ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

Τμήμα Μηχανικών Βιομηχανικής Σχεδίασης & Παραγωγής

ΤΊΤΛΟΣ:

Πρακτική εφαρμογή Μηχανικής μάθησης και Ανάλυσης σε δεδομένα ηλεκτρικής ενέργειας.

Επιβλέπων καθηγητής: Νικολάου Γρηγόρης
Φοιτητής: Νούλιας Αντρέας
ΑΜ:46446

ΑΘΉΝΑ 2021

Η Διπλωματική/Πτυχιακή Εργασία εξετάστηκε από την κάτωθι Εξεταστική Επιτροπή:

ΟΝΟΜΑ ΕΠΩΝΥΜΟ	ΒΑΘΜΙΔΑ	ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ
ΝΙΚΟΛΑΟΥ Γ.	ΛΕΚΤΟΡΑΣ	GRIGORIOS NIKOLAOU Digitally signed by GRIGORIOS NIKOLAOU Date: 2022.04.11 10:46:06 +03'00'
ΒΑΣΙΛΕΙΑΔΟΥ Σ.	ΕΠΙΚΟΥΡΗ ΚΑΘΗΓΗΤΡΙΑ	Soultana Vasileiadou Digitally signed by Soultana Vasileiadou Date: 2022.04.10 20:36:36 +03'00'
ΔΡΟΣΟΣ Χ.	ΕΔΙΠ	Christos Drosos Digitally signed by Christos Drosos Date: 2022.04.05 18:42:28 +03'00'

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/η κάτωθι υπογεγραμμένος/η Ανδρέας Νούλιας του Αριστοκλή, με αριθμό μητρώου 46446 φοιτητής/τρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Βιομηχανικής Σχεδίασης και Παραγωγής, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο/Η Δηλών/ούσα

Ανδρέας Νούλιας

Abstract \Περίληψη

Η εργασία αυτή έχει εκπονηθεί από τον Ανδρέα Νούλια για το Πανεπιστήμιο Δυτικής Αττικής και ο τίτλος είναι 'πρακτική εφαρμογή machine learning και analytics σε δεδομένα ηλεκτρικής ενέργειας'. Ο σκοπός τις εργασίας είναι να εξετάσουμε με την χρήση Μηχανικής μάθησης δεδομένα που αφορούν κατανάλωση ηλεκτρικού ρεύματος. Τα δεδομένα είναι δομημένα με time series, και σκοπός μας είναι με ένα dataset το οποίο κατεβάσαμε από το UCI και σχετίζεται όπως είπαμε με ηλεκτρική ενέργεια.

Σκοπός μας είναι να κάνουμε ανάλυση των δεδομένων σε πρώτη φάση έτσι ώστε να επιλέξουμε μοντέλα μηχανικής μάθησης που ταιριάζουν για το πρόβλημα μας και να κάνουμε forecast πάνω σε αυτά έτσι ώστε να δημιουργήσουμε ένα μοντέλο που θα μας κάνει πρόβλεψη της ηλεκτρικής ενέργειας. Το project μας θα υλοποιηθεί με την χρήση τελευταίων τεχνολογιών.

Για την επεξεργασία και την χρήση μηχανικής μάθησης θα χρησιμοποιήσουμε την γλώσσα προγραμματισμού Python και για την υλοποίηση του Project μας θα χρησιμοποιήσουμε το github έτσι ώστε να κρατήσουμε τον κώδικα και τα δεδομένα σε ένα project. Θα αναλύσουμε έννοιες όπως τι είναι η μηχανική μάθηση, χρονοσειρές και ανάλυση δεδομένων και σκοπός μας είναι εκτός από τα δεδομένα που έχούμε να βρούμε άλλες εξαρτώμενες-α δεδομένα\ μεταβλητές όπου θα μας βοηθήσουν να βγάλουμε καλύτερα συμπεράσματα για τα δεδομένα μας αλλά και καλύτερα αποτελέσματα στα μοντέλα μηχανικής μάθησης.

Η εργασία θα δομηθεί σε δύο κομμάτια το ένα θα είναι η θεωρικά μελέτη και η ανάλυση ορισμών και το δεύτερο κομμάτι θα είναι η πρακτική εφαρμογή και ότι έχουμε διαβάσει και έχουμε κάνει έρευνα να προσπαθήσουμε να τα υλοποιήσουμε και να βγάλουμε τα αποτελέσματα μετά από την μελέτη και εφαρμογή τεχνικών και μοντέλων μηχανικής μάθησης.

ΕΥΧΑΡΙΣΤΙΕΣ

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω μια σειρά από ανθρώπους που βοήθησαν για να πραγματοποιηθεί η εργασία αυτή. Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα, καθηγητή κ. Νικολάου Γρηγόρης για την ανάθεση της εργασίας αυτής σε εμένα, καθώς επίσης και για την ευχέρεια που μου έδωσε ώστε να χειριστώ το αχανές αυτό αντικείμενο και να το δομήσω σύμφωνα με τις επιθυμίες μου. Επίσης θα ήθελα να ευχαριστήσω και όλους τους υπόλοιπους καθηγητές του τμήματος μηχανικών βιομηχανικής σχεδίασης και παραγωγής για τις γνώσεις που αποκόμισα ώστε να είναι εφικτό πλέον να εφαρμόσω τις γνώσεις αυτές στην πράξη και να ενταχθώ στον κλάδο σχεδίαση σύγχρονων συστημάτων και υπηρεσιών ακολουθώντας τις βέλτιστες προσεγγίσεις στο διεπιστημονικό χώρο της σχεδίασης.Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου που με στηρίζει όλα αυτά τα χρόνια στην ακαδημαϊκή μου πορεία και μου έδωσε την δυνατότητα, το ήθος και τα εφόδια να φτάσω ως εδώ.

Contents

Abstract \Περίληψη4

ΕΥΧΑΡΙΣΤΙΕΣ5

Contents6

Α.Θεωρητικά για το project10

1. ορισμός χρονοσειρών10

2. Κατανόηση χρονοσειρών10

2.1 Τάση (Trend), Εποχικότητα (Seasonality), Θόρυβος(Noise)11

3. Μηχανική μάθηση15

4.Οι χρονοσειρές στην Μηχανική μάθηση16

5.time series forecasting, Forecasting and Modeling16

5.1Εισαγωγή στην Ανάλυση χρονοσειρών16

5.2 Η πρόβλεψη χρονοσειρών17

5.3 Χρονοσειρά και Στοχαστική Διαδικασία17

6.Μοντέλα που θα χρησιμοποιήσουμε18

6.1Artificial Neural Networks (ANNs)18

6.1.2 Αρχιτεκτονική ANN18

6.2 fb prophet20

6.3 Linear Regression21

7. Μέτρηση ακρίβειας προβλέψεων22

Β.Πρακτικό κομμάτι23

1.Ανάλυση δεδομένων23

2.RNN model29

2.1 Τι είναι το lstm;29

2.2 Τι είναι το Lag31

2.3 Πρακτικη εφαρμογή31

2.3.3 Συμπεράσματα για LTSM50

Το seasonality μας θα πρέπει να κάνουμε πιο εκτενέστερη και σε βάθος ανάλυση.50

3.Fbprophet50

3.1 Γιατί το Facebook Prophet;50

3.2 Fbprophet per day51

3.3 Fbprophet per month53

3.4 Fbprophet per year54

3.5 Fbprophet seasonality56

3.6 Fbprophete raw data58

3.7 Συμπεράσματα59

4.WeKa60

4.1Τι είναι το WeKa?60

4.2Εξέταση των δεδομένων μας με το Weka60

4.3Συμπέρασμα σχετικά με το Weka65

Γ. Συμπεράσματα66

Βιβλιογραφία67

Πίνακας Εικόνων

Εικονα 1:Εποχικοτητα και ταση Πηγη https://towardsdatascience.com/trend-seasonality-moving-average-auto-regressive-model-my-journey-to-time-series-data-with-edc4c0c8284b 13
Εικονα 2:Παρουσιαση εποχικοτητας, τασης σε δεδομενα stock market Πηγη: https://towardsdatascience.com/trend-seasonality-moving-average-auto-regressive-model-my-journey-to-time-series-data-with-edc4c0c8284b 14
Εικονα 3: Ενδεικτικος κωδικας Python για την εποχικοτητα (αποτελεσμα εικονα 4) Πηγη: https://towardsdatascience.com/trend-seasonality-moving-average-auto-regressive-model-my-journey-to-time-series-data-with-edc4c0c8284b 15
Εικονα 4: Αρχιτεκτονικη τροφοδοσιας τριων επιπεδων μοντελου ANN 20
Εικόνα 5: Forecasts accuracy measuring table 24
Εικονα 6: Screenshot με τα Δεδομενα 26
Εικονα 7: Δεδομενα ανα μηνα για το 2008 27
Εικονα 8: Δεδομενα ανα χρονο 27
Εικονα 9: Δεδομενα για το 2007 28
Εικονα 10: Δεδομενα για το Ιανουαριο του 2007 28
Εικονα 11: Ιστογραμμα για καθε μεταβλητη 29
Εικονα 12: Ιστογραμμα για καθε μεταβλητη ανα ετος 29
Εικονα 13: Αρχιτεκτονικη 31
Εικονα 14: lstm διαγραμμα train results 34
Εικονα 15:lstm διαγραμμα test results 34
Εικονα 16:lstm διαγραμμα train results 37
Εικονα 17:lstm διαγραμμα train results 38
Εικονα 18:lstm διαγραμμα train results 40
Εικονα 19: lstm διαγραμμα train results 41
Εικονα 20:lstm διαγραμμα test results 42
Εικονα 21:lstm διαγραμμα train results 43
Εικονα 22:lstm διαγραμμα test results 44
Εικονα 23:lstm διαγραμμα train results 45
Εικονα 24:lstm διαγραμμα test results 46
Εικονα 25:lstm διαγραμμα train results 47
Εικονα 26:lstm διαγραμμα test results 48
Εικονα 27:lstm διαγραμμα train results 49
Εικονα 28:lstm διαγραμμα train results 50
Εικονα 29:lstm διαγραμμα test results 51
Εικόνα 30: Data plot diagram (days) 53
Εικόνα 31: Data plot diagram (months) 54
Εικόνα 32:Data plot diagram (years) 56
Εικόνα 33: Data plot diagram (years-seasonality) 58
Εικόνα 34: Raw data diagram 60
Εικόνα 35:Weka interface 61
Εικόνα 36: Models installation 62
Εικόνα 37: data insert weka 62
Εικόνα 38: Model results 63
Εικόνα 39: Weka model results 63
Εικόνα 40: Train Forecasting results 64

Εικόνα 41: Test Forecasting results 64
Εικόνα 42:Weka model results 65
Εικόνα 43: Weka model configuration 65
Εικονα 44: weka model plot 66

Α.Θεωρητικά για το project

1. ορισμός χρονοσειρών

Τι είναι μια χρονοσειρά;

Μια χρονοσειρά είναι μια ακολουθία σημείων δεδομένων που εμφανίζονται σε διαδοχική σειρά για κάποιο χρονικό διάστημα. Αυτό μπορεί να αντιπαραβληθεί με δεδομένα διατομής, τα οποία καταγράφουν ένα χρονικό σημείο.

Στον τομέα των επενδύσεων, μια χρονοσειρά παρακολουθεί την κίνηση των επιλεγμένων σημείων δεδομένων, όπως η τιμή μιας ασφάλειας, σε συγκεκριμένο χρονικό διάστημα με τα σημεία δεδομένων να καταγράφονται σε τακτά χρονικά διαστήματα. Δεν υπάρχει ελάχιστος ή μέγιστος χρόνος που πρέπει να συμπεριληφθεί, επιτρέποντας τη συλλογή των δεδομένων με τρόπο που παρέχει τις πληροφορίες που αναζητούνται από τον επενδυτή ή τον αναλυτή που εξετάζει τη δραστηριότητα.

Στη πτυχιακή μας θα προσπαθήσουμε να εξετάσουμε ένα dataset το οποίο είναι βασισμένο σε καταγραφή τάσης και έντασης ρεύματος με βάση τον χρόνο άρα είναι δεδομένα με βάση χρονοσειρές

2. Κατανόηση χρονοσειρών

Μια χρονοσειρά μπορεί να ληφθεί για οποιαδήποτε μεταβλητή που αλλάζει με την πάροδο του χρόνου. Στον τομέα των επενδύσεων, είναι συνηθισμένο να χρησιμοποιείτε μια χρονοσειρά για να παρακολουθείτε την τιμή ενός τίτλου με την πάροδο του χρόνου. Αυτό μπορεί να παρακολουθείται βραχυπρόθεσμα, όπως η τιμή ενός χρεογράφου την ώρα κατά τη διάρκεια μιας εργάσιμης ημέρας ή μακροπρόθεσμα, όπως η τιμή ενός χρεογράφου που κλείνει την τελευταία ημέρα κάθε μήνα κατά τη διάρκεια της πορεία πέντε ετών.

