστο μοντέλο SVR. Δίνεται επίσης η ετήσια ισχύς που διοχετεύεται στο δίκτυο.

Αιολικό Πάρκο 2 (Dei)			
	Παραγόμενη Ισχύς (MW)	Ισχύς που διοχετεύεται στο δίκτυο (MW)	Απορριπτόμενη Ισχύς (MW)
<b>2017</b>	15170,55	10462,30	4764,14
<b>2018</b>	16273,86	10581,12	5747,73
<b>Συνολικά</b>	<b>31444,41</b>	<b>21043,42</b>	<b>10511,87</b>

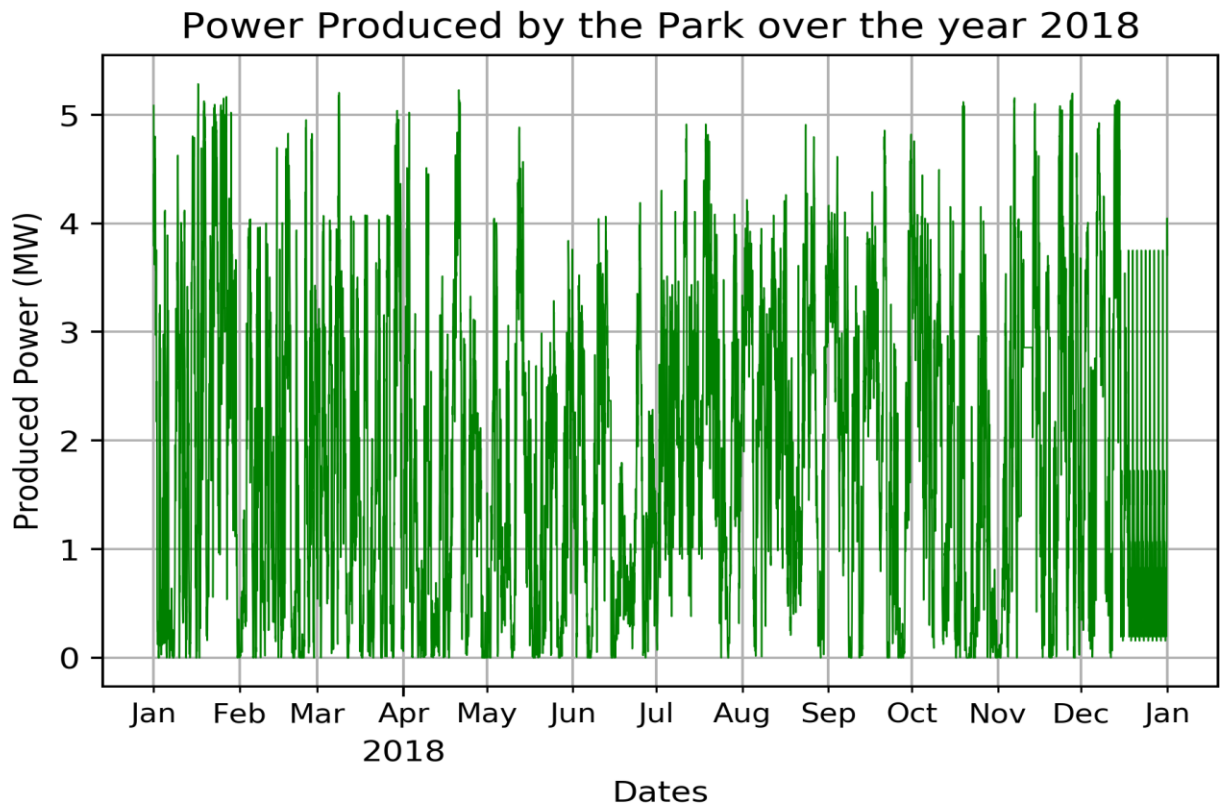


(α)

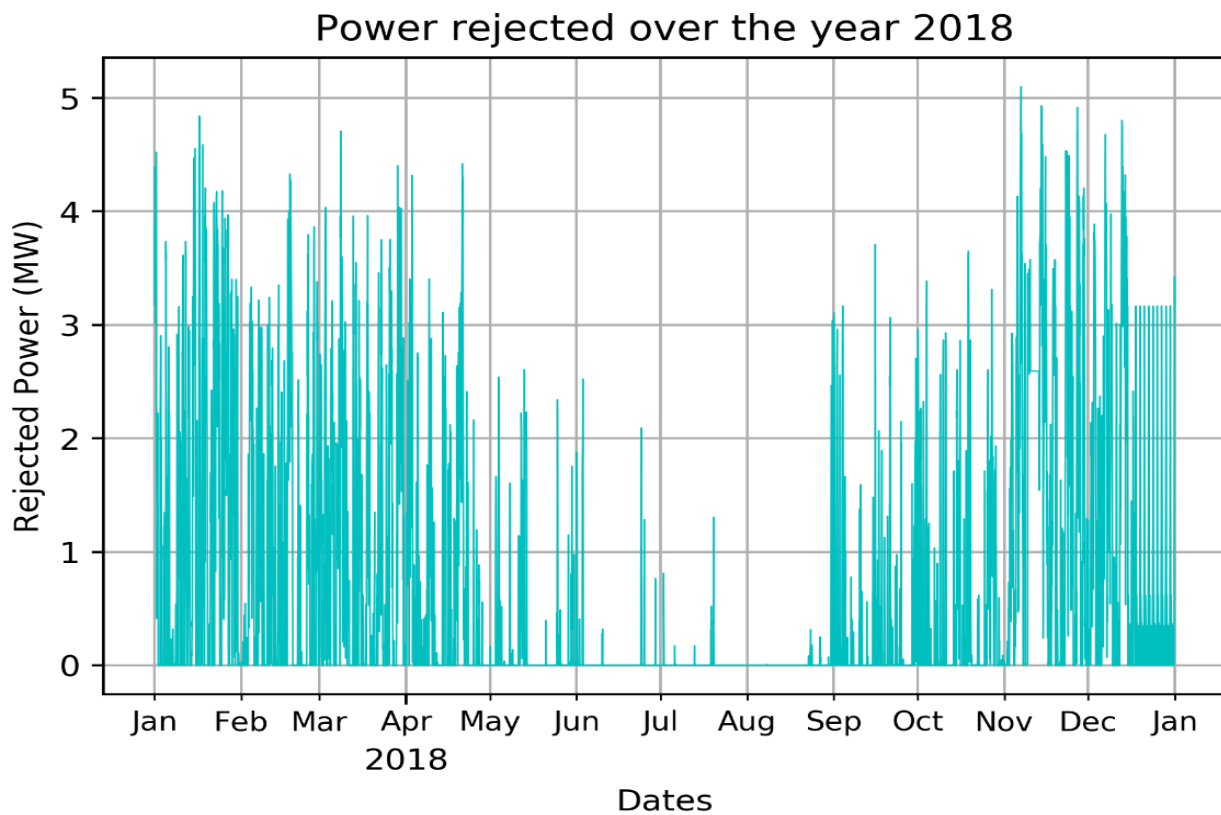


(β)

Σχήμα 6.3 – Διαγράμματα του αιολικού πάρκου 2 (Dei) για το έτος 2017: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς.



(α)



(β)

Σχήμα 6.4 – Διαγράμματα του αιολικού πάρκου 2 (Dei) για το έτος 2018: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς.

## 6.2.3 – Αιολικό Πάρκο 3 (Diethnis)

Το αιολικό πάρκο 3 (Diethnis) έχει εγκατεστημένη (ονομαστική) ισχύ **11,5 MW**. Στον Πίνακα 6.3 δίνονται οι τιμές που υπολογίστηκαν για την ετήσια παραγόμενη από το πάρκο ισχύ και την ετήσια απορριπτόμενη ισχύ, για τα έτη 2017 και 2018 καθώς και συνολικά για τα δύο αυτά έτη. Δίνεται επίσης η τιμή της ετήσιας ισχύος που διοχετεύεται στο δίκτυο, η οποία προκύπτει απλά από το άθροισμα όλων των τιμών της μετρημένης στον υποσταθμό ισχύος που αναφέρονται στο συγκεκριμένο έτος. Υπενθυμίζεται ότι οι τιμές της παραγόμενης και της απορριπτόμενης ισχύος στηρίζονται σε προβλέψεις που πραγματοποιήθηκαν με το μοντέλο SVR, επομένως αποτελούν *εκτιμήσεις* του αντίστοιχου μεγέθους και όχι πραγματικές τιμές. Για τις εκτιμήσεις αυτές δε λαμβάνεται υπόψη το σφάλμα του μοντέλου.

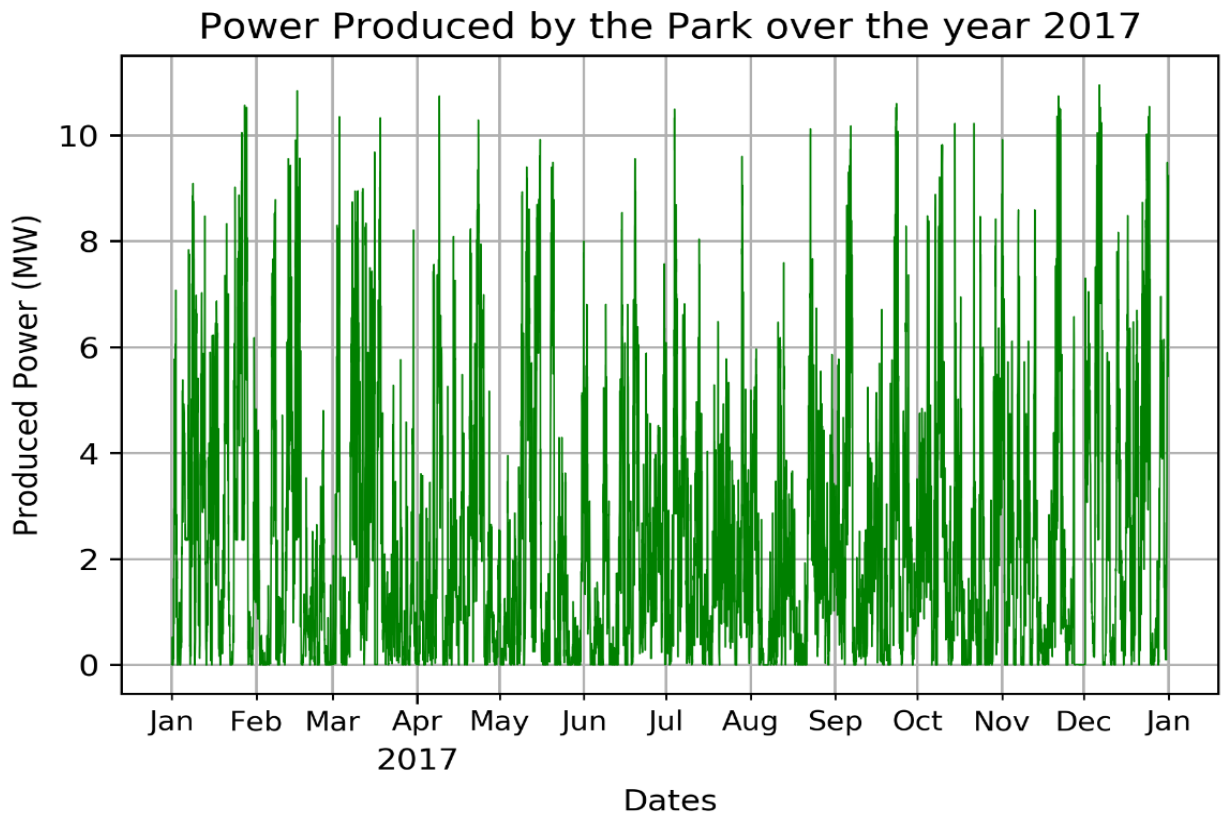
Από τον Πίνακα 6.3 προκύπτει ότι και για αυτό το πάρκο, και για τα δύο έτη η απορριπτόμενη ισχύς αποτελεί ένα αρκετά σημαντικό ποσοστό της συνολικής παραγόμενης ισχύος (34% της συνολικής παραγόμενης ισχύος για το 2017 και 34,1% για το 2018). Παρατηρούμε ότι για αυτό το πάρκο, η ετήσια παραγόμενη ισχύς είναι περίπου η ίδια και για τα δύο έτη.

Στα Σχήματα 6.5 και 6.6 απεικονίζεται, για τα έτη 2017 και 2018 αντίστοιχα, η μεταβολή της παραγόμενης και της απορριπτόμενης ισχύος κατά τη διάρκεια του έτους. Αρχικά παρατηρούμε και εδώ τις μεγάλες διακυμάνσεις των μεγεθών αυτών κατά τη διάρκεια του έτους, γεγονός που οφείλεται στον τυχαίο και απρόβλεπτο χαρακτήρα του ανέμου. Σαν γενική συμπεριφορά μπορούμε και εδώ να παρατηρήσουμε (Σχήματα 6.5-β και 6.6-β) ότι και για τα δύο έτη, η απορριπτόμενη ισχύς ελαχιστοποιείται τους καλοκαιρινούς μήνες (Ιούνιο, Ιούλιο και Αύγουστο). Το γεγονός αυτό είναι ενδεικτικό της αύξησης της ζήτησης ηλεκτρικής ενέργειας που παρατηρείται το καλοκαίρι, κυρίως λόγω των αυξημένων αναγκών κλιματισμού των χώρων. Από τα Σχήματα 6.5-α και 6.6-α μπορούμε επίσης να παρατηρήσουμε ότι η ισχύς που παρήγαγε το πάρκο ήταν γενικά αυξημένη τον μήνα Ιούλιο, γεγονός που πιθανώς οφείλεται σε ισχυρότερους ανέμους τον μήνα αυτό.