Η ανάλυση χρονοσειρών μπορεί να είναι χρήσιμη για να δείτε πώς αλλάζει ένα δεδομένο περιουσιακό στοιχείο, ασφάλεια ή οικονομική μεταβλητή με την πάροδο του χρόνου. Μπορεί επίσης να χρησιμοποιηθεί για να εξετάσει πώς οι αλλαγές που σχετίζονται με το επιλεγμένο σημείο δεδομένων συγκρίνονται με τις μεταβολές άλλων μεταβλητών κατά την ίδια χρονική περίοδο

Ακόμα μπορούμε να εξετάσουμε την περιοδικότητα σε εποχές, δεκαετίες , εξάμηνα κτλ. Το χρονικό πεδίο στο οποίο θα εξετάσουμε τα δεδομένα μας σχετίζεται με το πρόβλημα που κάνουμε ανάλυση και θέλουμε να βρούμε απαντήσεις πάνω σε αυτό. Αρα το βασικά μας ερώτημα και κατι όπου αναγνωρίσαμε κατά τις διάρκεια τις μελέτης μας στις χρονοσειρές είναι πώς καθορίζουμε τα χρονικα διαστήματα που θα εξετάσουμε τα δεδομένα μας και ποιος είναι ο σωστός τρόπος επιλογής του.

2.1 Τάση (Trend), Εποχικότητα (Seasonality), Θόρυβος(Noise)

Πριν προχωρήσουμε, πρέπει να συζητήσουμε κάτι σημαντικό, τα περισσότερα δεδομένα χρονοσειρών μπορούν να περιγραφούν από τρία στοιχεία. Και αυτά είναι η τάση, η εποχικότητα και ο θόρυβος.

Τάση → μια γενική συστηματική γραμμική ή (τις περισσότερες φορές) μη γραμμική συνιστώσα που αλλάζει με την πάροδο του χρόνου και δεν επαναλαμβάνεται

Εποχικότητα → μια γενική συστηματική γραμμική ή (τις περισσότερες φορές) μη γραμμική συνιστώσα που αλλάζει με την πάροδο του χρόνου και επαναλαμβάνεται

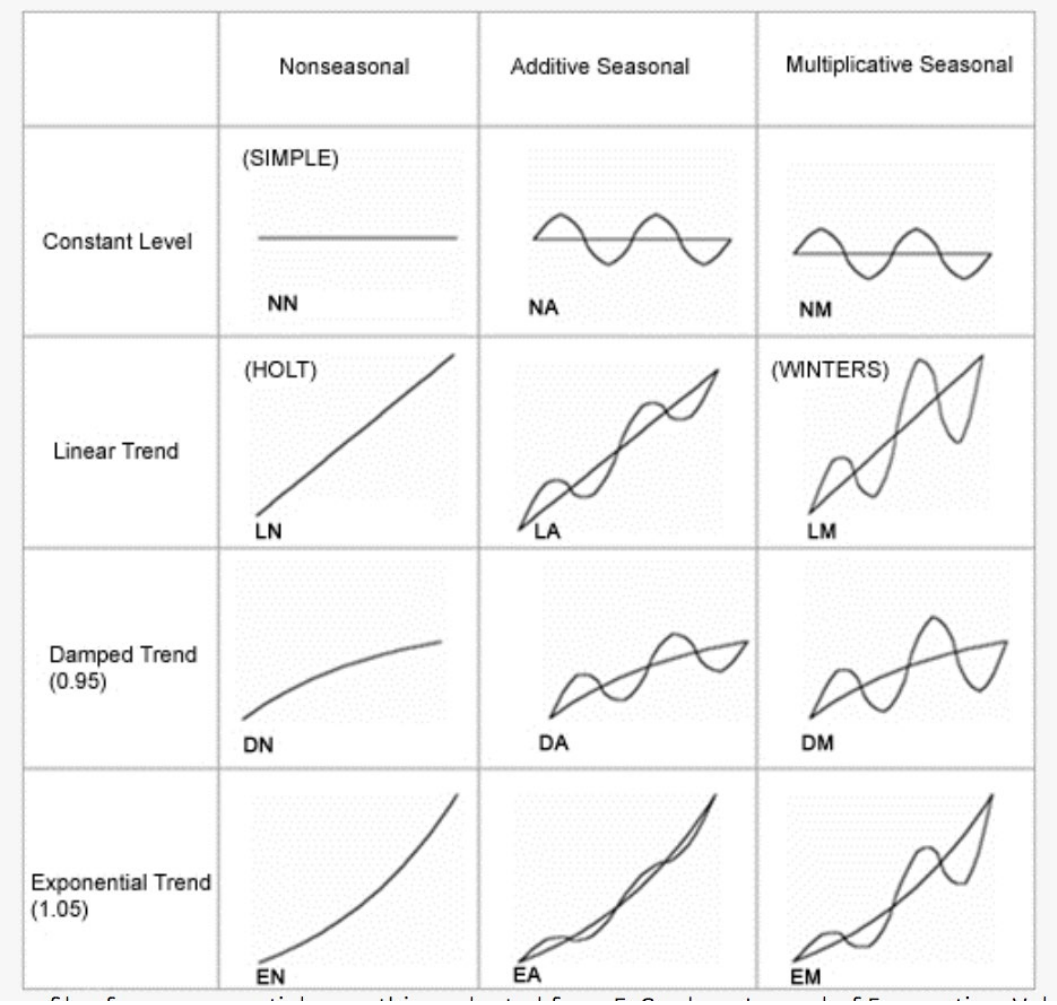
Θόρυβος → ένα μη συστηματικό στοιχείο που δεν είναι ούτε Τάση ούτε εποχικότητα Εποχικότητα στα δεδομένα

Η εποχικότητα υπάρχει όταν μια σειρά επηρεάζεται από περιοδικούς παράγοντες (π.χ. το τρίμηνο του έτους, ο μήνας ή η ημέρα της εβδομάδας). Η εποχικότητα αφορά πάντα μια σταθερή και γνωστή περίοδο. Ως εκ τούτου, χρονοσειρές που εμφανίζουν εποχικότητα ονομάζονται περιοδικές χρονοσειρές.

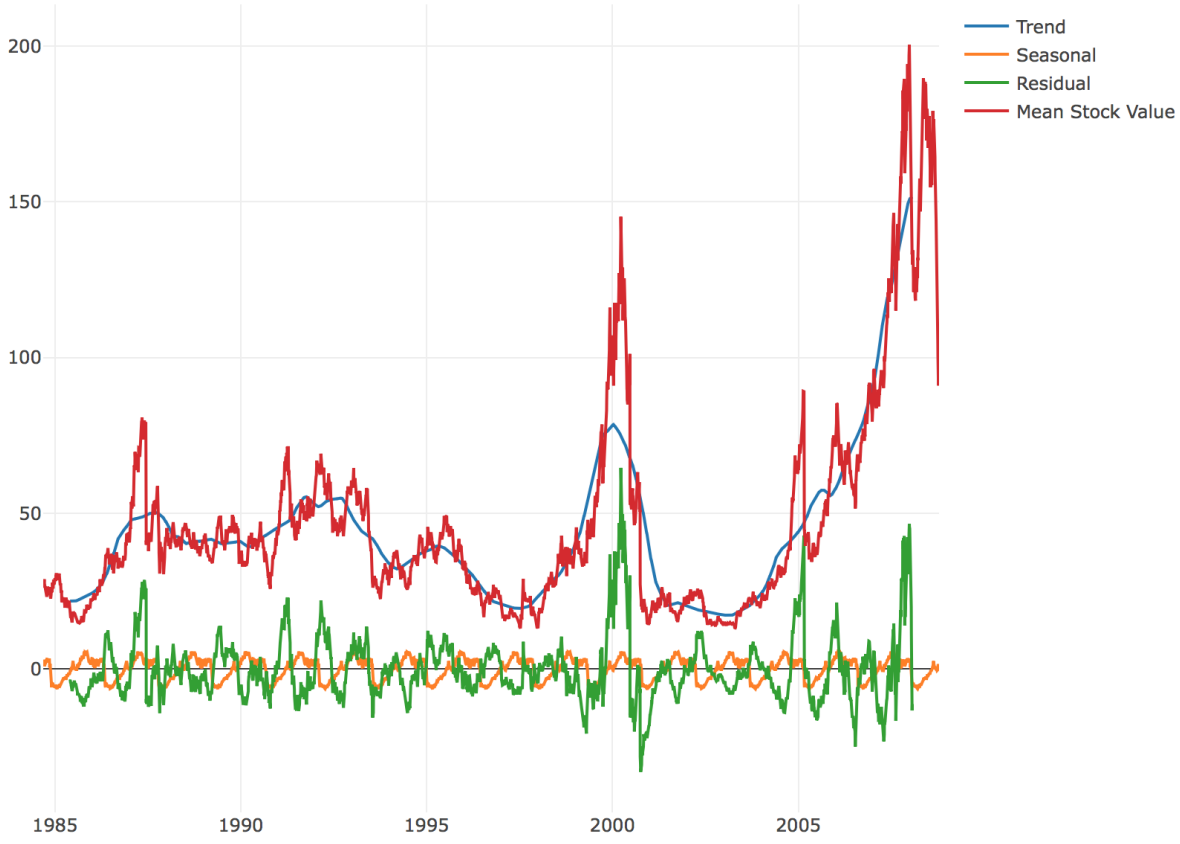
Το κυκλικό μοτίβο αφορά δεδομένα που παρουσιάζουν αυξήσεις και πτώσεις που δεν πραγματοποιούνται σε σταθερές περιόδους. Η διάρκεια αυτών των διακυμάνσεων είναι συνήθως τουλάχιστον 2 χρόνια. Σκεφτείτε οικονομικούς κύκλους που συνήθως διαρκούν αρκετά χρόνια, αλλά όπου η διάρκεια του τρέχοντος κύκλου είναι άγνωστη εκ των προτέρων.

Πολλοί άνθρωποι μπερδεύουν την κυκλική συμπεριφορά με την εποχιακή συμπεριφορά, αλλά είναι πραγματικά πολύ διαφορετικές. Αν οι διακυμάνσεις δεν είναι σταθερής περιόδου τότε είναι κυκλικές. εάν η περίοδος είναι αμετάβλητη και σχετίζεται με κάποιο περιοδικό μοτίβο, τότε το μοτίβο είναι εποχιακό. Γενικά, η μέση διάρκεια των κύκλων είναι μεγαλύτερη από τη διάρκεια ενός εποχιακού μοτίβου και το μέγεθος των κύκλων τείνει να είναι πιο μεταβλητό από το μέγεθος των

εποχιακών προτύπων.



Εικόνα 1:Εποχικότητα και τάση Πηγή https://towardsdatascience.com/trend-seasonality-moving-average-auto-regressive-model-my-journey-to-time-series-data-with-edc4c0c8284b



Εικόνα 2:Παρουσίαση εποχικότητας, τάσης σε δεδομένα stock market Πηγή: https://towardsdatascience.com/trend-seasonality-moving-average-auto-regressive-model-my-journey-to-time-series-data-with-edc4c0c8284b

Ο προσδιορισμός της εποχικότητας στα δεδομένα χρονοσειρών είναι σημαντικός για την ανάπτυξη ενός χρήσιμου μοντέλου χρονοσειρών. Υπάρχουν πολλά εργαλεία που είναι χρήσιμα για τον εντοπισμό της εποχικότητας στα δεδομένα χρονοσειρών.Η θεωρία του υποβάθρου και η γνώση των δεδομένων μπορούν να παρέχουν μια εικόνα για την παρουσία και τη συχνότητα της εποχικότητας.Οι γραφικές παραστάσεις χρονοσειρών όπως η γραφική παράσταση εποχιακών υποσειρών, η γραφική παράσταση αυτοσυσχέτισης ή μια φασματική γραφική παράσταση μπορούν να βοηθήσουν στον εντοπισμό προφανών εποχιακών τάσεων στα δεδομένα.Στατιστική ανάλυση και δοκιμές, όπως η συνάρτηση αυτοσυσχέτισης, τα περιοδογραφήματα ή τα φάσματα ισχύος μπορούν να χρησιμοποιηθούν για τον προσδιορισμό της παρουσίας εποχικότητας. Η ανίχνευση της αυτοσυσχέτισης σε δεδομένα χρονοσειρών μπορεί να γίνει με διάφορους τρόπους. Ένα προκαταρκτικό μέτρο για την ανίχνευση της αυτοσυσχέτισης είναι ένα γράφημα χρονοσειρών υπολειμμάτων έναντι χρόνου. Εάν δεν υπάρχει αυτοσυσχέτιση, τα υπολείμματα θα πρέπει να εμφανίζονται τυχαία και διάσπαρτα γύρω στο μηδέν. Εάν υπάρχει ένα μοτίβο στα υπάρχοντα υπολείμματα, τότε είναι πιθανή η αυτοσυσχέτιση.

```
from statsmodels.tsa.seasonal import seasonal_decompose
decomposition = seasonal_decompose(df.Mean, freq=365)
trace1 = go.Scatter(
    x = df.Date,y = decomposition.trend,
    name = 'Trend',mode='line'
)
trace2 = go.Scatter(
    x = df.Date,y = decomposition.seasonal,
    name = 'Seasonal',mode='line'
)
trace3 = go.Scatter(
    x = df.Date,y = decomposition.resid,
    name = 'Residual',mode='line'
)
trace4 = go.Scatter(
    x = df.Date,y = df.Mean,
    name = 'Mean Stock Value',mode='line'
)
data = [trace1,trace2,trace3,trace4]
plot(data)
```

Εικόνα 3: Ενδεικτικός κώδικας Python για την εποχικότητα (αποτέλεσμα εικόνα 4) Πηγή: https://towardsdatascience.com/trend-seasonality-moving-average-auto-regressive-model-my-journey-to-time-series-data-with-edc4c0c8284b

3. Μηχανική μάθηση

Η Μηχανική μάθηση είναι υποπεδίο της επιστήμης των υπολογιστών, που αναπτύχθηκε από τη μελέτη της αναγνώρισης προτύπων και της υπολογιστικής θεωρίας μάθησης στην τεχνητή νοημοσύνη. Το 1959, ο Άρθουρ Σάμουελ ορίζει τη μηχανική μάθηση ως "Πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μάθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί". Η μηχανική μάθηση διερευνά τη μελέτη και την κατασκευή αλγορίθμων που μπορούν να μάθαίνουν από τα δεδομένα και να κάνουν προβλέψεις σχετικά με αυτά. Τέτοιοι αλγόριθμοι λειτουργούν κατασκευάζοντας μοντέλα από πειραματικά δεδομένα, προκειμένου να κάνουν προβλέψεις βασιζόμενες στα δεδομένα ή να εξάγουν αποφάσεις που εκφράζονται ως το αποτέλεσμα.

Η μηχανική μάθηση είναι στενά συνδεδεμένη και συχνά συγχέεται με υπολογιστική στατιστική, ένας κλάδος, που επίσης επικεντρώνεται στην πρόβλεψη μέσω της χρήσης των υπολογιστών. Έχει ισχυρούς δεσμούς με την μαθηματική βελτιστοποίηση, η οποία παρέχει μεθόδους, τη θεωρία και τομείς εφαρμογής. Η Μηχανική μάθηση εφαρμόζεται σε μια σειρά από υπολογιστικές εργασίες, όπου τόσο ο σχεδιασμός όσο και ο ρητός προγραμματισμός των αλγορίθμων είναι ανέφικτος. Παραδείγματα εφαρμογών αποτελούν τα φίλτρα spam (spam filtering), η οπτική αναγνώριση χαρακτήρων (OCR),οι μηχανές αναζήτησης και η υπολογιστική όραση. Η Μηχανική μάθηση μερικές φορές συγχέεται με την εξόρυξη δεδομένων,όπου η τελευταία επικεντρώνεται περισσότερο στην εξερευνητική ανάλυση των δεδομένων, γνωστή και ως μη επιτηρούμενη μάθηση.

Στο πεδίο της ανάλυσης δεδομένων, η μηχανική μάθηση είναι μια μέθοδος που χρησιμοποιείται για την επινόηση πολύπλοκων μοντέλων και αλγορίθμων που οδηγούν στην πρόβλεψη. Τα αναλυτικά μοντέλα επιτρέπουν στους ερευνητές, τους επιστήμονες δεδομένων, τους μηχανικούς και τους αναλυτές να παράγουν αξιόπιστες αποφάσεις και αποτελέσματα και να αναδείξουν αλληλοσυσχετίσεις μέσω της μάθησης από ιστορικές σχέσεις και τάσεις στα δεδομένα.

Η Μάθηση (Learning) είναι μία από τις θεμελιώδεις ιδιότητες της νοήμονος συμπεριφοράς του ανθρώπου. Παρά τις μελέτες και τις έρευνες επί χρόνια από τους επιστήμονες του πεδίου της Γνωστικής Ψυχολογίας και τους φιλοσόφους, η έννοια της μάθησης δεν έχει γίνει πλήρως κατανοητή. Πώς, λοιπόν, θα μπορούσαν οι επιστήμονες του χώρου της ΤΝ να δημιουργήσουν υπολογιστικά συστήματα ικανά να μάθουν, να επιτύχουν, δηλαδή, τη λεγόμενη Μηχανική Μάθηση (Machine Learning). Αυτή μπορεί να οριστεί ως: το φαινόμενο κατά το οποίο ένα σύστημα βελτιώνει την απόδοσή του κατά την εκτέλεση μιας συγκεκριμένης εργασίας, χωρίς να υπάρχει ανάγκη να προγραμματιστεί εκ νέου. Βάσει του ορισμού αυτού, η Μηχανική Μάθηση έχει ως σκοπό τη δημιουργία μηχανών ικανών να μαθαίνουν, να βελτιώνουν, δηλαδή, την απόδοσή τους σε κάποιους τομείς μέσω της αξιοποίησης προηγούμενης γνώσης και εμπειρίας. Ένας σχετικός γενικός ορισμός Μηχανικής Μάθησης δίνεται από τον Mitchell (1997): «Ένα πρόγραμμα υπολογιστή λέμε ότι μαθαίνει από την εμπειρία Ε ως προς κάποια κλάση εργασιών Τ και μέτρο απόδοσης Ρ, αν η απόδοσή του σε εργασίες από το Τ, όπως μετριέται από το Ρ, βελτιώνεται μέσω της εμπειρίας Ε.»

Ορισμός: Ο πιο ολοκληρωμένος *ορισμός* τις μηχανικής μάθησης που βρήκαμε είναι:

«Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από εμπειρία Ε ως προς μια κλάση εργασιών Τ και ένα μέτρο επίδοσης Ρ, αν η επίδοσή του σε εργασίες της κλάσης Τ, όπως αποτιμάται από το μέτρο Ρ, βελτιώνεται με την εμπειρία Ε».[9] Αυτός ο ορισμός είναι σημαντικός για τον καθορισμό της μηχανικής μάθησης σε βασικό λειτουργικό πλαίσιο παρά με γνωστικούς όρους, ακολουθώντας έτσι

την πρόταση του Alan Turing στην εργασία του «Υπολογιστικές μηχανές και Νοημοσύνη», ότι το ερώτημα αν μπορούν οι μηχανές να σκεφτούν, μπορεί να αντικατασταθεί με το ερώτημα αν μπορούν οι μηχανές να κάνουν αυτό που εμείς (ως σκεπτόμενες οντότητες) μπορούμε να κάνουμε.»

4.Οι χρονοσειρές στην Μηχανική μάθηση

Η ακριβής πρόβλεψη χρονοσειρών είναι σημαντική γιατί εμφανίζοντας τον τρόπο με τον οποίο το παρελθόν συνεχίζει να επηρεάζει το μέλλον για τον προγραμματισμό της ημέρας (και όχι μονο άναλογα στον κλάδο στον οποιο εξετάζουμε τις χρονοσειρές) μας στις καθημερινές δραστηριότητες. Τα τελευταία χρόνια, έχει εξελιχθεί μια μεγάλη βιβλιογραφία σχετικά με τη χρήση της υπολογιστικής νοημοσύνης σε πολλές εφαρμογές πρόβλεψης. Διάφορες τεχνικές υπολογιστικής νοημοσύνης (γενετικοί αλγόριθμοι, νευρωνικά δίκτυα, μηχανή διάνυσμα υποστήριξης, ασαφείς κανόνες) συνδυάζονται σε ένα ξεχωριστό τρόπος πρόβλεψης ενός συνόλου χρονικών σειρών που αναφέρονται.

Η πρόβλεψη χρονικών σειρών (TSF) προβλέπει τη συμπεριφορά ενός δεδομένου φαινομένου που βασίζεται αποκλειστικά στο παρελθόν μοτίβα του ίδιου γεγονότος. Αρκετά TSF (κυρίως αναπτύχθηκαν στατιστικές) μέθοδοι, π.χ. HoltWinters ή ARIMA του Box-Jenkin. Οι χρονοσειρές μελετώνται για διάφορους σκοπούς όπως η πρόβλεψη του μέλλοντος με βάση τη γνώση του παρελθόντος, η κατανόηση του φαινομένου που βασίζεται στα μέτρα ή απλά μια συνοπτική περιγραφή των κύριων χαρακτηριστικών της σειράς.

Τις τελευταίες δύο δεκαετίες, τα μοντέλα μηχανικής μάθησης έχουν τραβήξει την προσοχή και έχουν καθιερωθεί ως σοβαροί διεκδικητές των κλασικών στατιστικών μοντέλων στην κοινότητα πρόβλεψης.

5.time series forecasting, Forecasting and Modeling

5.1Εισαγωγή στην Ανάλυση χρονοσειρών

Στην πράξη προσαρμόζεται ένα κατάλληλο μοντέλο σε μια δεδομένη χρονική σειρά και την αντίστοιχη.Οι παράμετροι εκτιμώνται χρησιμοποιώντας τις γνωστές τιμές δεδομένων. Η διαδικασία προσαρμογής μιας χρονοσειράς σε ένα σωστό μοντέλο ονομάζεται Ανάλυση Χρονοσειρών . Περιλαμβάνει μεθόδους που προσπαθούν να κατανοούν τη φύση της σειράς και είναι συχνά χρήσιμο για μελλοντικές προβλέψεις και προσομοίωση.Στην πρόβλεψη χρονοσειρών, οι παρατηρήσεις του παρελθόντος συλλέγονται και αναλύονται για να αναπτυχθεί μια κατάλληλη μαθηματικό μοντέλο που αποτυπώνει την υποκείμενη διαδικασία παραγωγής δεδομένων για τη σειρά . Στη συνέχεια προβλέπονται τα μελλοντικά γεγονότα χρησιμοποιώντας το μοντέλο. Αυτή η προσέγγιση είναι ιδιαίτερα χρήσιμη όταν δεν υπάρχουν πολλές γνώσεις για το στατιστικό πρότυπο που ακολουθούν οι διαδοχικές παρατηρήσεις ή όταν λείπει ένα ικανοποιητικό επεξηγηματικό μοντέλο. Η πρόβλεψη χρονοσειρών είναι σημαντική εφαρμογές σε διάφορους τομείς. Συχνά είναι πολύτιμες στρατηγικές αποφάσεις και προληπτικά μέτρα λαμβάνονται με βάση τα αποτελέσματα των προβλέψεων. Κάνοντας έτσι μια καλή πρόβλεψη, δηλαδή προσαρμόζοντας ένα κατάλληλο μοντέλο σε μια χρονοσειρά είναι

πολύ σημαντική. Τις τελευταίες δεκαετίες έχουν γίνει πολλές προσπάθειες από ερευνητές για την ανάπτυξη και βελτίωση κατάλληλων μοντέλων πρόβλεψης χρονοσειρών.