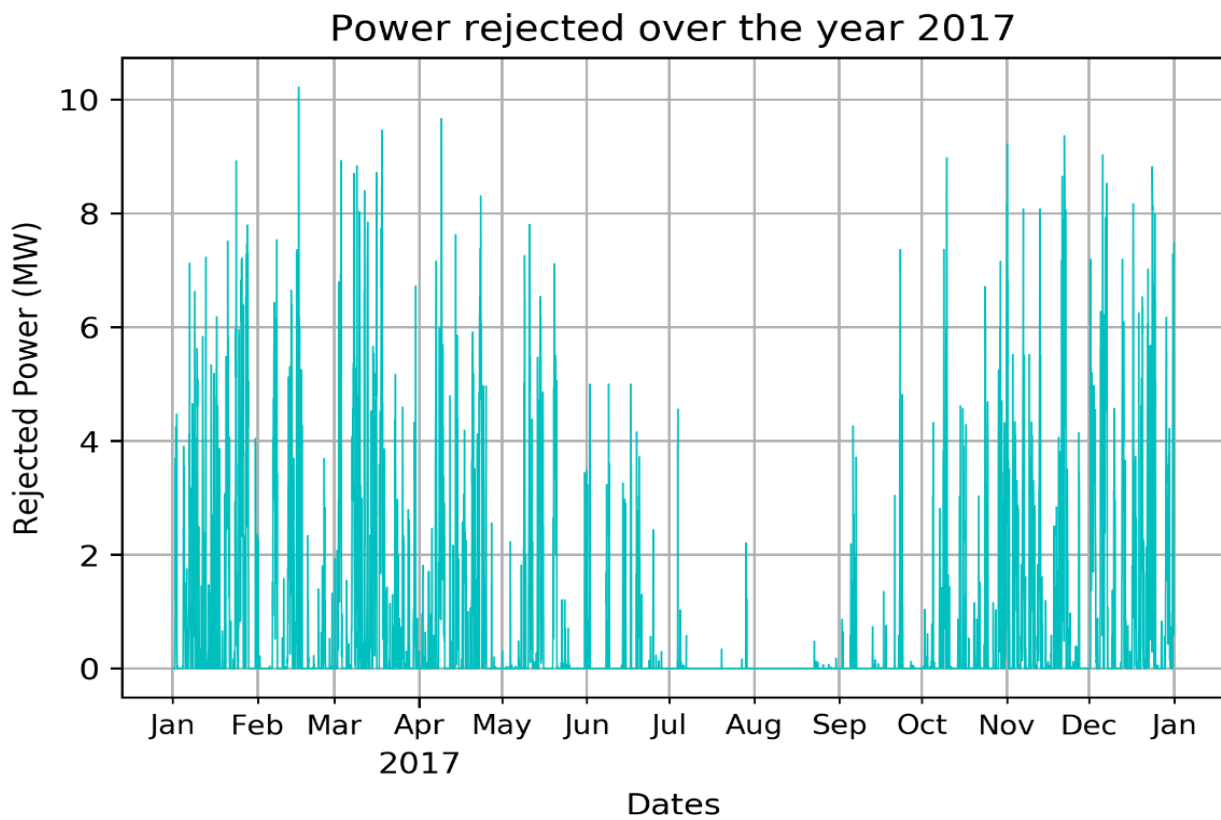
**Πίνακας 6.3** – Αποτελέσματα προβλέψεων για το 3ο Αιολικό Πάρκο (Diethnis): στον πίνακα φαίνονται, για τα έτη 2017 και 2018, οι εκτιμήσεις για την συνολική (ετήσια) παραγόμενη από το πάρκο ισχύ και την απορριπτόμενη ισχύ, όπως προέκυψαν από το βέλτιστο μοντέλο SVR. Δίνεται επίσης η ετήσια ισχύς που διοχετεύεται στο δίκτυο.

Αιολικό Πάρκο 3 (Diethnis)			
	Παραγόμενη Ισχύς (MW)	Ισχύς που διοχετεύεται στο δίκτυο (MW)	Απορριπτόμενη Ισχύς (MW)
<b>2017</b>	20163,85	13475,20	6868,61
<b>2018</b>	20138,40	13583,03	6872,60
<b>Συνολικά</b>	<b>40302,25</b>	<b>27058,23</b>	<b>13741,21</b>



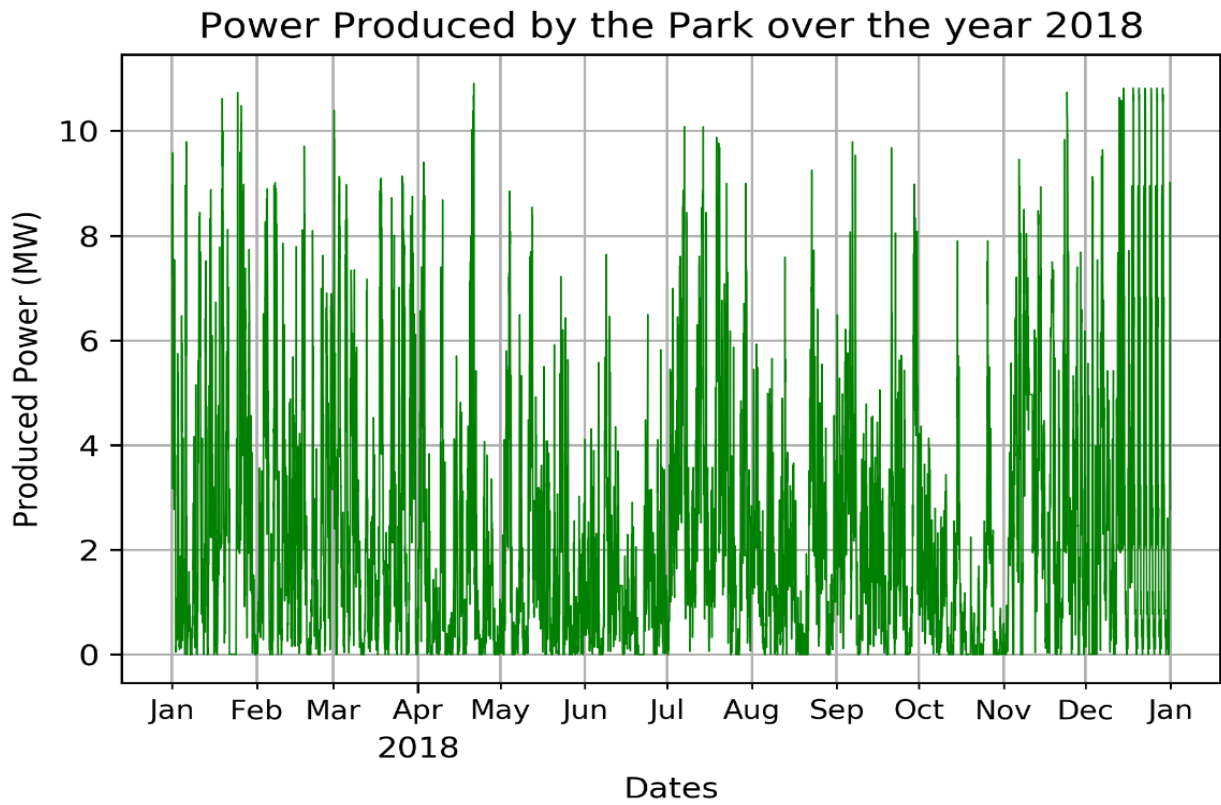


(α)

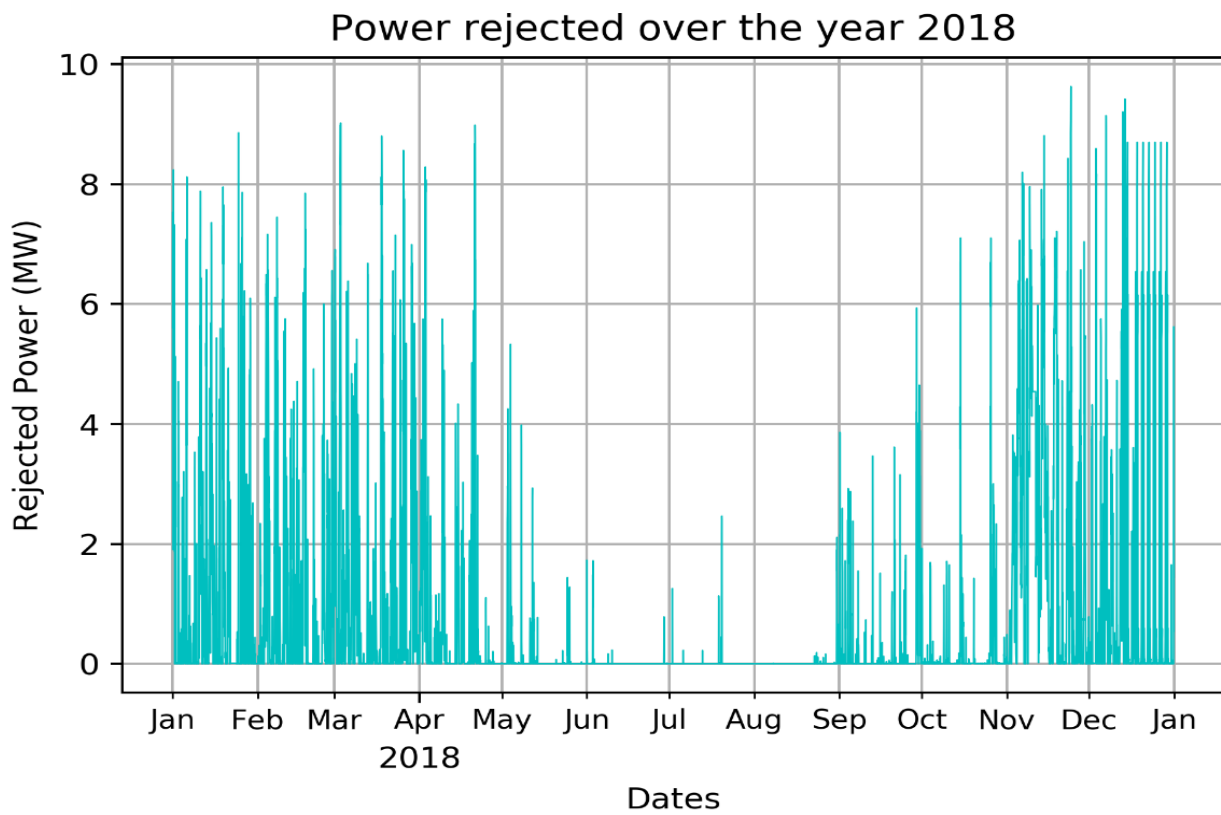


(β)

Σχήμα 6.5 – Διαγράμματα του αιολικού πάρκου 3 (Diethnis) για το έτος 2017: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς.



(α)



(β)

Σχήμα 6.6 – Διαγράμματα του αιολικού πάρκου 3 (Diethnis) για το έτος 2018: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς.