5.2 Η πρόβλεψη χρονοσειρών

Είναι η διαδικασία ανάλυσης δεδομένων χρονοσειρών με χρήση στατιστικών και μοντελοποίησης για την πραγματοποίηση προβλέψεων και την ενημέρωση για τη λήψη στρατηγικών αποφάσεων. Δεν είναι πάντα μια ακριβής πρόβλεψη και η πιθανότητα των προβλέψεων μπορεί να διαφέρει πολύ — ειδικά όταν έχουμε να κάνουμε με τις συνήθως κυμαινόμενες μεταβλητές στα δεδομένα χρονοσειρών καθώς και με παράγοντες εκτός του ελέγχου μας. Ωστόσο, η πρόβλεψη της εικόνας σχετικά με το ποια αποτελέσματα είναι πιο πιθανό - ή λιγότερο πιθανό - να προκύψουν από άλλα πιθανά αποτελέσματα. Συχνά, όσο πιο ολοκληρωμένα είναι τα δεδομένα που έχουμε, τόσο πιο ακριβείς μπορεί να είναι οι προβλέψεις. Ενώ η πρόβλεψη και η «πρόβλεψη» σημαίνουν γενικά το ίδιο πράγμα, υπάρχει μια αξιοσημείωτη διάκριση. Σε ορισμένους κλάδους, η πρόβλεψη μπορεί να αναφέρεται σε δεδομένα σε ένα συγκεκριμένο μελλοντικό χρονικό σημείο, ενώ η πρόβλεψη αναφέρεται σε μελλοντικά δεδομένα γενικά. Η πρόβλεψη σειρών χρησιμοποιείται συχνά σε συνδυασμό με την ανάλυση χρονοσειρών. Η ανάλυση χρονοσειρών περιλαμβάνει την ανάπτυξη μοντέλων για την κατανόηση των δεδομένων για την κατανόηση των υποκείμενων αιτιών. Η ανάλυση μπορεί να παρέχει το «γιατί» πίσω από τα αποτελέσματα που βλέπετε. Στη συνέχεια, η πρόβλεψη κάνει το επόμενο βήμα για το τι πρέπει να γίνει με αυτή τη γνώση και τις προβλέψιμες προεκτάσεις του τι μπορεί να συμβεί στο μέλλον.

Φυσικά, υπάρχουν περιορισμοί όταν αντιμετωπίζουμε το απρόβλεπτο και το άγνωστο. Η πρόβλεψη χρονοσειρών δεν είναι αλάνθαστη και δεν είναι κατάλληλη ή χρήσιμη για όλες τις καταστάσεις. Επειδή στην πραγματικότητα δεν υπάρχει ρητό σύνολο κανόνων για το πότε πρέπει ή όχι να χρησιμοποιείτε την πρόβλεψη, εναπόκειται στους αναλυτές και τις ομάδες δεδομένων να γνωρίζουν τους περιορισμούς της ανάλυσης και τι μπορούν να υποστηρίξουν τα μοντέλα τους. Δεν ταιριάζει κάθε μοντέλο σε κάθε σύνολο δεδομένων ή δεν απαντά σε κάθε ερώτηση. Οι ομάδες δεδομένων θα πρέπει να χρησιμοποιούν πρόβλεψη χρονοσειρών όταν κατανοούν την επιχειρηματική ερώτηση και έχουν τα κατάλληλα δεδομένα και δυνατότητες πρόβλεψης για να απαντήσουν σε αυτήν την ερώτηση. Η καλή πρόβλεψη λειτουργεί με καθαρά, χρονικά σφραγισμένα δεδομένα και μπορεί να προσδιορίσει τις γνήσιες τάσεις και μοτίβα στα ιστορικά δεδομένα. Οι αναλυτές μπορούν να πουν τη διαφορά μεταξύ τυχαίων διακυμάνσεων ή ακραίων τιμών και μπορούν να διαχωρίσουν τις γνήσιες πληροφορίες από τις εποχιακές διακυμάνσεις. Η ανάλυση χρονοσειρών δείχνει πώς αλλάζουν τα δεδομένα με την πάροδο του χρόνου και η καλή πρόβλεψη μπορεί να προσδιορίσει την κατεύθυνση προς την οποία αλλάζουν τα δεδομένα.

5.3 Χρονοσειρά και Στοχαστική Διαδικασία

Μια χρονοσειρά είναι μη ντετερμινιστικής φύσης, δηλαδή δεν μπορούμε να προβλέψουμε με βεβαιότητα τι θα γίνει συμβαίνουν στο μέλλον. Γενικά μια χρονοσειρά {x(t),t = 0,1, 2,...} θεωρείται ότι ακολουθεί ορισμένες μοντέλο πιθανότητας που περιγράφει την κοινή κατανομή της τυχαίας μεταβλητής x_tThe η μαθηματική έκφραση που περιγράφει τη δομή πιθανοτήτων μιας χρονοσειράς ορίζεται ως στοχαστική διαδικασία . Έτσι η ακολουθία των παρατηρήσεων της σειράς είναι στην πραγματικότητα ένα δείγμα συνειδητοποίηση της στοχαστικής διαδικασίας που το παρήγαγε.

Μια συνήθης υπόθεση είναι ότι οι μεταβλητές χρονοσειρών x_t είναι ανεξάρτητες και πανομοιότυπες κατανέμεται (i.i.d) ακολουθώντας την κανονική κατανομή. Ωστόσο, όπως αναφέρεται, ενα ενδιαφέρον σημείο είναι ότι οι χρονοσειρές στην πραγματικότητα δεν είναι ακριβώς i.i.d. ακολουθούν λίγο πολύ

κάποιους κανονικό μοτίβο μακροπρόθεσμα. Για παράδειγμα αν η θερμοκρασία σήμερα μιας συγκεκριμένης πόλης είναι εξαιρετικά υψηλή, τότε μπορεί εύλογα να υποτεθεί ότι η αυριανή θερμοκρασία θα είναι επίσης πιθανό να είναι υψηλό. Αυτός είναι ο λόγος για τον οποίο η πρόβλεψη χρονοσειρών χρησιμοποιώντας μια κατάλληλη τεχνική,οι αποδόσεις καταλήγουν κοντά στην πραγματική τιμή.

6.Μοντέλα που θα χρησιμοποιήσουμε

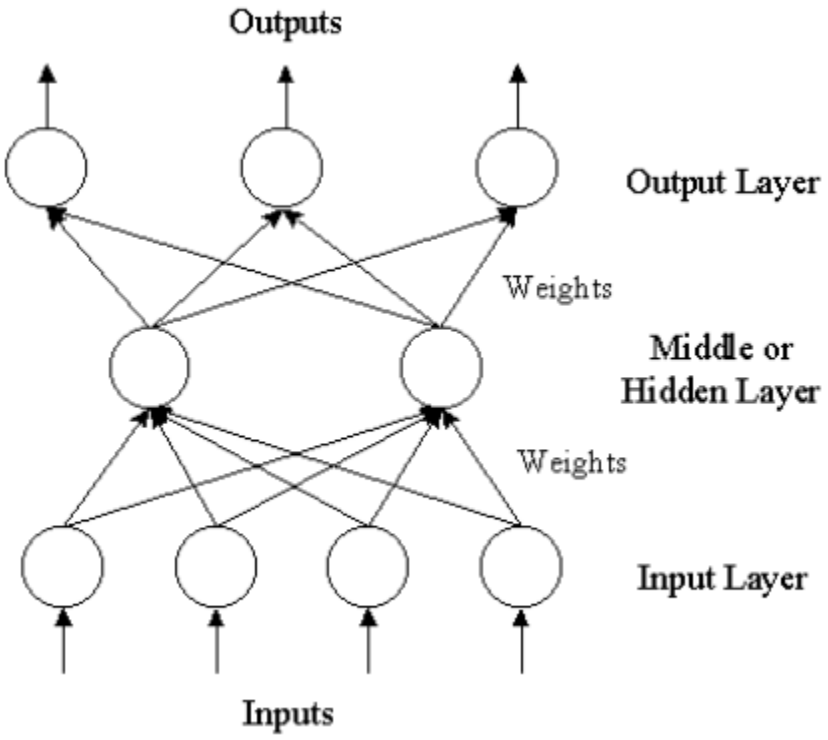
6.1Artificial Neural Networks (ANNs)

Η προσέγγιση των τεχνητών νευρωνικών δικτύων (ANNs) έχει προταθεί ως εναλλακτική τεχνική για την πρόβλεψη χρονοσειρών και κέρδισε τεράστια δημοτικότητα τα τελευταία χρόνια. Ο βασικός στόχος των ANN ήταν η κατασκευή ενός μοντέλου για τη μίμηση της νοημοσύνης του ανθρώπινου εγκεφάλου στη μηχανή. Παρόμοια με το έργο ενός ανθρώπινου εγκεφάλου, τα ANN προσπαθούν να αναγνωρίσουν κανονικότητες και μοτίβα στα δεδομένα εισόδου, να μάθουν από την εμπειρία και στη συνέχεια να παρέχουν γενικευμένα αποτελέσματα με βάση τις γνωστές προηγούμενες γνώσεις τους. Αν και η ανάπτυξη των ANN είχε κυρίως βιολογικά κίνητρα, αλλά στη συνέχεια εφαρμόστηκαν σε πολλούς διαφορετικούς τομείς, ειδικά για σκοπούς πρόβλεψης και ταξινόμησης. Παρακάτω θα αναφέρουμε τα κύρια χαρακτηριστικά των ANN, τα οποία τα καθιστούν αρκετά αγαπημένα για ανάλυση και πρόβλεψη χρονοσειρών. Πρώτον, τα ANN βασίζονται σε δεδομένα και προσαρμόζονται στη φύση τους. Δεν χρειάζεται να προσδιορίσετε μια συγκεκριμένη φόρμα μοντέλου ή να κάνετε οποιαδήποτε εκ των προτέρων υπόθεση σχετικά με τη στατιστική κατανομή των δεδομένων. το επιθυμητό μοντέλο διαμορφώνεται προσαρμοστικά με βάση τα χαρακτηριστικά που παρουσιάζονται από τα δεδομένα. Αυτή η προσέγγιση είναι αρκετά χρήσιμη για πολλές πρακτικές καταστάσεις, όπου δεν υπάρχει διαθέσιμη θεωρητική καθοδήγηση για μια κατάλληλη διαδικασία παραγωγής δεδομένων. Δεύτερον, τα ANN είναι εγγενώς μη γραμμικά, γεγονός που τα καθιστά πιο πρακτικά και ακριβή στη μοντελοποίηση πολύπλοκων προτύπων δεδομένων, σε αντίθεση με διάφορες παραδοσιακές γραμμικές προσεγγίσεις, όπως οι μέθοδοι ARIMA . Υπάρχουν πολλές περιπτώσεις που υποδηλώνουν ότι τα ANN έκαναν πολύ καλύτερη ανάλυση και πρόβλεψη από διάφορα γραμμικά μοντέλα. Τέλος, όπως προτείνουν οι Hornik και Stinchcombe, τα ANN είναι καθολικοί λειτουργικοί προσεγγιστές. Έχουν δείξει ότι ένα δίκτυο μπορεί να προσεγγίσει οποιαδήποτε συνεχή λειτουργία με οποιαδήποτε επιθυμητή ακρίβεια. Τα ANN χρησιμοποιούν παράλληλη επεξεργασία των πληροφοριών από τα δεδομένα για να προσεγγίσουν μια μεγάλη κατηγορία συναρτήσεων με υψηλό βαθμό ακρίβειας. Επιπλέον, μπορούν να αντιμετωπίσουν καταστάσεις, όπου τα δεδομένα εισόδου είναι λανθασμένα, ελλιπή ή ασαφή.

6.1.2 Αρχιτεκτονική ANN

Τα πιο ευρέως χρησιμοποιούμενα ANN σε προβλήματα πρόβλεψης είναι τα πολυστρωματικά perceptrons (MLPs) Τα οποία χρησιμοποιούν ένα ενιαίο κρυφό δίκτυο τροφοδοσίας προς τα εμπρός (FNN). Το μοντέλο χαρακτηρίζεται από ένα δίκτυο τριών στρωμάτων, δηλ. επίπεδο εισόδου, κρυφού και εξόδου, που συνδέονται με ακυκλικούς συνδέσμους. Μπορεί να υπάρχουν περισσότερα από ένα κρυφά επίπεδα. Οι κόμβοι σε διάφορα στρώματα είναι επίσης γνωστοί ως στοιχεία επεξεργασίας. Η αρχιτεκτονική τροφοδοσίας τριών επιπέδων των μοντέλων ANN

μπορεί να είναι απεικονίζεται διαγραμματικά ως εξής:



Εικόνα 4: Αρχιτεκτονική τροφοδοσίας τριών επιπέδων μοντέλου ANN

Η έξοδος του μοντέλου υπολογίζεται χρησιμοποιώντας την ακόλουθη μαθηματική έκφραση:

$$y_t = \alpha_0 + \sum_{j=1}^{q} \alpha_j g\left(\beta_{0j} + \sum_{i=1}^{p} \beta_{ij} y_{t-i}\right) + \varepsilon_t, \ \forall t$$

Εδώ y_{t-i} (i = 1,2,..., p) είναι οι είσοδοι p και y_tείναι η έξοδος. Οι ακέραιοι p, q είναι ο αριθμός των εισόδου και κρυφούς κόμβους αντίστοιχα a_j (j = 0,1,2,...,q) και $\beta_{ij}(i = 0,1,2,\ldots,p; j = 0,1,2,\ldots,q)$ είναι τα βάρη σύνδεσης και ε_t είναι το τυχαίο a_0 και β_{0j} είναι οι όροι μεροληψίας. Συνήθως, το λογιστική σιγμοειδής συνάρτηση $g(x) = \frac{1}{1+e^{-x}}$ φαρμόζεται ως η μη γραμμική συνάρτηση ενεργοποίησης. Αλλα). Μπορούν επίσης να χρησιμοποιηθούν συναρτήσεις ενεργοποίησης, όπως γραμμική, υπερβολική εφαπτομένη, Gaussian κ.λπ. Το μοντέλο τροφοδοσίας ANN στην πραγματικότητα εκτελεί μια μη γραμμική λειτουργική χαρτογράφηση από το προηγούμενες παρατηρήσεις της χρονοσειράς στη μελλοντική τιμή, δηλ y_t = f($y_{t-1}, y_{t-2}, \ldots \ldots y_{t-p}$,w)+ ε_t, *όπου το w είναι ένα διάνυσμα όλων των παραμέτρων και f είναι μια συνάρτηση που καθορίζεται από τη δομή του δικτύου και βάρη σύνδεσης.*

Για την εκτίμηση των βαρών σύνδεσης χρησιμοποιούνται μη γραμμικές διαδικασίες ελαχίστου τετραγώνου, οι οποίες είναι με βάση την ελαχιστοποίηση της συνάρτησης σφάλματος:

$$F(\Psi) = \sum_t e_t^2 = \sum_t (y_t - \hat{y}_t)^2$$

Εδώ η Ψ είναι ο χώρος όλων των βαρών σύνδεσης.Οι τεχνικές βελτιστοποίησης που χρησιμοποιούνται για την ελαχιστοποίηση της συνάρτησης σφάλματος αναφέρονται ως Κανόνες μάθησης. Ο πιο γνωστός κανόνας μάθησης στη λογοτεχνία είναι ο Backpropagation ή Γενικευμένος κανόνας Δέλτα

6.2 fb prophet

Από προεπιλογή, ο Prophet χρησιμοποιεί ένα γραμμικό μοντέλο για την πρόβλεψή του. Κατά την πρόβλεψη της ανάπτυξης, υπάρχει συνήθως κάποιο μέγιστο δυνατό σημείο: συνολικό μέγεθος αγοράς, συνολικό μέγεθος πληθυσμού, κ.λπ. Αυτό ονομάζεται φέρουσα ικανότητα και η πρόβλεψη θα πρέπει να κορεστεί σε αυτό το σημείο. Τώρα περιγράφουμε ένα μοντέλο πρόβλεψης χρονοσειρών που έχει σχεδιαστεί για να χειρίζεται τα κοινά χαρακτηριστικά των επιχειρηματικών χρονοσειρών που φαίνονται στο. Είναι σημαντικό ότι έχει επίσης σχεδιαστεί για να έχει διαισθητικές παραμέτρους που μπορούν να προσαρμοστούν χωρίς να γνωρίζουμε τις λεπτομέρειες του υποκείμενου μοντέλου. Αυτό είναι απαραίτητο για αναλυτής για να συντονίσει αποτελεσματικά το μοντέλο όπως περιγράφεται στο. Η εφαρμογή μας είναι διαθέσιμη ως λογισμικό ανοιχτού κώδικα σε Python και R, που ονομάζεται Prophet. Χρησιμοποιούμε ένα μοντέλο αποσυνθέσιμης χρονοσειράς (Harvey & Peters 1990) με τρία κύρια στοιχεία του μοντέλου: τάση, εποχικότητα και διακοπές. Συνδυάζονται στην ακόλουθη εξίσωση:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t.$$

Το $g_{(t)}$ είναι η συνάρτηση τάσης που μοντελοποιεί τις μη περιοδικές αλλαγές στην τιμή των χρονοσειρών, το $s_{(t)}$αντιπροσωπεύει περιοδικές αλλαγές (π.χ., εβδομαδιαία και ετήσια εποχικότητα), και το h(t) αντιπροσωπεύει τα αποτελέσματα των αργιών που συμβαίνουν στις δυνητικά ακανόνιστα προγράμματα για μία ή περισσότερες ημέρες. Ο όρος σφάλματος αντιπροσωπεύει οποιεσδήποτε ιδιοσυγκρασιακές αλλαγές που δεν καλύπτονται από το μοντέλο. Αργότερα θα κάνουμε την παραμετρική υπόθεση ότι είναι κανονικά κατανεμημένη. Αυτή η προδιαγραφή είναι παρόμοια με ένα γενικευμένο προσθετικό μοντέλο (GAM) (Hastie & Tibshirani1987), μια κατηγορία μοντέλων παλινδρόμησης με δυνητικά μη γραμμικούς εξομαλυντές που εφαρμόζονται σε εκείνα τα ελαττώματα. Εδώ χρησιμοποιούμε μόνο το χρόνο ως παλινδρομικό αλλά πιθανώς αρκετές γραμμικές και μη γραμμικές συναρτήσεις του χρόνου ως συνιστώσες. Η μοντελοποίηση της εποχικότητας ως πρόσθετης συνιστώσας είναι η ίδια προσέγγιση που ακολουθείται από την εκθετική εξομάλυνση (Gardner 1985). Η πολλαπλασιαστική εποχικότητα, όπου το εποχιακό αποτέλεσμα είναι ένας παράγοντας που πολλαπλασιάζει g(t), μπορεί να επιτευχθεί μέσω αναλογικού μετασχηματισμού. Η διατύπωση GAM έχει το πλεονέκτημα ότι αποσυντίθεται εύκολα και φιλοξενεί νέα συστατικά όπως απαιτείται, για παράδειγμα όταν αναγνωρίζεται μια νέα πηγή εποχικότητας ed.GAMs επίσης πολύ γρήγορα, είτε χρησιμοποιώντας back tting είτε L-BFGS (Byrd et al. 1995) (προτιμάμε το δεύτερο) έτσι ώστε ο χρήστης να μπορεί να αλλάξει διαδραστικά τις παραμέτρους του μοντέλου. Στην πραγματικότητα, πλαισιώνουμε το πρόβλημα πρόβλεψης ως άσκηση καμπυλότητας, η οποία είναι εγγενώς διαφορετική από τα μοντέλα χρονοσειρών που εξηγούν ρητά τη δομή της χρονικής εξάρτησης στα δεδομένα. Ενώ εγκαταλείπουμε ορισμένα σημαντικά συμπερασματικά πλεονεκτήματα της χρήσης

ενός παραγωγικού μοντέλου όπως το ARIMA, αυτή η διατύπωση παρέχει μια σειρά από πρακτικά πλεονεκτήματα:

- Flexibility: Μπορούμε εύκολα να προσαρμόσουμε την εποχικότητα με πολλαπλές περιόδους και να αφήσουμε τους αναλυτές κάνουν διαφορετικές υποθέσεις σχετικά με τις τάσεις.
- Σε αντίθεση με τα μοντέλα ARIMA, οι μετρήσεις δεν χρειάζεται να είναι σε τακτά χρονικά διαστήματα και δεν χρειάζεται να αντικαταστήσουμε τις τιμές που λείπουν π.χ. από την αφαίρεση ακραίων στοιχείων.
- Η εκμάθηση είναι πολύ γρήγορη, επιτρέποντας στον αναλυτή να εξερευνήσει διαδραστικά πολλές προδιαγραφές μοντέλων, για παράδειγμα σε μια εφαρμογή Shiny (Chang et al. 2015).
-Το μοντέλο πρόβλεψης έχει εύκολα ερμηνεύσιμες παραμέτρους που μπορούν να αλλάξουν από τον αναλυτή για να επιβάλει υποθέσεις στην πρόβλεψη. Επιπλέον, οι αναλυτές συνήθως έχουν εμπειρία με την παλινδρόμηση και είναι εύκολα σε θέση να επεκτείνουν το μοντέλο ώστε να συμπεριλάβει νέα στοιχεία..

Η αυτόματη πρόβλεψη έχει μακρά ιστορία, με πολλές μεθόδους προσαρμοσμένες σε συγκεκριμένους τύπους χρονοσειρών (Tashman & Leach 1991, De Gooijer & Hyndman 2006). Η προσέγγισή μας καθοδηγείται τόσο από τη φύση των χρονοσειρών που προβλέπουμε στο Facebook (τμηματικές τάσεις, πολλαπλή εποχικότητα, αργίες) όσο και από τις προκλήσεις που σχετίζονται με την πρόβλεψη σε κλίμακα

6.3 Linear Regression

Ο οπίσθιος πολλαπλασιασμός είναι μια τεχνική αναρρίχησης σε λόφους. Τρέχει το κίνδυνος παγίδευσης στο τοπικό βέλτιστο. Το σημείο εκκίνησης των βαρών σύνδεσης γίνεται ένα σημαντικό ζήτημα για να να μειώσει την πιθανότητα εγκλωβισμού σε τοπικό βέλτιστο. Η τυχαία προετοιμασία βάρους δεν εγγυάται τη δημιουργία μια καλή αφετηρία. Μπορεί να ενισχυθεί με πολλαπλές γραμμικές οπισθοδρόμηση. Σε αυτή τη μέθοδο, τα βάρη μεταξύ του στρώματος εισόδου και το κρυφό στρώμα εξακολουθούν να αρχικοποιούνται τυχαία αλλά βάρη μεταξύ κρυφού στρώματος και στρώματος εξόδου λαμβάνεται από πολλαπλή γραμμική παλινδρόμηση. Το βάρος w_{ij} μεταξύ του κόμβου εισόδου i και του κρυφού κόμβου j αρχικοποιείται με ομοιόμορφη τυχαιοποίηση. Μόλις εισαχθεί x_i^s του δείγματος s έχει τροφοδοτηθεί στον κόμβο εισόδου και $w_{ij}'s$ έχουν εκχωρηθεί τιμές, η τιμή εξόδου s r_j^s του κρυφού κόμβου j μπορεί να υπολογιστεί ως:

$$R_j^s = f(\sum_i w_{ij} x_i^s),$$

όπου f είναι συνάρτηση μεταφοράς. Η τιμή εξόδου του ο κόμβος εξόδου μπορεί να υπολογιστεί ως:

$$y^s = f(\sum_j v_j R_j^s)$$

πού v_jείναι το βάρος μεταξύ του κρυφού στρώματος και του στρώμα εξόδου. Υποθέστε σιγμοειδές συνάρτηση $f_{(x)=}\frac{1}{1+e^{-x}}$ χρησιμοποιείται ως συνάρτηση μεταφοράς του δικτύου. Με την επέκταση του Taylor,

$$f(x) \cong \frac{1}{2} + \frac{x}{4}$$

Εφαρμόζοντας τη γραμμική προσέγγιση, έχουμε την παρακάτω προσεγγιστική γραμμική σχέση μεταξύ των
έξοδος y και vj:

$$y^s = \frac{1}{2} + \frac{1}{4}\left(\sum_j^m v_j R_j^s\right)$$

$$\text{or } 4y^s - 2 = v_1 R_1^s + v_2 R_2^s + \ldots + v_m R_m^s$$

$$s = 1,2,..N$$

όπου m είναι ο αριθμός των κρυφών κόμβων.Ν είναι ο συνολικός αριθμός δειγμάτων εκπαίδευσης. Το σύνολο των εξισώσεων είναι ένα τυπικό μοντέλο πολλαπλής γραμμικής παλινδρόμησης. Τα R_i^s θεωρούνται ως παλίνδρομοι $v_j's$ μπορούν να εκτιμηθούν με τυπική μέθοδο παλινδρόμησης. Μόλις ληφθούν τα $v_j's$ ολοκληρώνεται η προετοιμασία του δικτύου και ξεκινά η εκπαίδευση.

7. Μέτρηση ακρίβειας προβλέψεων

Accuracy measuring tool	Formulation	Reference
MAE	$MAE = \frac{\sum_{t=1}^{n} \lvert e_t \rvert}{n}$	Makridakis *et al.*, 2003
ME	$ME = \frac{\sum_{t=1}^{n} e_t}{n}$	Makridakis *et al.*, 2003
MSE	$MSE = \frac{1}{n}\sum_{t=1}^{n} e_t^2$	Makridakis *et al.*, 2003
MPE	$MPE = \frac{1}{n}\sum_{t=1}^{n} PE_t$	Makridakis *et al.*, 2003
MAPE	$MPE = \frac{1}{n}\sum_{t=1}^{n} \lvert PE_t \rvert$	Makridakis *et al.*, 2003

$$PE_t = \left(\frac{Y_t - F_t}{Y_t} \right) \times 100$$

Where and F_t is the forecasted value for time t.