## 6.2.4 – Αιολικό Πάρκο 4 (EuroWind)

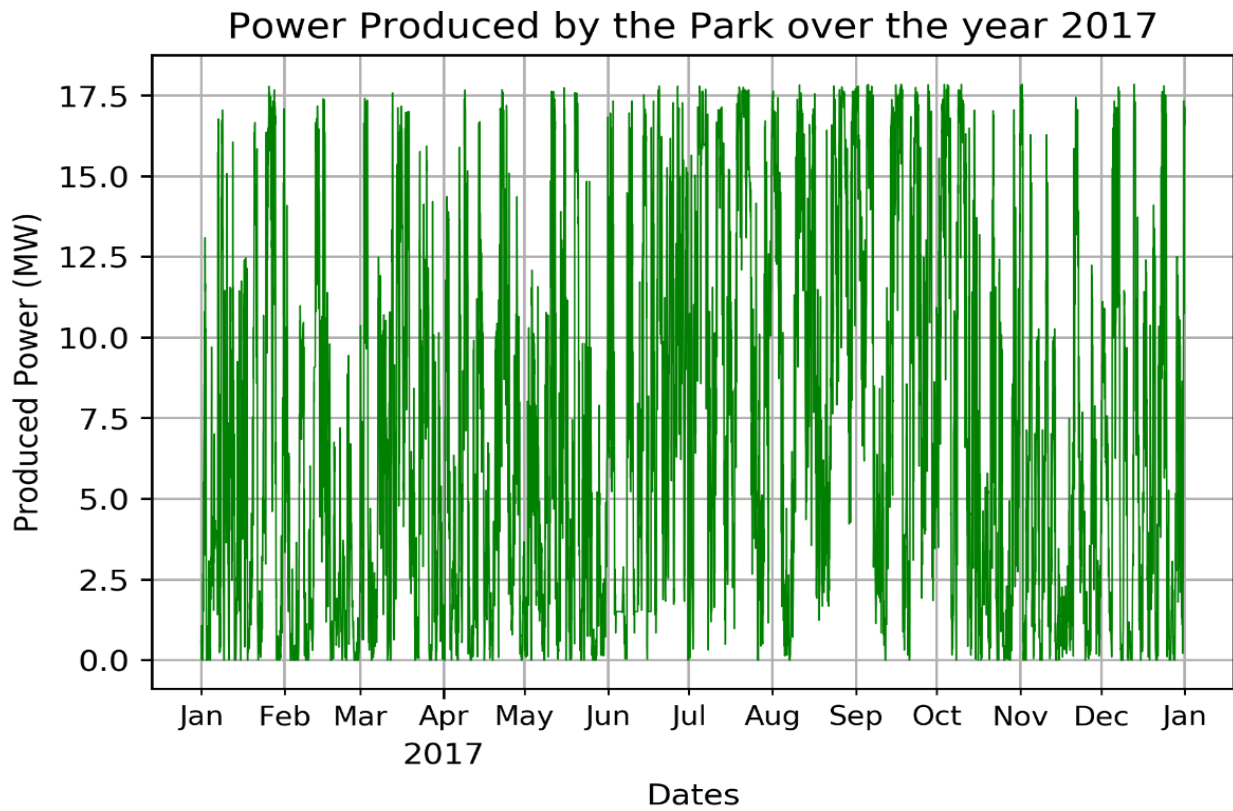
Το αιολικό πάρκο 4 (EuroWind) έχει εγκατεστημένη (ονομαστική) ισχύ **17,8 MW**. Στον Πίνακα 6.4 δίνονται οι τιμές που υπολογίστηκαν για την ετήσια παραγόμενη από το πάρκο ισχύ και την ετήσια απορριπτόμενη ισχύ, για τα έτη 2017 και 2018 καθώς και συνολικά για τα δύο αυτά έτη. Δίνεται επίσης η τιμή της ετήσιας ισχύος που διοχετεύεται στο δίκτυο, η οποία προκύπτει απλά από το άθροισμα όλων των τιμών της μετρημένης στον υποσταθμό ισχύος που αναφέρονται στο συγκεκριμένο έτος. Υπενθυμίζεται ότι οι τιμές της παραγόμενης και της απορριπτόμενης ισχύος στηρίζονται σε προβλέψεις που πραγματοποιήθηκαν με το μοντέλο SVR, επομένως αποτελούν *εκτιμήσεις* του αντίστοιχου μεγέθους και όχι πραγματικές τιμές. Για τις εκτιμήσεις αυτές δε λαμβάνεται υπόψη το σφάλμα του μοντέλου.

Από τον Πίνακα 6.4 προκύπτει ότι και για αυτό το πάρκο, και για τα δύο έτη η απορριπτόμενη ισχύς αποτελεί ένα αρκετά σημαντικό ποσοστό της συνολικής παραγόμενης ισχύος (30% της συνολικής παραγόμενης ισχύος για το 2017 και 34% για το 2018). Παρατηρούμε επίσης ότι το 2018 το πάρκο παρήγαγε περισσότερη ισχύ, γεγονός που πιθανώς οφείλεται σε ευνοϊκότερες συνθήκες ανέμου για το συγκεκριμένο έτος.

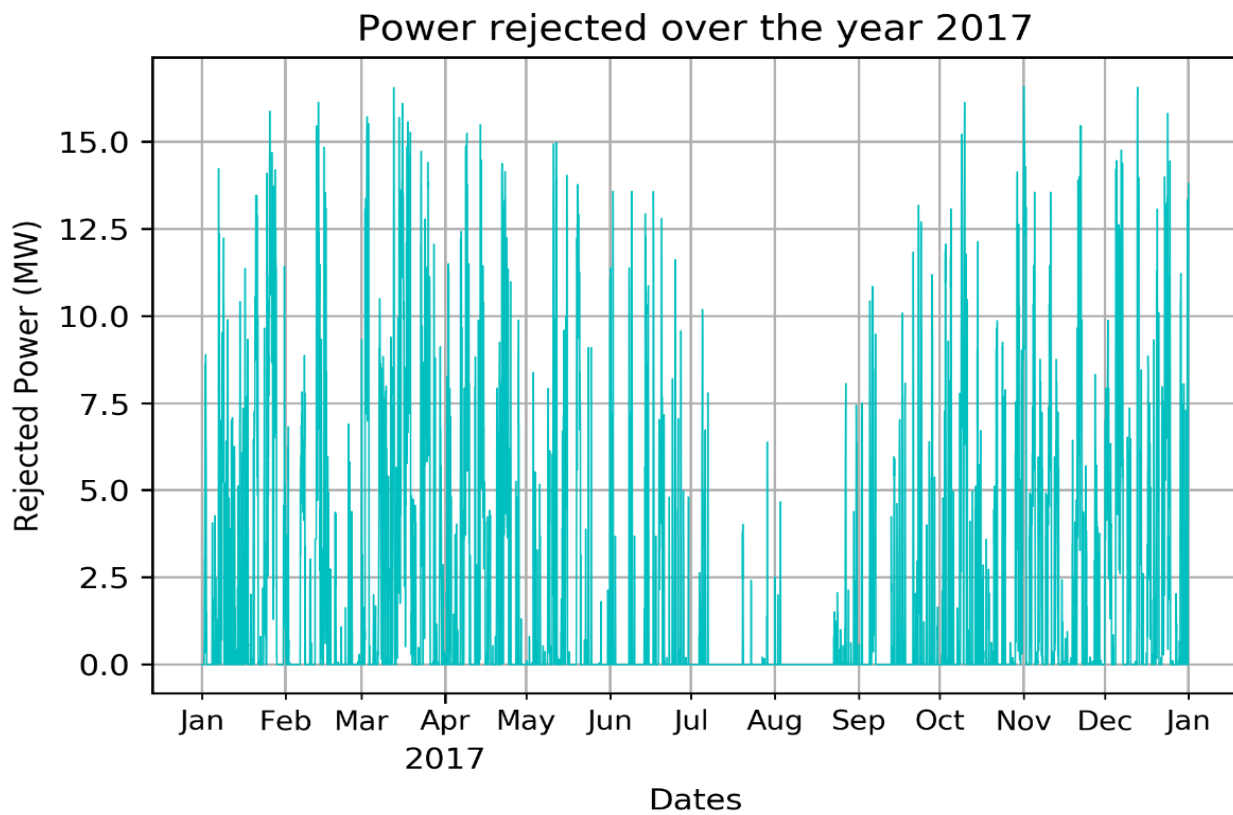
Στα Σχήματα 6.7 και 6.8 απεικονίζεται, για τα έτη 2017 και 2018 αντίστοιχα, η μεταβολή της παραγόμενης και της απορριπτόμενης ισχύος κατά τη διάρκεια του έτους. Αρχικά παρατηρούμε και εδώ τις μεγάλες διακυμάνσεις των μεγεθών αυτών κατά τη διάρκεια του έτους, γεγονός που οφείλεται στον τυχαίο και απρόβλεπτο χαρακτήρα του ανέμου. Σαν γενική συμπεριφορά μπορούμε και εδώ να παρατηρήσουμε (Σχήματα 6.7-β και 6.8-β) ότι και για τα δύο έτη, η απορριπτόμενη ισχύς ελαχιστοποιείται τους καλοκαιρινούς μήνες (Ιούνιο, Ιούλιο και Αύγουστο). Το γεγονός αυτό είναι ενδεικτικό της αύξησης της ζήτησης ηλεκτρικής ενέργειας που παρατηρείται το καλοκαίρι, κυρίως λόγω των αυξημένων αναγκών κλιματισμού των χώρων. Από τα Σχήματα 6.7-α και 6.8-α μπορούμε επίσης να παρατηρήσουμε ότι η ισχύς που παρήγαγε το πάρκο ήταν γενικά αυξημένη τους μήνες Ιούλιο ως Σεπτέμβριο, γεγονός που οφείλεται σε ισχυρότερους ανέμους τους μήνες αυτούς.

**Πίνακας 6.4** – Αποτελέσματα προβλέψεων για το 4ο Αιολικό Πάρκο (EuroWind): στον πίνακα φαίνονται, για τα έτη 2017 και 2018, οι εκτιμήσεις για την συνολική (ετήσια) παραγόμενη από το πάρκο ισχύ και την απορριπτόμενη ισχύ, όπως προέκυψαν από το βέλτιστο μοντέλο SVR. Δίνεται επίσης η ετήσια ισχύς που διοχετεύεται στο δίκτυο.

Αιολικό Πάρκο 4 (EuroWind)			
	Παραγόμενη Ισχύς (MW)	Ισχύς που διοχετεύεται στο δίκτυο (MW)	Απορριπτόμενη Ισχύς (MW)
<b>2017</b>	62073,53	43641,44	18723,84
<b>2018</b>	67930,45	44978,13	23231,32
<b>Συνολικά</b>	<b>130003,98</b>	<b>88619,57</b>	<b>41955,16</b>

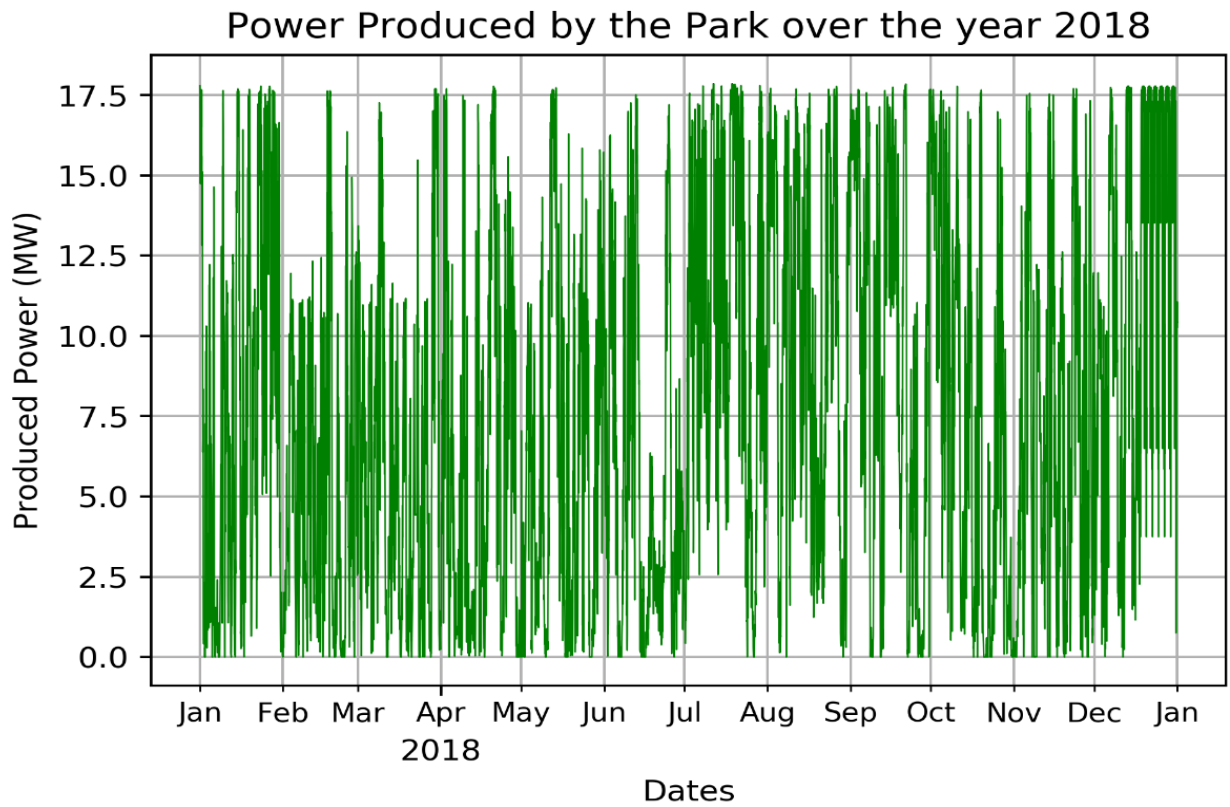


(α)

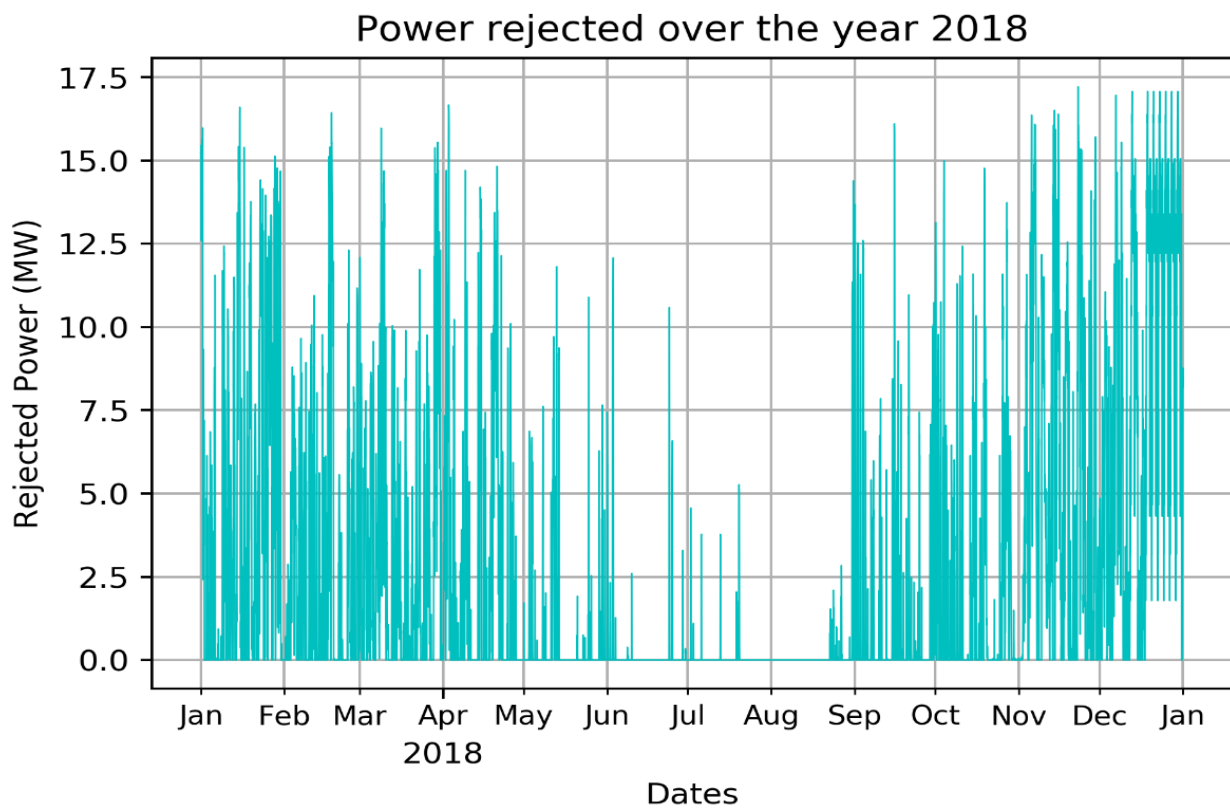


(β)

Σχήμα 6.7 – Διαγράμματα του αιολικού πάρκου 4 (EuroWind) για το έτος 2017: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς.



(α)



(β)

Σχήμα 6.8 – Διαγράμματα του αιολικού πάρκου 4 (EuroWind) για το έτος 2018: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς.

## 6.2.5 – Αιολικό Πάρκο 5 (Atabyrou)

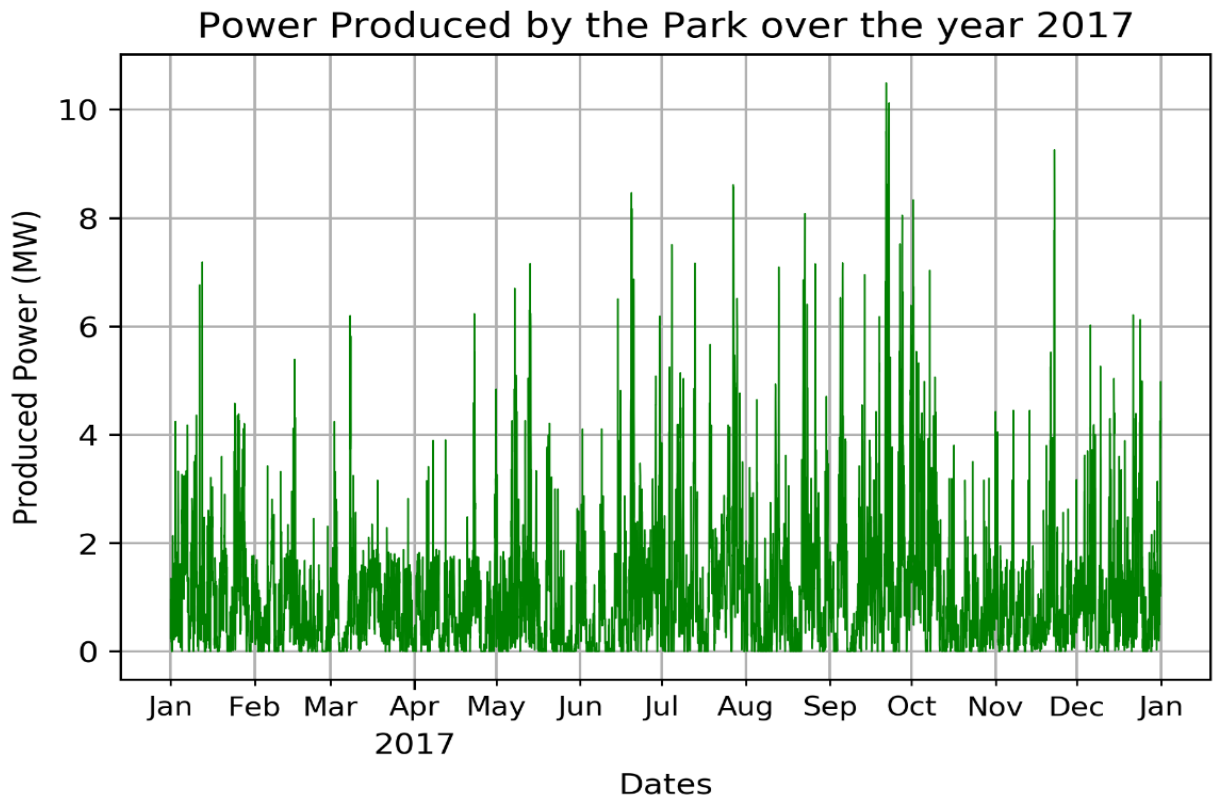
Το αιολικό πάρκο 5 (Atabyrou) έχει εγκατεστημένη (ονομαστική) ισχύ **11,7 MW**. Στον Πίνακα 6.5 δίνονται οι τιμές που υπολογίστηκαν για την ετήσια παραγόμενη και απορριπτόμενη ισχύ από το πάρκο, για τα έτη 2017 και 2018 καθώς και συνολικά για τα δύο αυτά έτη. Δίνεται επίσης η τιμή της ετήσιας ισχύος που διοχετεύεται στο δίκτυο. Υπενθυμίζεται και εδώ ότι οι τιμές της παραγόμενης και της απορριπτόμενης ισχύος στηρίζονται σε προβλέψεις, επομένως αποτελούν απλά εκτιμήσεις του αντίστοιχου μεγέθους και όχι πραγματικές τιμές.

Από τον Πίνακα 6.5 παρατηρούμε ότι η ετήσια παραγόμενη ισχύς, και για τα δύο έτη, είναι μικρότερη από την μετρημένη ετήσια ισχύ που διοχετεύτηκε στο δίκτυο. Η αιτία για αυτό προκύπτει αν παρατηρήσουμε τις προβλέψεις ισχύος που πραγματοποιήθηκαν για αυτό το πάρκο. Συγκεκριμένα, για ένα μεγάλο αριθμό δειγμάτων (λίγο πάνω από τα μισά), η τιμή της παραγόμενης ισχύος του πάρκου που εκτιμήθηκε από το μοντέλο SVR προέκυψε μικρότερη από τη μετρημένη τιμή της ισχύος στον υποσταθμό. Για αυτό και η συνολική τιμή της παραγόμενης ισχύος προκύπτει μικρότερη από την τιμή της ισχύος που διοχετεύτηκε στο δίκτυο. Οι πιθανές αιτίες για αυτή τη συμπεριφορά είναι δύο: είτε τα δεδομένα που έχουμε για αυτό το πάρκο δεν είναι επαρκή για να εκπαιδεύσουν σωστά το μοντέλο, είτε το μοντέλο SVR που χρησιμοποιούμε δεν είναι κατάλληλο για το συγκεκριμένο πάρκο. Δεδομένου όμως ότι αυτό το μοντέλο αποδείχθηκε αξιόπιστο για όλα τα υπόλοιπα πάρκα, μπορούμε με σχετική ασφάλεια να καταλήξουμε ότι το πρόβλημα εδώ είναι η ελλιπής εκπαίδευση του μοντέλου, λόγω ανεπάρκειας του συνόλου εκπαίδευσης (training set).

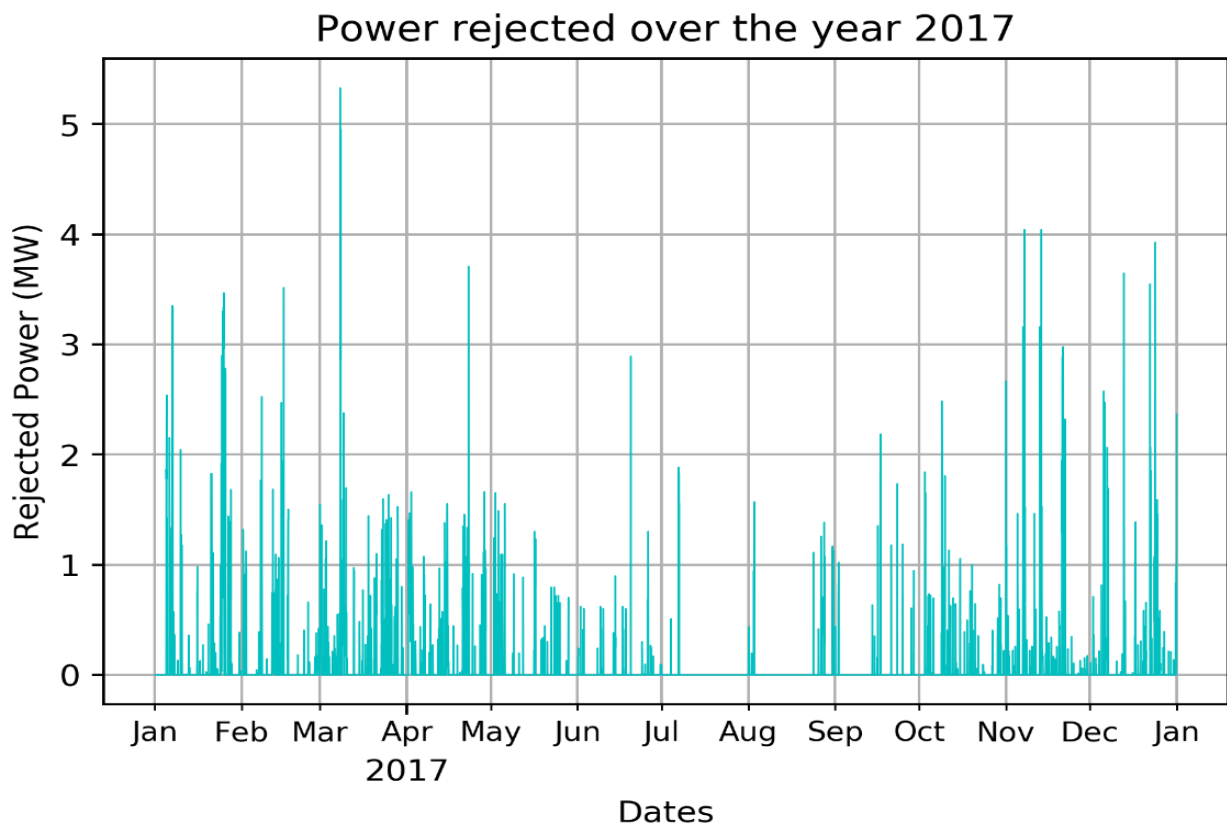
Στα Σχήματα 6.9 και 6.10 απεικονίζεται, για τα έτη 2017 και 2018 αντίστοιχα, η μεταβολή της παραγόμενης και της απορριπτόμενης ισχύος κατά τη διάρκεια του έτους. Μπορούμε και εδώ να διαπιστώσουμε τις γενικές συμπεριφορές που παρουσιάστηκαν και στα άλλα πάρκα, δηλαδή την ελαχιστοποίηση της απορριπτόμενης ισχύος τους μήνες Ιούνιο, Ιούλιο, Αύγουστο (Σχήματα 6.9-β και 6.10-β) και την σχετικά αυξημένη παραγωγή ισχύος τους μήνες Ιούλιο ως Σεπτέμβριο (Σχήματα 6.9-α και 6.10-α).

**Πίνακας 6.5** – Αποτελέσματα προβλέψεων για το 5ο Αιολικό Πάρκο (Atabyrou): στον πίνακα φαίνονται, για τα έτη 2017 και 2018, οι εκτιμήσεις για την συνολική (ετήσια) παραγόμενη από το πάρκο ισχύ και την απορριπτόμενη ισχύ, όπως προέκυψαν από το βέλτιστο μοντέλο SVR. Δίνεται επίσης η ετήσια ισχύς που διοχετεύεται στο δίκτυο.