Εικόνα 5: Forecasts accuracy measuring table

Β.Πρακτικό κομμάτι

1.Ανάλυση δεδομένων

Το πρώτο πράγμα που θα πρέπει να κάνουμε είναι να κατεβάσουμε τα δεδομένα μας και να ανοίξουμε το αρχείο και να δούμε λίγο την δομή του (επισυνάπτουμε τα δεδομένα που έχουμε). Τα δεδομένα μας είναι σε αρχείο txt(household_power_consumption.txt).

Πρέπει να εξετάσουμε τα labels που έχουμε στο αρχείο και τι πληροφορία μας δίνει. Τα labels είναι:

- Date: Ημερομηνία της μορφής M/ D/Y
- Time: Ώρα σε 24h Μορφή h:m:s
- Global_active_power: Η συνολική ενεργός ισχύς που καταναλώνεται από το νοικοκυριό (κιλοβάτ).)
- Global_reactive_power: Η συνολική άεργη ισχύς που καταναλώνεται από το νοικοκυριό (κιλοβάτ).
- Voltage: Μέση τάση (βολτ)
- Global_intensity: Μέση ένταση ρεύματος (ενισχυτές)

- Sub_metering_1: Ενεργή ενέργεια για κουζίνα (watt-ώρες ενεργού ενέργειας).
- Sub_metering_2: Ενεργή ενέργεια για πλυντήρια (watt-ώρες ενεργού ενέργειας).
- Sub_metering_3: Ενεργή ενέργεια για συστήματα ελέγχου του κλίματος (watt-ώρες ενεργού ενέργειας).

Η ενεργός και άεργη ενέργεια αναφέρεται στις τεχνικές λεπτομέρειες του εναλλακτικού ρεύματος.

Μια τέταρτη μεταβλητή υπομέτρησης μπορεί να δημιουργηθεί αφαιρώντας το άθροισμα τριών καθορισμένων μεταβλητών υπομέτρησης από τη συνολική ενεργό ενέργεια.

Πως το επιτυγχάνουμε αυτό?

Αρχικά παρατηρούμε ότι κάποιες τιμές λείπουν έτσι πήγαμε και τις αλλάξαμε ονομασία από ? σε NaN

```
21/12/2006;11:20:00;0.244;0.000;242.080;1.000;0.000;0.000;0.000
21/12/2006;11:21:00;0.242;0.000;241.670;1.000;0.000;0.000;0.000
21/12/2006;11:22:00;0.244;0.000;242.290;1.000;0.000;0.000;0.000
21/12/2006;11:23:00;?;?;?;?;?;?;
21/12/2006;11:24:00;?;?;?;?;?;?;
21/12/2006;11:25:00;0.246;0.000;241.740;1.000;0.000;0.000;0.000
21/12/2006;11:26:00;0.246;0.000;241.830;1.000;0.000;0.000;0.000
21/12/2006;11:27:00;0.244;0.000;240.960;1.000;0.000;0.000;0.000
21/12/2006;11:28:00;0.244;0.000;241.370;1.000;0.000;0.000;0.000
21/12/2006;11:29:00;0.244;0.000;241.330;1.000;0.000;0.000;0.000
21/12/2006;11:30:00;0.244;0.000;241.760;1.000;0.000;0.000;0.000
```

Για να φορτώσουμε τα δεδομένα μας να αφαιρέσουμε τις τιμές που είναι μέσα στο dataset μας δημιουργήσαμε το script με την χρήση της Python και της βιβλιοθήκης pandas (prepare_data.py για περισσότερες λεπτομερείς) και αποθηκεύουμε τα αποτελέσματα σε csv αρχείο (household_power_consumption.csv) και προσθέσαμε ένα ακόμα πεδίο το sub_metering_4: δημιουργήθηκε με τον εξής τρόπο:

dataset['sub_metering_4'] = (global_active_power* 1000 / 60) - (values[:,4] + values[:,5] + values[:,6]).

Ουσιαστικά ο υπολογισμός είναι ο εξής: [global_active_power]*1000/60 και αφαιρέσαμε το άθροισμα από τα column 4,5,6 για κάθε μία γραμμή ξεχωριστά.

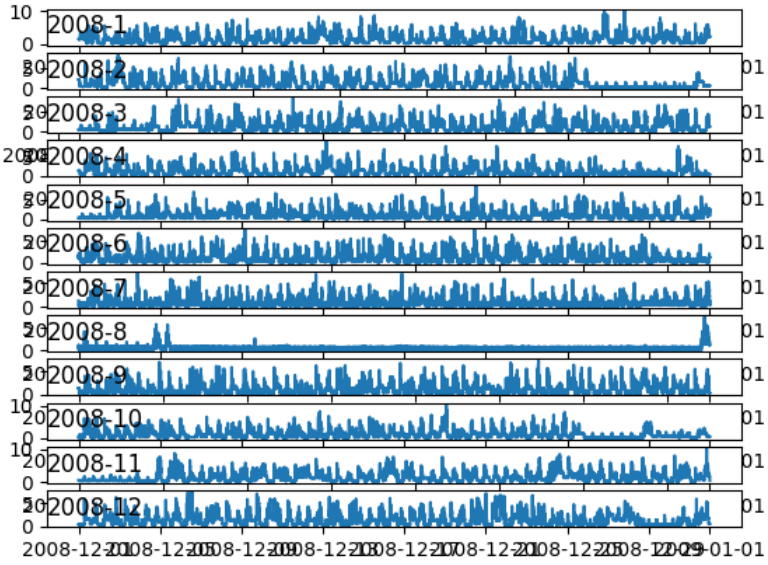
	A	B	C	D	E	F	G	H	I
1	datetime	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3	sub_metering_4
2	12/16/2006 17:24	4.216	0.418	234.84	18.4	0	1	17	52.26667
3	12/16/2006 17:25	5.36	0.436	233.63	23	0	1	16	72.333336
4	12/16/2006 17:26	5.374	0.498	233.29	23	0	2	17	70.566666
5	12/16/2006 17:27	5.388	0.502	233.74	23	0	1	17	71.8
6	12/16/2006 17:28	3.666	0.528	235.68	15.8	0	1	17	43.1
7	12/16/2006 17:29	3.52	0.522	235.02	15	0	2	17	39.666668
8	12/16/2006 17:30	3.702	0.52	235.09	15.8	0	1	17	43.7
9	12/16/2006 17:31	3.7	0.52	235.22	15.8	0	1	17	43.666668
10	12/16/2006 17:32	3.668	0.51	233.99	15.8	0	1	17	43.133335
11	12/16/2006 17:33	3.662	0.51	233.86	15.8	0	2	16	43.033333
12	12/16/2006 17:34	4.448	0.498	232.86	19.6	0	1	17	56.13333
13	12/16/2006 17:35	5.412	0.47	232.78	23.2	0	1	17	72.2
14	12/16/2006 17:36	5.224	0.478	232.99	22.4	0	1	16	70.066666
15	12/16/2006 17:37	5.268	0.398	232.91	22.6	0	2	17	68.8
16	12/16/2006 17:38	4.054	0.422	235.24	17.6	0	1	17	49.566666

Εικόνα 6: Screenshot με τα Δεδομένα

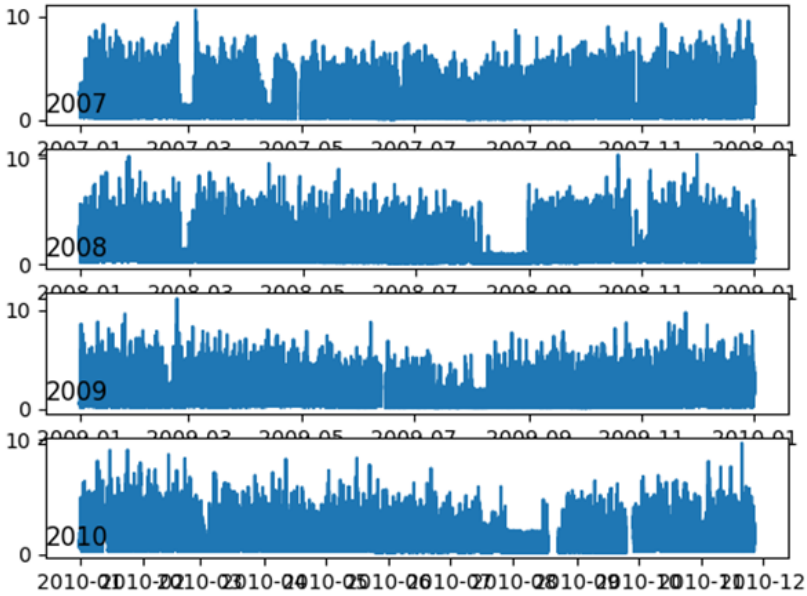
Τα νέα πεδία είναι τα εξής :

- Datetime: Κάναμε Merge το date και το time
- Global_active_power: έμεινε ίδιο
- Global_reactive_power: έμεινε ίδιο
- Voltage: έμεινε ίδιο
- Global_intensity: έμεινε ίδιο
- Sub_metering_1: έμεινε ίδιο
- Sub_metering_2: έμεινε ίδιο
- Sub_metering_3: έμεινε ίδιο
- Sub_metering_4: Το δημιουργήσαμε

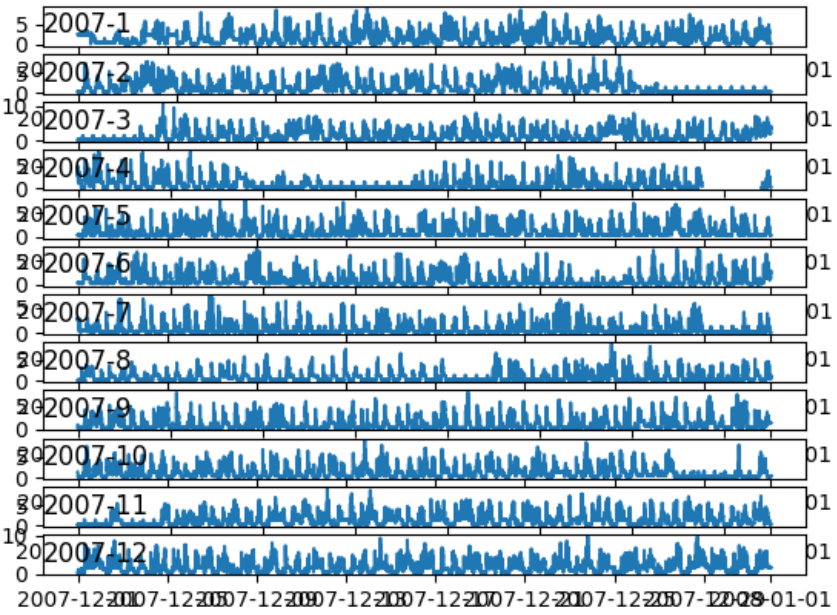
Οπτικοποίηση των δεδομένων μας για να δούμε μερικά στοιχεία:



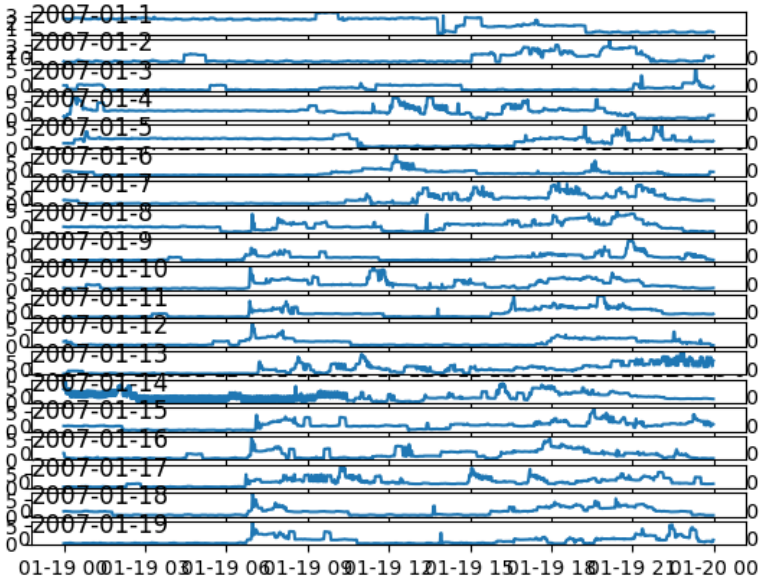
Εικόνα 7: Δεδομένα ανα μήνα για το 2008



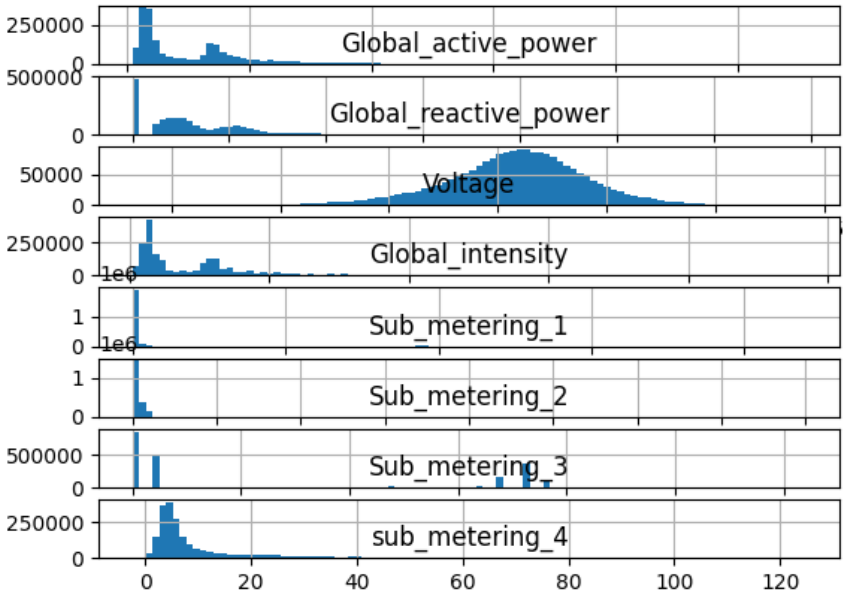
Εικόνα 8: Δεδομένα ανα χρόνο



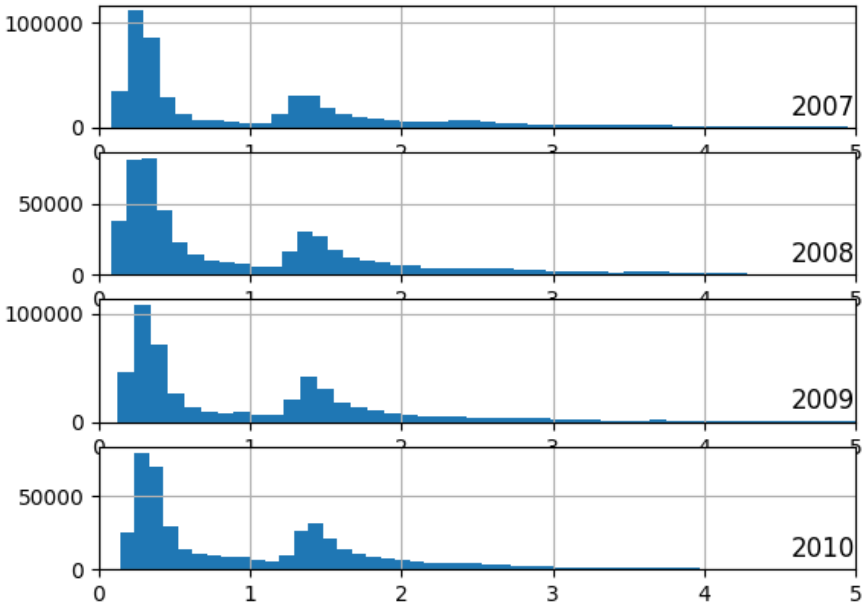
Εικόνα 9: Δεδομένα για το 2007



Εικόνα 10: Δεδομένα για το Ιανουάριο του 2007



Εικόνα 11: Ιστογράμμα για κάθε μεταβλητή



Εικόνα 12: Ιστογράμμα για κάθε μεταβλητή ανα έτος

Αφου έχουμε φτιάξει το dataset μας και το παρατηρούμε βλέπουμε ότι οι τιμές είναι ανα 1 λεπτό. Εμπειρικά καταλαβαίνουμε ότι δεν θα καταφέρουμε με αυτή την μορφή να εξετάσουμε τα δεδομένα μας όπως θέλουμε. Η σκέψη μου είναι άρα να κάνουμε grouping με την ημέρα τις τιμές έτσι ώστε να εξετάσουμε τα δεδομένα μας ανα ημέρα, ακόμα δεν υπάρχει λόγος και νόημα να εξετάσουμε ανα λεπτό για το δικό μας project και για την εφαρμογή που θέλουμε ανα λεπτό. Αυτή την στιγμή μας δημιουργεί μεγάλο πρόβλημα. π.χ. για ένα εργοστάσιο μάλλον μπορει να έχει εφαρμογή αλλα και πάλι για κάθε λεπτό μαλλον είναι υπερβολή άρα θα προσπαθήσουμε να εξετάσουμε ανα λέπτο για να δουμε εάν αυτή η σκέψη μας έχει κάποια βάση και στην συνέχεια θα το εξετάσουμε ανα ημέρα σε πρώτη φάση.

Είπαμε ήδη σε αυτό το σημείο θα εξετάσουμε με την δημιουργία έτοιμων μοντέλων μηχανικής μάθησης για να εξετάσουμε πως αντιδράνε με τα δεδομένα που έχουμε χωρίς να κάνουμε μια εκτενής έρευνα πάνω στο αντικείμενο για να δούμε γρήγορα αποτελέσματα και πως μπορούμε να συνεχίσουμε.

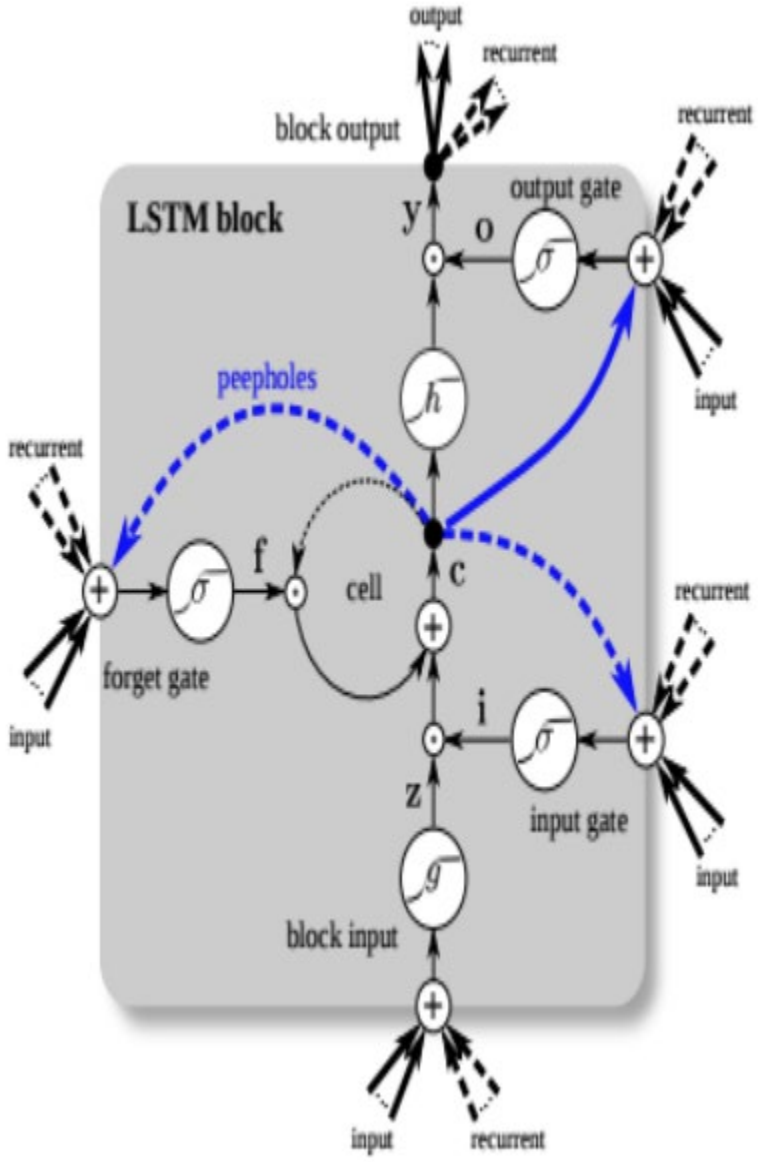
2.RNN model

Το επομενο μοντελο που θα χρησιμοποιήσουμε είναι ενα RNN (recurrent neural network) και συγκεκριμενα **LSTM**

2.1 Τι είναι το lstm;

Το δίκτυο μακράς βραχυπρόθεσμης μνήμης ή το δίκτυο LSTM είναι ένας τύπος επαναλαμβανόμενου νευρωνικού δικτύου που χρησιμοποιείται στο Deep learning

Το Long Short-Term Memory (LSTM) είναι μια αρχιτεκτονική ανατροφοδοτούμενου νευρωνικού δικτύου (RNN), το οποίο σχεδιάστηκε για να προσεγγίζει και να μοντελοποιεί χρονικές ακολουθίες και τις μεγάλου εύρους εξαρτήσεις τους με μεγαλύτερη ακρίβεια από άλλους τύπους RNN



Εικόνα 13: Αρχιτεκτονική

Το Long Short-Term Memory (LSTM) είναι μια αρχιτεκτονική ανατροφοδοτούμενου νευρωνικού δικτύου (RNN), το οποίο σχεδιάστηκε για να προσεγγίζει και να μοντελοποιεί χρονικές ακολουθίες και τις μεγάλου εύρους εξαρτήσεις τους με μεγαλύτερη ακρίβεια από άλλους τύπους RNN

Με τα LSTMs καταφέρνουμε και παρακάμπτουμε τα όποια προβλήματα δημιουργούνται από παλαιότερες αρχιτεκτονικές, επιτυγχάνοντας καλύτερη ακρίβεια στις προβλέψεις μας. Το Long Short-Term Memory (LSTM) είναι μια συγκεκριμένη αρχιτεκτονική ανατροφοδοτούμενου νευρωνικού δικτύου (RNN), το οποίο σχεδιάστηκε για να μοντελοποιεί χρονικές ακολουθίες και τις μεγάλου εύρους εξαρτήσεις τους με μεγαλύτερη ακρίβεια από άλλους τύπους RNN.

Τα LTSM Networks είναι αρκετά δημοφιλή στις μέρες μας. Τα LSTMs δεν έχουν ουσιαστική διαφορά στην αρχιτεκτονική τους από τα RNN, αλλά χρησιμοποιούν μια διαφορετική συνάρτηση για να

υπολογίσουν την κρυφή κατάσταση. Η μνήμη στα LSTM ονομάζεται κελιά (cells) και μπορούμε να τα θεωρήσουμε ως μαύρα πλαίσια που λαμβάνουν ως είσοδο την προηγούμενη κατάσταση ht-1 και την τρέχουσα είσοδο xt. Εσωτερικά αυτά τα κύτταρα αποφασίζουν τι πρέπει να διατηρούν (και τι να διαγράψουν) από τη μνήμη. Στη συνέχεια, συνδυάζουν την προηγούμενη κατάσταση, την τρέχουσα μνήμη και την είσοδο. Αποδεικνύεται ότι αυτός ο τύπος νευρωνικών δικτύων είναι πολύ αποδοτικός στη λήψη μακροπρόθεσμων εξαρτήσεων κάτι στο οποίο υστερούν τα RNNs

2.2 Τι είναι το Lag

Το Lag είναι ουσιαστικά καθυστέρηση. Ακριβώς όπως η συσχέτιση δείχνει πόσο όμοιες είναι δύο χρονοσειρές, η αυτοσυσχέτιση περιγράφει πόσο παρόμοια είναι η χρονοσειρά με τον εαυτό της.

Σκεφτείτε μια διακριτή ακολουθία τιμών, για lag 1, συγκρίνετε τη χρονοσειρά σας με μια χρονοσειρά με καθυστέρηση, με άλλα λόγια μετατοπίζετε τη χρονοσειρά κατά 1 πριν τη συγκρίνετε με την ίδια. Συνεχίστε να το κάνετε αυτό για όλο το μήκος της χρονοσειράς μετατοπίζοντάς το κατά 1 κάθε φορά. Τώρα έχετε συνάρτηση αυτοσυσχέτισης.

Από τις τιμές της συνάρτησης αυτοσυσχέτισης, μπορείτε να δείτε πόσο συσχετίζεται με τον εαυτό της. Για οποιαδήποτε χρονική σειρά θα έχετε τέλεια συσχέτιση σε καθυστέρηση/καθυστέρηση = 0, αφού συγκρίνετε τις ίδιες τιμές μεταξύ τους. Καθώς αλλάζετε τη χρονική σειρά σας, αρχίζετε να βλέπετε τις τιμές συσχέτισης να μειώνονται. Σημειώστε ότι εάν οι χρονοσειρές αποτελούνται από εντελώς τυχαίες τιμές, θα έχετε συσχέτιση μόνο στο lag=0 και όχι συσχέτιση οπουδήποτε αλλού. Στα περισσότερα σύνολα δεδομένων/χρονικές σειρές αυτό δεν συμβαίνει, καθώς οι τιμές τείνουν να μειώνονται με την πάροδο του χρόνου, έχοντας έτσι κάποια συσχέτιση σε τιμές χαμηλής καθυστέρησης.