Αιολικό Πάρκο 5 (Atabyrou)			
	Παραγόμενη Ισχύς (MW)	Ισχύς που διοχετεύεται στο δίκτυο (MW)	Απορριπτόμενη Ισχύς (MW)
<b>2017</b>	9394,01	13816,26	859,65
<b>2018</b>	10406,98	13289,80	1278,92
<b>Συνολικά</b>	<b>19800,99</b>	<b>27106,06</b>	<b>2138,57</b>

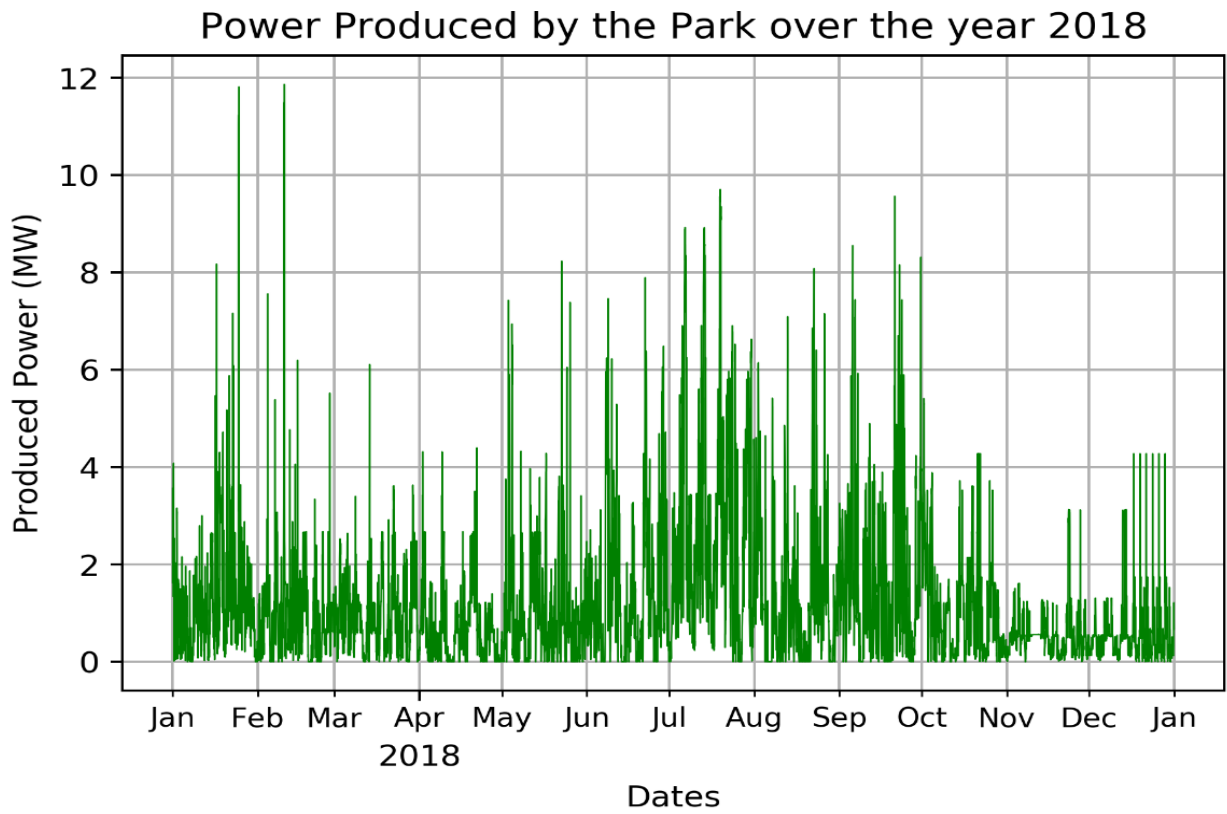


(α)

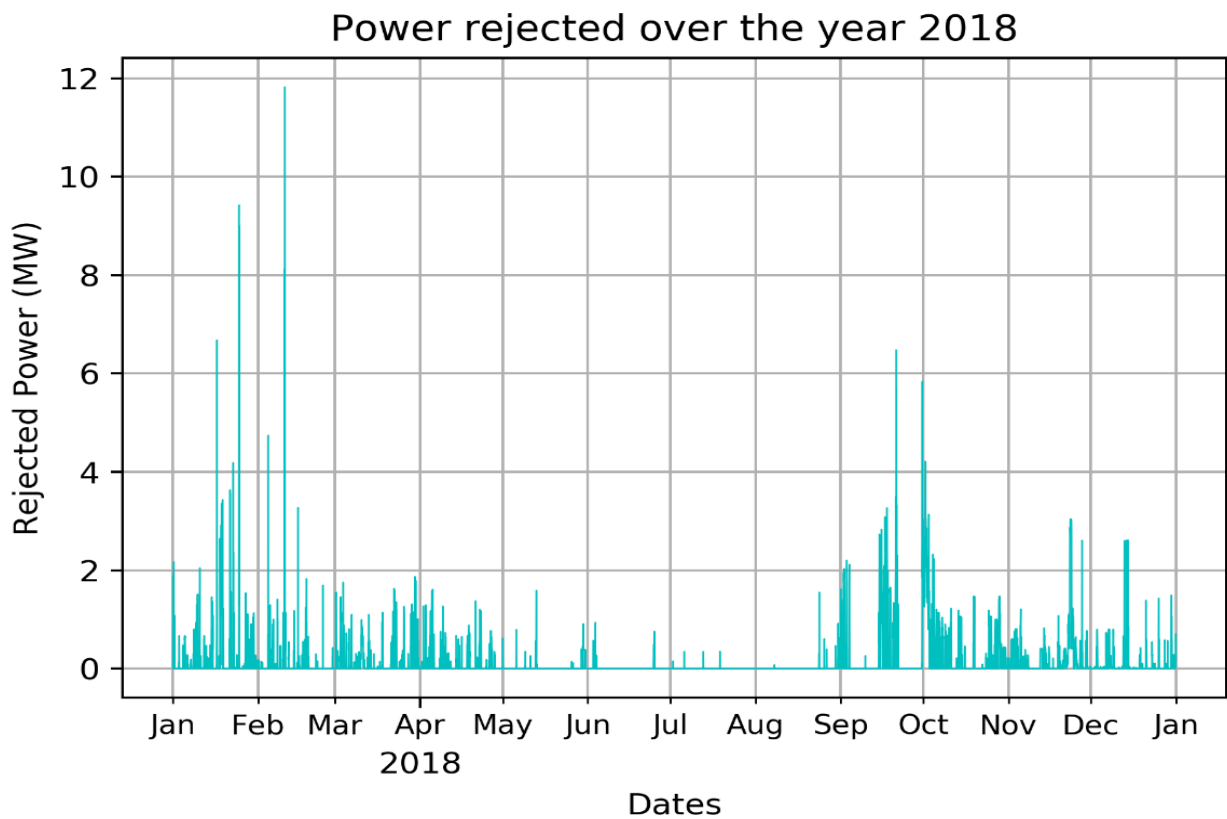


(β)

Σχήμα 6.9 – Διαγράμματα του αιολικού πάρκου 5 (Atabyrou) για το έτος 2017: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς.



(α)



(β)

**Σχήμα 6.10** – Διαγράμματα του αιολικού πάρκου 5 (Atabyrou) για το έτος 2018: α) Συνολική Παραγόμενη Ισχύς του πάρκου. β) Απορριπτόμενη Ισχύς.



### 6.3 – Συμπεράσματα

Έπειτα από τις προβλέψεις που πραγματοποιήθηκαν και την ανάλυση που προηγήθηκε στις προηγούμενες ενότητες, μπορούμε να εξάγουμε κάποια γενικά συμπεράσματα για τα αιολικά πάρκα της Ρόδου.

Όσον αφορά την ποσότητα της ετήσιας απορριπτόμενης ισχύος του κάθε πάρκου, παρατηρούμε ότι αυτή είναι σχετικά υψηλή, καθώς για όλα τα πάρκα αποτελεί ένα ποσοστό της τάξεως του 30% - 35% της ετήσιας συνολικής παραγόμενης ισχύος. Η μεγαλύτερη τιμή της ετήσιας απορριπτόμενης ισχύος προκύπτει για το πάρκο 1 (Aeiforiki), για το οποίο το αντίστοιχο ποσοστό φτάνει το 40%. Τα στοιχεία αυτά φανερώνουν ότι ένα μεγάλο ποσοστό της ισχύος των πάρκων μένει ανεκμετάλλευτο από το δίκτυο, και επομένως θα πρέπει να βρεθούν άλλοι τρόποι αξιοποίησης αυτής της περίσσειας ενέργειας ή αποτελεσματικά μέσα αποθήκευσης της.

Όσον αφορά την παραγωγή ισχύος για τα έτη 2017 και 2018, προέκυψε ότι όλα τα πάρκα παρήγαγαν περισσότερη ισχύ το έτος 2018 από ότι το 2017 ( με εξαίρεση το πάρκο 3 για το οποίο η παραγόμενη ισχύς ήταν περίπου ίδια και τα δύο έτη). Από αυτή την παρατήρηση μπορούμε να συμπεράνουμε ότι το έτος 2018 ήταν γενικά πιο ευνοϊκό από άποψη ανεμολογικών συνθηκών, πιθανώς με ισχυρότερους και μεγαλύτερης διάρκειας ανέμους.

Παρατηρούμε επίσης για όλα τα πάρκα της Ρόδου ότι η απορριπτόμενη ισχύς ελαχιστοποιείται τους καλοκαιρινούς μήνες, κυρίως τον Ιούνιο, Ιούλιο και Αύγουστο, γεγονός που όπως προαναφέρθηκε οφείλεται στην αύξηση της ζήτησης ηλεκτρικής ενέργειας που παρατηρείται το καλοκαίρι, κυρίως λόγω των αυξημένων αναγκών κλιματισμού των χώρων. Οι αυξημένες ανάγκες σε ενέργεια αυτούς τους μήνες σημαίνουν ότι το πάρκο διοχετεύει περισσότερη ενέργεια στο δίκτυο (πχ λειτουργεί για μεγαλύτερα διαστήματα με Set Point = 1, δηλαδή τροφοδοτώντας το δίκτυο με το σύνολο της ενέργειας που παράγει), με αποτέλεσμα τη μείωση της απορριπτόμενης ενέργειας. Ένα ακόμη συμπέρασμα που προκύπτει από τα διαγράμματα είναι ότι η παραγόμενη ισχύς είχε την τάση να είναι μεγαλύτερη τους μήνες Ιούλιο ως Σεπτέμβριο, λίγο πολύ για όλα τα πάρκα. Αυτό κατά πάσα πιθανότητα οφείλεται στους ισχυρότερους ανέμους αυτής της περιόδου, καθώς οι μήνες Ιούλιος, Αύγουστος και Σεπτέμβριος είναι συνήθως αυτοί τους οποίους πνέουν τα μελέμια στο Αιγαίο, με μέση ένταση ανέμου τα 6-7 μποφόρ.



### 7.1 – Γενικά Συμπεράσματα

Στα πλαίσια τη παρούσας μεταπτυχιακής εργασίας πραγματοποιήθηκε μια εκτίμηση της δυνατότητας παραγωγής ισχύος των 5 αιολικών πάρκων της Ρόδου, καθώς και της ετήσιας απορριπτόμενης ισχύος τους, δηλαδή της ισχύος που παράγουν τα πάρκα αλλά δεν χρησιμοποιείται από το δίκτυο γιατί ο διαχειριστής τους δικτύου ζητάει λιγότερη ισχύ. Η δυνατότητα διεξαγωγής αξιόπιστων προβλέψεων της απορριπτόμενης ισχύος είναι πολύ σημαντική, καθώς μπορεί να βοηθήσει στη σχεδίαση ενός αποτελεσματικού τρόπου αποθήκευσης της περίσσειας ισχύος ώστε να χρησιμοποιηθεί για την κάλυψη φορτίων αιχμής ή για άλλους σκοπούς. Οι προβλέψεις πραγματοποιήθηκαν με τεχνικές μηχανικής μάθησης (machine learning): χρησιμοποιώντας τα ανεμολογικά δεδομένα από γνωστές χρονοσειρές του ανέμου για τα έτη 2017 και 2018, καθώς και τις αντίστοιχες μετρήσεις ισχύος στον υποσταθμό με τον οποίο συνδέεται το κάθε πάρκο, εκπαιδεύτηκαν μοντέλα παλινδρόμησης τα οποία στη συνέχεια χρησιμοποιήθηκαν για διεξαγωγή προβλέψεων. Η διαδικασία περιγράφεται αναλυτικά στο *Κεφάλαιο 3*.

Για την εύρεση του βέλτιστου μοντέλου για το πρόβλημά μας εκπαιδεύτηκαν διάφορα μοντέλα παλινδρόμησης, μεταξύ των οποίων απλά γραμμικά μοντέλα (όπως τα Ridge και Lasso), μηχανές διανυσμάτων υποστήριξης (Support Vector Machines), μοντέλα συνόλων (Random Forest, AdaBoost, Gradient Boosting) καθώς και νευρωνικά δίκτυα (Multi-Layer Perceptron). Για το κάθε μοντέλο ρυθμίστηκαν οι υπερ-παραμέτροι στη βέλτιστη τιμή τους, και στη συνέχεια το μοντέλο αξιολογήθηκε ως προς την ικανότητά του να εκτελεί προβλέψεις ισχύος. Τα μοντέλα που προσαρμόστηκαν καλύτερα στα δεδομένα εκπαίδευσης και αποδείχθηκαν πιο αποτελεσματικά για το πρόβλημά μας ήταν το μοντέλο παλινδρόμησης *Διανυσμάτων Υποστήριξης (Support Vector Regression)*, το μοντέλο *k-κοντινότερων γειτόνων (K-Nearest Neighbors)* και το *Random Forest*. Τα τρία αυτά μοντέλα εξετάστηκαν πιο διεξοδικά για να βρεθεί ο τρόπος που η μεταβολή των υπερ-παραμέτρων τους επηρεάζει το μέσο απόλυτο σφάλμα πρόβλεψης και το χρόνο εκπαίδευσης του μοντέλου. Διερευνήθηκε επίσης η συμπεριφορά τους ως προς φαινόμενα Overfitting και Underfitting. Από την ανάλυση αυτή προέκυψε ότι το βέλτιστο μοντέλο για το πρόβλημά μας είναι ο παλινδρομητής *Διανυσμάτων Υποστήριξης (SVR)* με τιμές των υπερ-παραμέτρων:  $kernel = 'rbf'$ ,  $gamma = 50$ ,  $C = 100$ ,  $epsilon = 0,01$ . Το μοντέλο αυτό μπορεί να έχει σχετικά υψηλότερους χρόνους εκπαίδευσης σε σχέση με τα άλλα δύο μοντέλα, παρουσιάζει όμως το μικρότερο μέσο απόλυτο σφάλμα πρόβλεψης και την καλύτερη συνολική συμπεριφορά ως προς την προσαρμογή στα δεδομένα του προβλήματος. Επομένως, αυτό το μοντέλο χρησιμοποιήθηκε τελικά για τη διεξαγωγή προβλέψεων ισχύος στα αιολικά πάρκα.

Στα πλαίσια των προβλέψεων πραγματοποιήθηκαν σε κάθε πάρκο εκτιμήσεις της ετήσιας παραγόμενης ισχύος του πάρκου (δυνατότητα παραγωγής ισχύος), καθώς και της απορριπτόμενης ισχύος, δηλαδή αυτής που δε χρησιμοποιείται από το δίκτυο. Επισημαίνεται ότι

για τις εκτιμήσεις της απορριπτόμενης ισχύος δε λάβαμε υπόψη το σφάλμα του μοντέλου, το οποίο θεωρείται αμελητέο για το βέλτιστο μοντέλο που χρησιμοποιήσαμε. Κάποια από τα συμπεράσματα που προέκυψαν για όλα τα πάρκα είναι ότι η ποσότητα απορριπτόμενης ισχύος ελαχιστοποιείται τους καλοκαιρινούς μήνες (Ιούνιο, Ιούλιο και Αύγουστο), λόγω των αυξημένων αναγκών σε ενέργεια αυτής της περιόδου, ενώ η παραγόμενη ισχύς των πάρκων είναι γενικά μεγαλύτερη από τον Ιούλιο ως και τον Σεπτέμβριο, που είναι η περίοδος ισχυρότερων και με μεγαλύτερη διάρκεια ανέμων στο Αιγαίο (τα λεγόμενα μελτέμια). Διαπιστώθηκε επίσης ότι το έτος 2018 ήταν γενικά πιο ευνοϊκό από άποψη ανεμολογικών συνθηκών, γεγονός που οδήγησε σε μεγαλύτερη ετήσια παραγωγή ισχύος από τα πάρκα σε σύγκριση με το 2017. Τέλος, αξίζει να σημειωθεί ότι ετήσια απορριπτόμενη ισχύς κάθε πάρκου προέκυψε ότι είναι ένα σημαντικό ποσοστό (της τάξεως του 30% - 40%) της συνολικής ετήσιας παραγόμενης ισχύος. Αυτό φανερώνει την ανάγκη αποτελεσματικής αξιοποίησης αυτής της περίσσειας ενέργειας, ή την εύρεση αποτελεσματικών τρόπων αποθήκευσής της.

### 7.2 – Ανοιχτά Θέματα

Στην ενότητα αυτή παρουσιάζονται κάποιες προτάσεις για πιθανές μελλοντικές εργασίες, οι οποίες θα συμπληρώνουν την παρούσα εργασία και θα δρουν ως προέκταση αυτής.

Όσον αφορά τα μοντέλα παλινδρόμησης που δοκιμάστηκαν, θα μπορούσαν να εξεταστούν και άλλα μοντέλα για να διαπιστωθεί το πως προσαρμόζονται στο πρόβλημά μας, και ενδεχομένως να βρεθεί ένα μοντέλο ακόμη καλύτερο από το SVR. Κάποια μοντέλα που δεν εξετάστηκαν στα πλαίσια της παρούσας εργασίας, και θα μπορούσαν να εξεταστούν στο μέλλον, είναι το μοντέλο *LARS Lasso*, δηλαδή το γραμμικό μοντέλο Lasso με χρήση του αλγορίθμου LARS (Least-angle Regression), το μοντέλο *ARD (Automatic Relevance Determination)*, γνωστό και ως Sparse Bayesian Learning ή Relevance Vector Machine, ο αλγόριθμος *Passive-Aggressive Regression*, το μοντέλο *μερικών ελαχίστων τετραγώνων (Partial Least Squares Regression)*, το *Γκαουσιανό μοντέλο παλινδρόμησης GPR (Gaussian Process Regression)*, το μοντέλο *Voting Regression*, που ανήκει στην κατηγορία των μεθόδων συνόλων (ensemble methods), και το μοντέλο *ισοτονικής παλινδρόμησης (Isotonic Regression)*.

Κάποια από τα μοντέλα που δοκιμάστηκαν στην παρούσα εργασία θα μπορούσαν επίσης να εξεταστούν περεταίρω, σε ένα μεγαλύτερο εύρος των υπερ-παραμέτρων τους, για να διαπιστωθεί αν μπορεί να προκύψει μια ακόμη καλύτερη εκδοχή τους. Ένα τέτοιο μοντέλο είναι το *νευρωνικό δίκτυο MLP (Multi-layer Perceptron)*, για το οποίο θα μπορούσαν να δοκιμαστούν πολλοί διαφορετικοί συνδυασμοί αριθμού νευρώνων και κρυμμένων επιπέδων. Πολύπλοκα μοντέλα νευρωνικών δικτύων (δηλαδή με πολλά κρυφά επίπεδα και μεγάλο αριθμό νευρώνων ανά επίπεδο) πιθανώς να παρουσιάζουν ακόμη καλύτερη συμπεριφορά από το σχετικά απλό μοντέλο των τριών κρυφών επιπέδων που εξετάστηκε στην παρούσα εργασία. Ένα ακόμη μοντέλο που επιδέχεται βελτίωση είναι το μοντέλο *Διανυσμάτων Υποστήριξης (Support Vector*

Regression), για το οποίο μας δίνεται η δυνατότητα να κατασκευάσουμε τη δική μας συνάρτηση πυρήνα [66], η οποία πιθανώς να οδηγήσει σε καλύτερο μοντέλο.

Όσον αφορά τα κριτήρια τα οποία χρησιμοποιήθηκαν για τη βέλτιστη ρύθμιση των υπερ-παραμέτρων των μοντέλων, στόχος στην παρούσα εργασία ήταν η ελαχιστοποίηση του μέσου απόλυτου σφάλματος πρόβλεψης. Μια πρόταση για μελλοντική δουλειά είναι η προσπάθεια ρύθμισης των υπερ-παραμέτρων για βελτιστοποίηση (ελαχιστοποίηση) του χρόνου εκπαίδευσης των μοντέλων (βλ. και Κεφάλαιο 5) ή του χρόνου πρόβλεψης. Ο χρόνος πρόβλεψης είναι ο χρόνος που χρειάζεται το μοντέλο για να εκτελέσει τις προβλέψεις (έχει προηγηθεί η εκπαίδευσή του με το σύνολο εκπαίδευσης). Η μεταβλητή αυτή είναι πολύ σημαντική σε περιπτώσεις που απαιτούνται γρήγορες προβλέψεις, όπως στην περίπτωση εκτέλεσης πολύ βραχυχρόνιων προβλέψεων (σε βάθος μερικών λεπτών ή και δευτερολέπτων στο χρόνο) για την εξισορρόπηση ισχύος στο δίκτυο.

Τέλος, όσον αφορά τα αιολικά πάρκα της Ρόδου, θα μπορούσε να γίνει προσπάθεια εύρεσης περισσότερων ανεμολογικών δεδομένων και μετρήσεων ισχύος, πχ από παλαιότερα έτη ή μελλοντικά από νέες μετρήσεις, ώστε να χρησιμοποιηθούν για την περαιτέρω αξιολόγηση της ικανότητας πρόβλεψης ισχύος των μοντέλων. Επίσης, θα μπορούσε να διερευνηθεί περισσότερο η συμπεριφορά του πάρκου 5 (Atabyrou), για το οποίο δεν προέκυψαν ικανοποιητικά αποτελέσματα στην παρούσα εργασία (βλ. Ενότητα 6.2.5): πιθανώς απαιτείται η συλλογή περισσότερων ανεμολογικών δεδομένων για την σωστή εκπαίδευση ενός μοντέλου για αυτό το πάρκο.



## ΠΑΡΑΡΤΗΜΑ Α - ΚΩΔΙΚΑΣ

---

Παρακάτω παρατίθενται τα βασικά τμήματα του κώδικα που χρησιμοποιήθηκε για την ανάπτυξη των μοντέλων. Παρουσιάζουμε τον κώδικα για το 1<sup>ο</sup> Πάρκο μόνο, καθώς η διαδικασία για τα υπόλοιπα 4 πάρκα είναι παρόμοια. Επίσης, ο κώδικας που παρατίθεται αναφέρεται στο μοντέλο παλινδρόμησης *SVR (Support Vector Regressor)*. Για τα υπόλοιπα μοντέλα η διαδικασία που ακολουθήθηκε είναι αντίστοιχη.

### Προεπεξεργασία Δεδομένων

Τα ανεμολογικά δεδομένα του πάρκου εισάγονται στην Python και απομονώνονται τα δείγματα με Set Point = 1. Από τις στήλες Wind Speed και Wind Direction (βλ. Πίνακα 3.1) κατασκευάζεται το διάνυσμα εισόδου (*X\_data*) και από τη στήλη Power κατασκευάζεται το διάνυσμα εξόδου (*Y\_target*). Στη συνέχεια οι 2 αυτοί πίνακες χωρίζονται σε Training Set και Testing Set με αναλογία 80% / 20% αντίστοιχα. Τέλος, οι είσοδοι κανονικοποιούνται με τη βοήθεια του *MinMaxScaler()* ώστε η τιμή τους να κυμαίνεται πάντα από 0 έως 1.

```
# Numpy arrays and Pandas dataframes are used for data manipulation
import numpy as np
import pandas as pd

# Import the data for the 1st Park ("Aeiforiki") from the .xlsx file
data = pd.read_excel('C:/...../rodos(2017-2018).xlsx', sheet_name='Aeiforiki')

# Extract the samples with Set Point = 1
sub_set = data.loc[data['set point'] == 1]

# Create the input and output arrays by selecting the appropriate columns from the sub_set
X_data = sub_set.iloc[:,1:3].values
Y_target = sub_set.iloc[:,3:4].values

# Split the input and output arrays in Training and Testing sets
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_data, Y_target, test_size=0.2)

# Scale the training and testing sets with MinMaxScaler
from sklearn import preprocessing

# Initialize the transformer
min_max_scalerX = preprocessing.MinMaxScaler()

# The transformer is fitted on the training set, and then transforms it
X_train_minmax = min_max_scalerX.fit_transform(X_train)
# The same transformer is then applied to the testing set
X_test_minmax = min_max_scalerX.transform(X_test)
```

## **Βέλτιστη Ρύθμιση Υπερ-παραμέτρων**

Ορίζονται τα σύνολα των τιμών των υπερ-παραμέτρων που θέλουμε να δοκιμαστούν για να βρεθούν οι βέλτιστες τιμές. Εφαρμόζεται ο αλγόριθμος *GridSearchCV*( ) ο οποίος επιστρέφει το βέλτιστο μοντέλο, με τις βέλτιστες τιμές των υπερ-παραμέτρων από αυτές που δοκιμάστηκαν.

```
# Import the packages for the SVR Regressor and the GridSearchCV algorithm
from sklearn import svm
from sklearn.grid_search import GridSearchCV
```

```
# Specify the grid with the hyper-parameter values that we want to search
parameter_grid = [
    {'kernel': ['linear'],
     'C': np.logspace(-3, 3, 7),
     'epsilon': [0, 0.01, 0.1, 0.5, 1]
    },
    {'kernel': ['rbf'],
     'gamma': np.logspace(-3, 2, 6),
     'C': np.logspace(-3, 3, 7),
     'epsilon': [0, 0.01, 0.1, 0.5, 1]
    }
]
```

```
# Instantiate the Grid Search model
gridSearch_SVR = GridSearchCV(svm.SVR(), parameter_grid, n_jobs=-1, cv=5)
# Train the Grid Search Regressor on the training set
gridSearch_SVR.fit(X_train_minmax, y_train)

# Print the optimal parameter values and the best score
print('Best parameters set:', gridSearch_SVR.best_params_)
print('Best score:', gridSearch_SVR.best_score_)
```

## **Τελική Αξιολόγηση του Βέλτιστου Μοντέλου**

Το βέλτιστο μοντέλο στη συνέχεια εφαρμόζεται στο σύνολο δοκιμής (test set) και εκτελεί πρόβλεψη. Υπολογίζεται το *Μέσο Απόλυτο Σφάλμα* της πρόβλεψης, προκειμένου να αξιολογηθεί το μοντέλο.

```
# Apply the optimal Regressor to the test set, and view the accuracy score
gridSearch_SVR.score(X_test_minmax, y_test)

# Calculate the Mean Absolute Error of the prediction made on the test set
from sklearn.metrics import mean_absolute_error

y_predicted = gridSearch_SVR.predict(X_test_minmax)
print('Mean Absolute Error:', mean_absolute_error(y_test, y_predicted))
```



## Κατασκευή Καμπυλών Εκπαίδευσης

Στο στάδιο αυτό κατασκευάζονται οι καμπύλες εκπαίδευσης καθώς και οι καμπύλες του χρόνου εκπαίδευσης για κάθε υπερ-παράμετρο του μοντέλου. Στις καμπύλες αυτές μπορούμε να παρατηρήσουμε την επίδραση που έχει η κάθε υπερ-παράμετρος στη συμπεριφορά του μοντέλου. Ορίζουμε τη συνάρτηση `plot_results`, η οποία είναι υπεύθυνη για την κατασκευή των καμπυλών, και στη συνέχεια ορίζουμε για κάθε υπερ-παράμετρο ένα εύρος τιμών και κατασκευάζουμε τις καμπύλες με τη βοήθεια αυτής της συνάρτησης.

**# Import the plotting library:**

```
import matplotlib.pyplot as plt
```

**# The optimal model calculated with GridSearchCV:**

```
SVR regr = svm.SVR(kernel='rbf', gamma=50, C=100, epsilon=0.01)
```

**# The function for plotting the results:**

```
def plot_results(_model, _param, _grid):
```

```
    train_scores = []
```

```
    test_scores = []
```

```
    train_MAEscores = []
```

```
    test_MAEscores = []
```

```
    train_time = []
```

**# Extract information from the cross-validation model:**

```
train_scores = _model.cv_results_['mean_train_score']
```

```
test_scores = _model.cv_results_['mean_test_score']
```

```
train_MAEscores = [abs(number) for number in train_scores]
```

```
test_MAEscores = [abs(number) for number in test_scores]
```

```
train_time = _model.cv_results_['mean_fit_time']
```

**# Plot the MAE scores over the hyper-parameter:**

```
fig_scores, graph_scores = plt.subplots()
```

```
graph_scores.semilogx(_grid, train_MAEscores, '-g', label='Train Set')
```

```
graph_scores.semilogx(_grid, test_MAEscores, '-c', label='Test Set')
```

```
graph_scores.set(xlabel=_param, ylabel='Mean Absolute Error',  
                 title='Mean Absolute Error = f(%s)' % _param)
```

```
graph_scores.grid(b=True, which='both')
```

```
graph_scores.minorticks_on()
```

```
graph_scores.legend()
```

```
filename = "C:/Users/.../Desktop/{}_MAE.png".format(_param)
```

```
fig_scores.savefig(filename, format='png', dpi=320, bbox_inches='tight')
```

**# Plot the train time:**

```
fig_trainTime, graph_trainTime = plt.subplots()
```

```
graph_trainTime.semilogx(_grid, train_time, '-r', label='Train Time')
```

```
graph_trainTime.set(xlabel=_param, ylabel='Training Time (sec)',
                    title='Training Time = f(%)' %_param)
graph_trainTime.grid(b=True, which='both')
graph_trainTime.minorticks_on( )
filename = "C:/Users/.../Desktop/{}_trainTime.png".format(_param)
fig_trainTime.savefig(filename, format='png', dpi=320, bbox_inches='tight')
```

- **gamma:**

```
# Define a grid for the gamma parameter:
```

```
gamma_grid = np.concatenate( (np.linspace(0.002, 0.01, 5), np.linspace(0.02, 0.1, 5),
                             np.linspace(0.2, 1, 5), np.linspace(2, 10, 9),
                             np.linspace(20, 100, 9))
                             )
```

```
parameter_grid = {'gamma': gamma_grid}
```

```
# Create the grid search model and fit to the training data:
```

```
SVR_gridSearch = GridSearchCV(SVRegr, param_grid=parameter_grid, return_train_score=True,
                              cv=5, scoring='neg_mean_absolute_error', n_jobs=-1, verbose=2)
```

```
SVR_gridSearch.fit(X_train_minmax, y_train)
```

```
# Dict containing the results:
```

```
gamma_results = SVR_gridSearch.cv_results_
```

```
# Plot Results:
```

```
plot_results(SVR_gridSearch, 'gamma', gamma_grid)
```

- **C:**

```
# Define a grid for the C parameter:
```

```
C_grid = np.concatenate( (np.linspace(0.002, 0.01, 5), np.linspace(0.02, 0.1, 5),
                          np.linspace(0.2, 1, 5), np.linspace(2, 10, 5),
                          np.linspace(20, 100, 9), np.linspace(200, 1000, 9))
                          )
```

```
parameter_grid = {'C': C_grid}
```

```
# Create the grid search model and fit to the training data:
```

```
SVR_gridSearch = GridSearchCV(SVRegr, param_grid=parameter_grid, return_train_score=True,
                              cv=5, scoring='neg_mean_absolute_error', n_jobs=-1, verbose=2)
```

```
SVR_gridSearch.fit(X_train_minmax, y_train)
```

```
# Dict containing the results:
```

```
C_results = SVR_gridSearch.cv_results_
```

```
# Plot Results:
```

```
plot_results(SVR_gridSearch, 'C', C_grid)
```

- **epsilon:**

```
# Define a grid for the epsilon parameter:
```

```
epsilon_grid = np.concatenate( (np.linspace(0.002, 0.01, 9), np.linspace(0.02, 0.1, 9),  
                               np.linspace(0.2, 1, 5), np.linspace(2, 10, 5))  
                               )
```

```
parameter_grid = {'epsilon': epsilon_grid}
```

```
# Create the grid search model and fit to the training data:
```

```
SVR_gridSearch = GridSearchCV(SVRegr, param_grid=parameter_grid, return_train_score=True,  
                               cv=5, scoring='neg_mean_absolute_error', n_jobs=-1, verbose=2)
```

```
SVR_gridSearch.fit(X_train_minmax, y_train)
```

```
# Dict containing the results:
```

```
epsilon_results = SVR_gridSearch.cv_results_
```

```
# Plot Results:
```

```
plot_results(SVR_gridSearch, 'epsilon', epsilon_grid)
```

## Λιεξαγωγή Προβλέψεων στα Αιολικά Πάρκα

Στο στάδιο αυτό χρησιμοποιείται το βέλτιστο μοντέλο SVR για την πρόβλεψη της ετήσιας ισχύος του αιολικού πάρκο, καθώς και της ετήσιας απορριπτόμενης ισχύος. Στον κώδικα που ακολουθεί φαίνεται η διαδικασία για το έτος 2018 (αντίστοιχη είναι και για το 2017).

```
##### Train the optimal model #####
```

```
# The optimal regression model that was selected:
```

```
estimator = svm.SVR(kernel='rbf', gamma=50, C=100, epsilon=0.01)
```

```
# Import the data for the 1st Park ("Aeiforiki") from the .xlsx file
```

```
data = pd.read_excel('C:/Users/.../rodos(2017-2018).xlsx', sheet_name='Aeiforiki')
```

```
# Extract the samples with SetPoint=1 and non-negative power (power<0 means no measurement  
# was made)
```

```
sub_set = data.loc[(data['set point']==1) & (data['power']>=0)]
```

```
# Create the input and output arrays by selecting the appropriate columns from the sub_set
```

```
X_data = sub_set.iloc[:,1:3].values
```

```
Y_target = sub_set.iloc[:,3:4].values
```

```
# Split the 2 arrays in Training and Testing sets and pass them in arrays
```

```
X_train, X_test, y_train, y_test = train_test_split(X_data, Y_target, test_size=0.2)
```

```
y_train = y_train.ravel()
```

```
# Scale the training set with MinMaxScaler
```

```
# Initialize the transformer:
```

```
min_max_scalerX = preprocessing.MinMaxScaler()
```

```
# The transformer is fitted on the training set, and then transforms it:
X_train_minmax = min_max_scalerX.fit_transform(X_train)

# Train the Estimator on the training set
estimator.fit(X_train_minmax, y_train)

##### Predict the power produced by the park #####

# Import the .xlsx file containing only the 2018 data for the park
data_2018 = pd.read_excel('C:/Users/.../rodos2018.xlsx', sheet_name='Aeiforiki')

# Extract the samples with SetPoint!=1 and non-negative power (power<0 means no measurement
# was made)
prediction_subset = data_2018.loc[(data_2018['set point']!=1) & (data_2018['power']>=0)]

# Create the table with the samples for which we will predict the output
prediction_Xdata = prediction_subset.iloc[:,1:3].values
# The same table but also containing the dates:
prediction_Xdata_withDates = prediction_subset.iloc[:,0:3].values
# Create the table with the power used (by the grid) for these samples
power_used = prediction_subset.iloc[:,3:4].values

# Scale the input (features) table with the same transformer as before
prediction_Xdata_minmax = min_max_scalerX.transform(prediction_Xdata)

# Use the trained Estimator to predict the output (the power produced)
predicted_power = estimator.predict(prediction_Xdata_minmax)
# Convert the 1-D array to a 2-D array
predicted_power = predicted_power[:,np.newaxis]

# Combine those measurements in a single array
final_array = np.concatenate( (prediction_Xdata_withDates, predicted_power, power_used), axis=1)

##### Calculate the rejected power #####

# Calculate the rejected power for each sample and store it in a new array
P_rejected_array = np.empty( ((final_array.shape[0]),1), dtype='float64')
P_rejected_array[:,0] = final_array[:,3] - final_array[:,4]
# Replace negative values (prediction<power_used) with 0
P_rejected_array[P_rejected_array < 0] = 0

# Combine this table with the previous analytical table
final_array_with_P_rejected = np.concatenate( (final_array,P_rejected_array), axis=1)
# Export the array in excel:
df2 = pd.DataFrame(final_array_with_P_rejected)
df2.columns = ['Date & Time','Wind Speed','Wind Direction','Predicted Power','Power
              Used','Rejected Power']
filepath = 'C:/Users/.../Aeiforiki2018_AnalyticalTable.xlsx'
df2.to_excel(filepath, index=None)
```

```
##### Calculate the annual sums #####

# Calculate Rejected Power sum
P_rejected_sum = P_rejected_array.sum(axis=0)

# Calculate Power Used By Grid sum
PowerUsed_subset = data_2018.loc[data_2018['power']>=0]
power_used_array = PowerUsed_subset.iloc[:,3:4].values
power_used_sum = power_used_array.sum(axis=0)

# Calculate Power Produced By Park sum
# Calculate sum for the Set Points != 1:
predicted_power_array = final_array_with_P_rejected[:,3]
PredictedPower_sum = predicted_power_array.sum(axis=0)
# Calculate sum for the Set Points == 1 && power>=0 :
PowerSP1_subset = data_2018.loc[(data_2018['set point']==1) & (data_2018['power']>=0)]
power_sp1_array = PowerSP1_subset.iloc[:,3:4].values
power_sp1_sum = power_sp1_array.sum(axis=0)
# Sum them:
power_produced_sum = PredictedPower_sum + power_sp1_sum[0]

# Print the sums
print('Total Power Produced by Park:', power_produced_sum)
print('Total Power Used by Grid:', power_used_sum[0])
print('Total Power rejected:', P_rejected_sum[0])
```

## Κατασκευή Ετήσιου Χρονοδιαγράμματος Μεταβολής της Ισχύος

Στο τελευταίο αυτό στάδιο χρησιμοποιούνται τα στοιχεία των προβλέψεων του προηγούμενου βήματος για την κατασκευή του ετήσιου χρονοδιαγράμματος μεταβολής της παραγόμενης από το πάρκο ισχύος και της απορριπτόμενης ισχύος, για τα έτη 2017 και 2018. Παρουσιάζεται η διαδικασία μόνο για το έτος 2018, καθώς είναι πανομοιότυπη και για το 2017.

```
# Create the dataframe for samples with SetPoint!=1 (those that were predicted)
data_2018_Predicted = pd.read_excel('C:/Users/.../Aeiforiki2018_AnalyticalTable.xlsx')
subset_2018_Predicted = data_2018_Predicted.loc[data_2018_Predicted['Predicted Power'] >= 0]
df_dates = subset_2018_Predicted.iloc[:,0:1]
df_P_produced = subset_2018_Predicted.iloc[:,3:4]
df_P_rejected = subset_2018_Predicted.iloc[:,5:6]
df_2018_Predicted = pd.concat( (df_dates,df_P_produced,df_P_rejected), axis=1)

# Create the dataframe for samples with SetPoint=1 (those used for training)
data_2018_Training = pd.read_excel('C:/Users/.../rodos2018.xlsx', sheet_name='Aeiforiki')
subset_2018_Training = data_2018_Training.loc[ (data_2018_Training['set point']==1) &
                                                (data_2018_Training['power']>=0)]
df_dates = subset_2018_Training.iloc[:,0:1]
df_P_produced = subset_2018_Training.iloc[:,3:4]
df_2018_Training = pd.concat( (df_dates,df_P_produced), axis=1)
```

```
df_2018_Training.insert(loc=2, column='Rejected Power', value=0)
df_2018_Training.columns = ['Date & Time', 'Predicted Power', 'Rejected Power']

# Concatenate the 2 dataframes to get the table for the whole year, then sort it
df_2018 = pd.concat( (df_2018_Predicted,df_2018_Training), axis=0, sort=False)
df_2018_ordered = df_2018.sort_values(by=['Date & Time'])

# Plot the 2018 curves for the Produced Power
fig_P_produced, graph_P_produced= plt.subplots()
graph_P_produced.plot(df_2018_ordered.iloc[:,0], df_2018_ordered.iloc[:,1], '-g', label='2018',
                      linewidth=0.5)
graph_P_produced.set(xlabel='Dates', ylabel='Produced Power (MW)',
                    title='Power Produced by the Park over the year 2018')
graph_P_produced.grid(b=True, which='both')

from matplotlib.dates import MonthLocator, WeekdayLocator, DateFormatter, YearLocator
months = MonthLocator()
monthsFmt = DateFormatter("%b")
years = YearLocator(month=4, day=1)
yrsFmt = DateFormatter("\n%Y")
graph_P_produced.xaxis.set_minor_locator(months)
graph_P_produced.xaxis.set_minor_formatter(monthsFmt)
for tick in graph_P_produced.xaxis.get_minor_ticks():tick.label.set_fontsize(9)
graph_P_produced.xaxis.set_major_locator(years)
graph_P_produced.xaxis.set_major_formatter(yrsFmt)

filename = "C:/Users/.../Park_2018_P_produced.png"
fig_P_produced.savefig(filename, format='png', dpi=350, bbox_inches='tight')

# Plot the 2018 curves for the Rejected Power:
fig_P_produced, graph_P_produced= plt.subplots()
graph_P_produced.plot(df_2018_ordered.iloc[:,0], df_2018_ordered.iloc[:,2], '-c', label='2018',
                      linewidth=0.5)
graph_P_produced.set(xlabel='Dates', ylabel='Rejected Power (MW)',
                    title='Power rejected over the year 2018')
graph_P_produced.grid(b=True, which='both')

from matplotlib.dates import MonthLocator, WeekdayLocator, DateFormatter, YearLocator
months = MonthLocator()
monthsFmt = DateFormatter("%b")
years = YearLocator(month=4, day=1)
yrsFmt = DateFormatter("\n%Y")
graph_P_produced.xaxis.set_minor_locator(months)
graph_P_produced.xaxis.set_minor_formatter(monthsFmt)
for tick in graph_P_produced.xaxis.get_minor_ticks():tick.label.set_fontsize(9)
graph_P_produced.xaxis.set_major_locator(years)
graph_P_produced.xaxis.set_major_formatter(yrsFmt)

filename = "C:/Users/.../Park_2018_P_rejected.png"
fig_P_produced.savefig(filename, format='png', dpi=350, bbox_inches='tight')
```

## ΒΙΒΛΙΟΓΡΑΦΙΑ

---

- [1] R. Kohavi and F. Provost, "Glossary Of Terms," *Machine Learning*, vol. 30, pp. 271-274, 1998.
- [2] A. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210-229, July 1959.
- [3] T. Mitchell, *Machine Learning*, McGraw Hill, 1997.
- [4] S. Theodoridis, *Machine Learning - A Bayesian and Optimization Perspective*, London: Academic Press, 2015.
- [5] P. N. Stuart J. Russell, *Artificial Intelligence: A Modern Approach*, Third Edition, Prentice Hall, 2010.
- [6] X. Zhu, *Semi-supervised learning literature survey*, Computer Sciences, University of Wisconsin-Madison, 2008.
- [7] G. Varoquaux, "Scikit-learn: machine learning in Python," [Online]. Available: <https://scipy-lectures.org/packages/scikit-learn/index.html#supervised-learning-classification-of-handwritten-digits>. [Accessed September 2019].
- [8] The European Wind Energy Association, "Wind in power - 2015 European Statistics," February 2016. [Online]. Available: <https://windeurope.org/wp-content/uploads/files/about-wind/statistics/EWEA-Annual-Statistics-2015.pdf>. [Accessed September 2019].
- [9] Υπουργείο Περιβάλλοντος Ενέργειας και Κλιματικής Αλλαγής, «Ετήσια Έκθεση της Υπηρεσίας Α.Π.Ε. Έτος 2010,» Φεβρουάριος 2011. [Ηλεκτρονικό]. Available: <http://www.ypeka.gr/LinkClick.aspx?fileticket=G3A0b8ZdmNA%3d&tabid=546&language=el-GR>. [Πρόσβαση Σεπτέμβριος 2019].
- [10] J. P. Heinermann, "Wind Power Prediction with Machine Learning Ensembles (Ph.D. Dissertation)," Carl von Ossietzky University of Oldenburg, 21 September 2016. [Online]. Available: <https://d-nb.info/1122481861/34>. [Accessed September 2019].
- [11] A. M. Foley, P. Leahy, A. Marvuglia and E. McKeogh, "Current methods and advances in forecasting of wind power generation," *Renewable Energy*, vol. 37, no. 1, pp. 1-8, 2012.
- [12] N. Treiber, S. Spath, J. Heinermann, L. von Bremen and O. Kramer, "Comparison of Numerical Models and Statistical Learning for Wind Speed Prediction," in *ESANN*, Bruges (Belgium), 2015.
- [13] S. Aggarwal and M. Gupta, "Wind power forecasting: A review of statistical models," *International Journal of Energy Science*, vol. 3, 2013.
- [14] A. Kusiak, H. Zheng and Z. Song, "Short-term prediction of wind farm power: a data mining approach," *IEEE Transactions on Energy Conversion*, vol. 24, no. 1, pp. 125-136, 2009.
- [15] O. Kramer, N. Treiber and F. Gieseke, "Machine Learning in Wind Energy Information Systems," in *EnviroInfo*, 2013.

- [16] S. Salcedo-Sanz, J. Rojo-Alvarez, M. Martinez-Ramon and G. Camps-Valls, "Support vector machines in engineering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 3, pp. 234-267, 2014.
- [17] J. Torres, R. Aguilar and K. Zuniga, "Deep Learning to predict the generation of a wind farm," *Journal of Renewable and Sustainable Energy*, vol. 10, no. 1, 2018.
- [18] P. Ramasamy, S. Chandel and A. Yadav, "Wind speed prediction in the mountainous region of india using an artificial neural network model," *Renewable Energy*, vol. 80, pp. 338-347, 2015.
- [19] D. M. Hawkins, "The problem of overfitting," *Journal of Chemical Information and Modeling*, vol. 44, no. 1, pp. 1-12, 2004.
- [20] Martha K. Smith, "Overfitting," 13 June 2014. [Online]. Available: <https://web.ma.utexas.edu/users/mks/statmistakes/ovefitting.html>. [Accessed September 2019].
- [21] Pedregosa, Varoquaux, Gramfort et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.
- [22] Scikit-learn, "Cross-validation: evaluating estimator performance - scikit-learn 0.21.3 documentation," [Online]. Available: [https://scikit-learn.org/stable/modules/cross\\_validation.html](https://scikit-learn.org/stable/modules/cross_validation.html). [Accessed September 2019].
- [23] Scikit-learn, "Preprocessing data - scikit-learn 0.21.3 documentation," [Online]. Available: <https://scikit-learn.org/stable/modules/preprocessing.html#scaling-features-to-a-range>. [Accessed September 2019].
- [24] M. Claesen and Bart De Moor, "Hyperparameter Search in Machine Learning," in *MIC 2015: The XI Metaheuristics International Conference*, Belgium, 2015.
- [25] G. C. Cawley and N. L. Talbot, "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation," *Journal of Machine Learning Research*, vol. 11, pp. 2079-2107, 2010.
- [26] Scikit-learn, "Tuning the hyper-parameters of an estimator - scikit-learn 0.21.3 documentation," [Online]. Available: [https://scikit-learn.org/stable/modules/grid\\_search.html](https://scikit-learn.org/stable/modules/grid_search.html). [Accessed September 2019].
- [27] Scikit-learn, "Model evaluation: quantifying the quality of predictions - scikit-learn 0.21.3 documentation," [Online]. Available: [https://scikit-learn.org/stable/modules/model\\_evaluation.html](https://scikit-learn.org/stable/modules/model_evaluation.html). [Accessed September 2019].
- [28] C. J. Willmot and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Research*, vol. 30, no. 1, pp. 79-82, 2005.
- [29] C. J. Willmot, "On the use of dimensioned measures of error to evaluate the performance of spatial interpolators," *International Journal of Geographical Information Science*, vol. 20, no. 1, pp. 89-102, 2006.
- [30] R. Hyndman and A. Koehler, "Another look at measures of forecast accuracy," November 2005. [Online]. Available:



- <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.154.9771&rep=rep1&type=pdf>. [Accessed September 2019].
- [31] A. Colin Cameron and F. A. Windmeijer, "An R-squared measure of goodness of fit for some common nonlinear regression models," *Journal of Econometrics*, vol. 77, no. 2, pp. 329-342, 1997.
- [32] Scikit-learn, "Supervised learning - scikit-learn 0.21.3 documentation," [Online]. Available: [https://scikit-learn.org/stable/supervised\\_learning.html#supervised-learning](https://scikit-learn.org/stable/supervised_learning.html#supervised-learning). [Accessed September 2019].
- [33] B. F. Swindel, "Geometry of Ridge Regression Illustrated," *The American Statistician*, vol. 35, no. 1, pp. 12-15, 1981.
- [34] A. Y. Ng, "Feature selection, L1 vs. L2 regularization, and rotational invariance," in *21st International Conference on Machine Learning*, Banff, Canada, 2004.
- [35] Anuja Nagpal, "L1 and L2 Regularization Methods," October 2017. [Online]. Available: <https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c>. [Accessed September 2019].
- [36] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B*, vol. 58, no. 1, pp. 267-288, 1996.
- [37] H. Zou and T. Hastie, "Regularization and variable selection via the Elastic Net," *Journal of the Royal Statistical Society, Series B*, vol. 67, pp. 301-320, 2005.
- [38] M.E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *Journal of Machine Learning Research*, vol. 1, pp. 211-244, 2001.
- [39] Jason Brownlee, "Gradient Descent For Machine Learning," March 2016. [Online]. Available: <https://machinelearningmastery.com/gradient-descent-for-machine-learning/>. [Accessed September 2019].
- [40] Lili Jiang, "What are kernels in machine learning and SVM and why do we need them?," October 2016. [Online]. Available: <https://www.quora.com/What-are-kernels-in-machine-learning-and-SVM-and-why-do-we-need-them/answer/Lili-Jiang?srid=oOgT>. [Accessed September 2019].
- [41] C. Cortes and V. N. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, September 1995.
- [42] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, New York: Springer, 2008.
- [43] H. Drucker, C. C. Burges, L. Kaufman, A. J. Smola and V. N. Vapnik, "Support Vector Regression Machines," in *Advances in Neural Information Processing Systems 9*, NIPS, 1996.
- [44] A. J. Smola and B. Scholkopf, "A Tutorial on Support Vector Regression," *Statistics And Computing*, 30 September 2003.
- [45] N. Altman, "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression," *The American Statistician*, vol. 46, no. 3, pp. 175-185, 1992.

- [46] J. L. Bentley, A survey of techniques for fixed-radius near neighbor searching, Stanford, California: Stanford Linear Accelerator Center, 1975.
- [47] R. Samworth, "Optimal weighted nearest neighbour classifiers," *Annals of Statistics*, vol. 40, no. 5, p. 2733–2763, 2012.
- [48] J. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509-517, 1975.
- [49] S. Omohundro, "Five Balltree Construction Algorithms," *International Computer Science Institute Technical Report*, 1989.
- [50] J. Quinlan, "Simplifying decision trees," *International Journal of Man-Machine Studies*, vol. 27, no. 3, p. 221–234, 1987.
- [51] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21-45, 2006.
- [52] T. K. Ho, "Random Decision Forests," in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, 1995.
- [53] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.
- [54] P. Geurts, D. Ernst and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3-42, 2006.
- [55] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119-139, 1997.
- [56] H. Drucker, "Improving Regressors using Boosting Techniques," in *ICML '97 Proceedings of the Fourteenth International Conference on Machine Learning*, 1997.
- [57] L. Mason, J. Baxter, P. Bartlett and M. Frean, "Boosting Algorithms as Gradient Descent," in *Advances in Neural Information Processing Systems 12*, MIT Press. pp. 512–518, 1999.
- [58] J. Friedman, "Stochastic Gradient Boosting," March 1999. [Online]. Available: <https://statweb.stanford.edu/~jhf/ftp/stobst.pdf>. [Accessed September 2019].
- [59] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, p. 533–536, 1986.
- [60] Y. LeCun, L. Bottou, G. Orr and K.-R. Muller, "Efficient BackProp," in *Neural Networks: tricks of the trade*, Springer, 1998.
- [61] D. P. Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, 2014.
- [62] T. Lim, W. Loh and Y. Shih, "A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms," *Machine Learning*, vol. 40, no. 3, p. 203–228, 2000.
- [63] A. Klein, S. Falkner, J. T. Springenberg and F. Hutter, "Learning Curve Prediction with Bayesian Neural Networks," in *International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

- [64] T. Domhan, J. Springenberg και F. Hutter, «Speeding Up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves,» σε *International Joint Conference on Artificial Intelligence*, North America, 2015.
- [65] J. Brownlee, "How to use Learning Curves to Diagnose Machine Learning Model Performance," February 2019. [Online]. Available: <https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/>. [Accessed September 2019].
- [66] Scikit-learn, "Kernel functions: Custom Kernels - scikit-learn 0.21.3 documentation," [Online]. Available: <https://scikit-learn.org/stable/modules/svm.html#custom-kernels>. [Accessed September 2019].





**«Πρόβλεψη της Απορριπτόμενης Ισχύος των Αιολικών Πάρκων της Ρόδου με Μεθόδους  
Μηχανικής Μάθησης»**

του  
**Ιωάννη Χανή**

**2019**