2.3 Πρακτικη εφαρμογή

Το μοντελο είναι lstm regression model που χρησιμοποιεί ώς Lost function to mean square error και σε δευτερο χρόνο χρησιμοποιήσαμε το metrics το MAPE, ξεκιναμε να κανουμε αρχικοποιηση τις παραμέτρους, οι παραμετοι οι οποιοι ξεκινησαμε να παιζουμε και να κανουμε αλλαγές από το αρχικο μας μοντελο είναι οι:

Εποχες(epochs),Dataset split,Lag,Layer depth.

Παραθέτουμε τις τιμες των παραμέτρων και τα αποτελέσματα για κάθε αλλαγή στις παραμέτρους που εχουμε κάνει

2.3.1 lstm 1

Οι αρχικοποιήσεις μας:

Παρακάτω διαφοροποιούμε τις διαφορετικές παραμέτρους του μοντέλου. Για την ακρίβεια επηρρεάζουμε το TRAIN TEST SPLIT το οποίο αποτελεί το ποσοστό με βάση το οποίο διαχωρίζουμε τα δεδομένα σε, δεδομένα εκπαίδευσης(τα δεδομένα εκείνα που χρησιμοποιεί το νευρωνικό δίκτυο για να βελτιώσει τα βάρη με βάση τα οποία ελαχιστοποιεί τον gradient) . Επίσης τροποποιούμε το Lag

(επεξήγηση πιο πάνω στο 3.2). Επίσης τροποποιούμε την μεταβλητή LSTM_LAYER_DEPTH η οποία περιγράφει το πλήθος των ενδιάμεσων Layers του νευρωνικού μας συστήματος. Τέλος δοκιμάζουμε διαφορετικές τιμές για την μεταβλητή EPOCHS η οποία αναφέρεται στο πλήθος τον φορών που θα περάσουν τα δεδομένα εκμάθησης από το νευρωνικό κατά την εκπαίδευση.

Οι διαφορετικές αρχικοποιήσεις είναι οι εξής:

TRAIN TEST SPLIT 0.2

Στο σημείο αυτό ορίζουμε ότι το 80% των δεδομένων χρησιμοποιούνται για να τροφοδοτήσουμε(Feeding) την εκπαίδευση του νευρωνικού μας συστήματος και το 20% για να ελέχξουμε την αποδοτικότητα του νευρωνικού και να κάνουμε την πρόβλεψη (χρησιμοποιούμε τα χρονικά δεδομένα για να πάρουμε τις τιμές της πρόβλεψης και συγκρίνουμε τα αποτελέσματα με τις πραγματικές τιμές με τον δείκτη Mean absolute percentage error [MAPE accuracy])

LAG 1

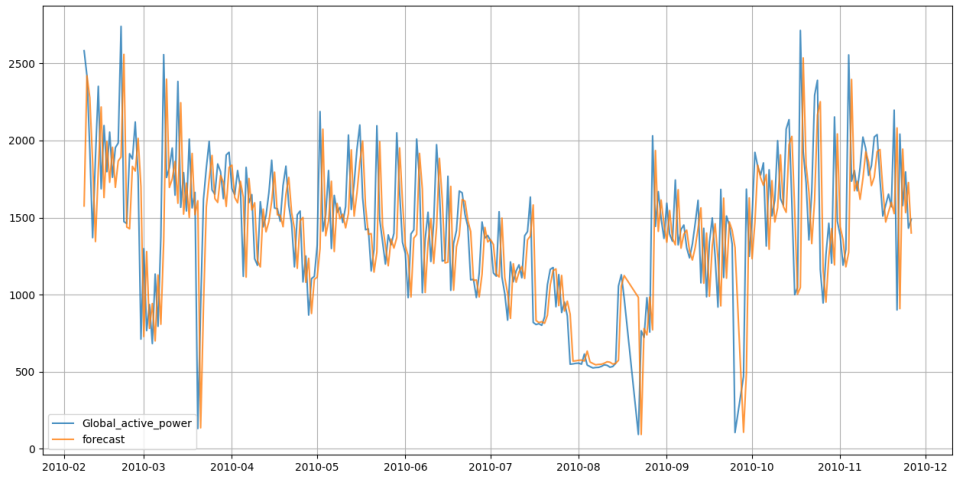
Με βάση την παραπάνω ανάλυση θέτουμε το Lag στο 1 δηλαδή αναζητούμε την αυτοσυσχέτιση της χρονοσειράς σε χρονική μετατόπιση 1 μονάδας (κατά την εξέλιξη της εργασίας μελετήσαμε διαφορετικές χρονικές μονάδες, το παρόν νευρωνικό αφορά χρονική μετατόπιση ανά μέρα)

LSTM_LAYER_DEPTH 128

Η συγκεκριμένη εκδοχή αφορά νευρωνικό με 128 νευρώνες στο επίπεδο (Layer)

EPOCHS 50

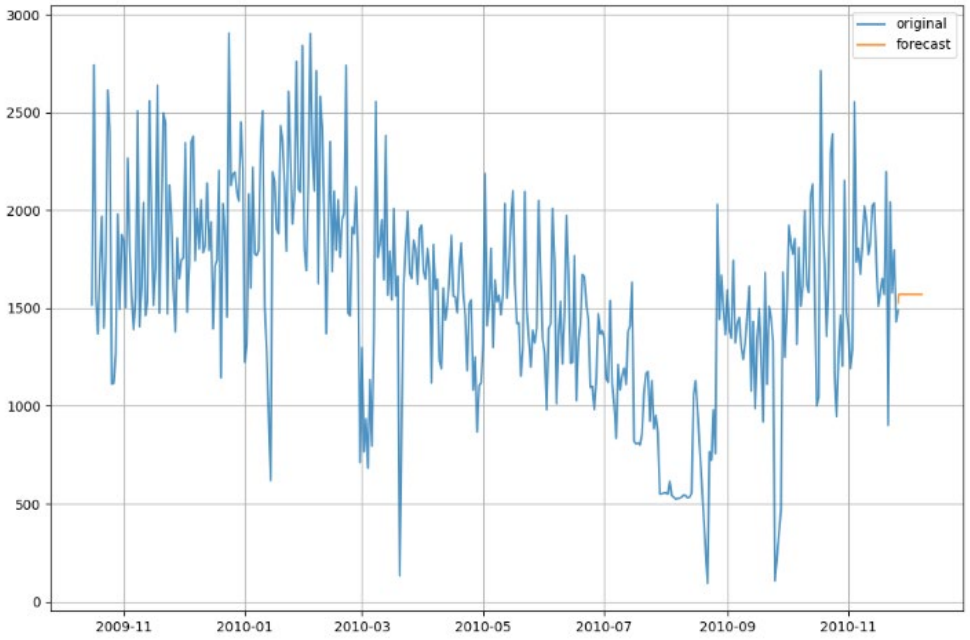
Όπως προαναφέρθηκε η μεταβλητή epochs εκφράζει το πλήθος των φορών που τα δεδομένα θα περάσουν μέσα στο νευρωνικό για εκπαίδευση(πλήθος επανατροφοδοτήσεων) στο παράδειγμα είναι 50.



Εικόνα 14: lstm διαγραμμα train results

Ενώ στο train τα δεδομένα μας και το accuracy φαίνεται να είναι πολύ μεγάλο άλλα όταν τρέχουμε το test dataset split βλέπουμε ότι το accuracy από τα metrics είναι στο 50%

tf.Tensor(49.04687, shape=(), dtype=float32) αυτό μεταφράζετε σε 100-49 =51%



Εικόνα 15:lstm διαγραμμα test results

3.2.2 lstm 1- Denses added

Στο συγκεκριμένο παράδειγμα προσθέσαμε επιπλέον επίπεδα (Layers) νευρώνων τύπου Dense (αποτελούν την τυπική δομή νευρώνα στην βιβλιοθήκη Tensorflow, αποτελούνται από ένα διάνυσμα βάρους [weight] με τυχαίες τιμές kernel weights οι τιμές αυτές τροποποιούνται με βάση το Loss function και μία τιμή bias, αποτελούν την πρωταρχική εκδοχή νευρώνων)

```
lag=1,
LSTM_layer_depth=128,
epochs=50,
train_test_split=0.2
```

Με την προσθήκη Dense με τις ίδιες παραμέτρους .

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 128)               66560
_________________________________________________________________
dense_5 (Dense)              (None, 250)               32250
_________________________________________________________________
dense_6 (Dense)              (None, 125)               31375
_________________________________________________________________
dense_7 (Dense)              (None, 60)                7560
_________________________________________________________________
dense_8 (Dense)              (None, 30)                1830
_________________________________________________________________
dense_9 (Dense)              (None, 1)                 31
=================================================================
Total params: 139,606
Trainable params: 139,606
Non-trainable params: 0
_________________________________________________________________
tf.Tensor(48.02668, shape=(), dtype=float32)
```

tf.Tensor(48.95756, shape=(), dtype=float32) 100-48 = 52%

Ας δοκιμάσουμε να αλλάξουμε λίγο τώρα άλλες παραμέτρους αρχικά ας παίξουμε με το lag.

2.3.2 lstm 2

TRAIN TEST SPLIT 0.2

Η τιμή παραμένει ίδια όπως στο παραπάνω παράδειγμα (βλ. Lstm1)

LAG 10

Με βάση την παραπάνω ανάλυση θέτουμε το Lag στο 10 δηλαδή αναζητούμε την αυτοσυσχέτιση της χρονοσειράς σε χρονική μετατόπιση 10 μονάδων.

LSTM_LAYER_DEPTH 64

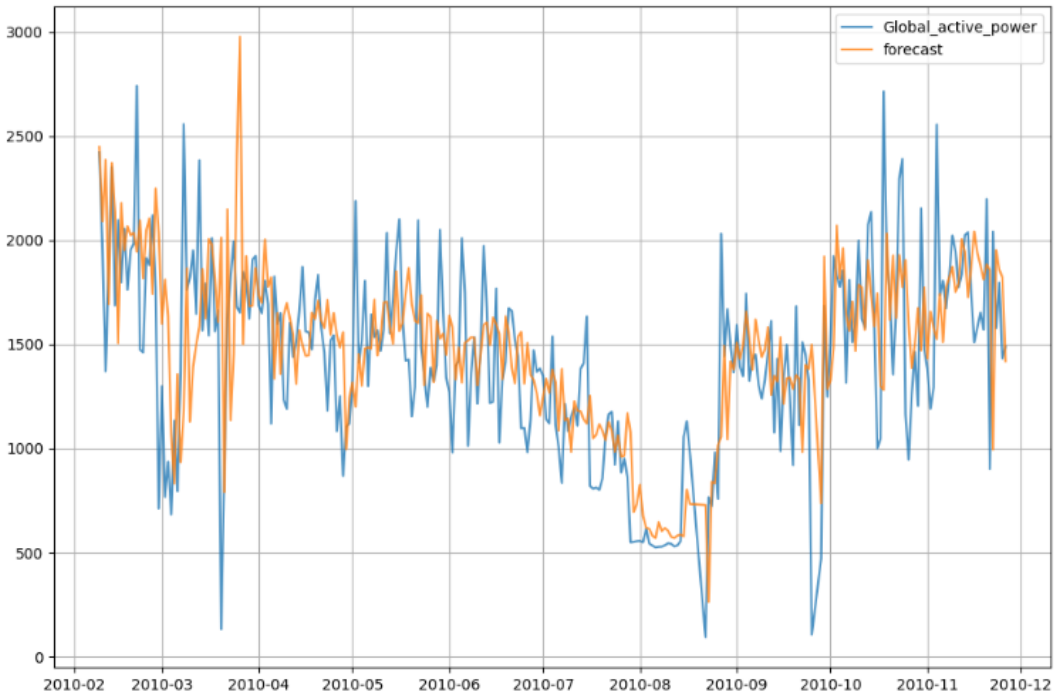
Η συγκεκριμένη εκδοχή αφορά νευρωνικό με 64 νευρώνες στο επίπεδο (Layer)

EPOCHS 350

Όπως προαναφέρθηκε η μεταβλητή epochs εκφράζει το πλήθος των φορών που τα δεδομένα θα περάσουν μέσα στο νευρωνικό για εκπαίδευση(πλήθος επανατροφοδοτήσεων) στο παράδειγμα είναι 350.

Νεα αλλαγή στις παραμέτρους με τις :

```
train_test_split: 0.2
lag: 10
LSTM_layer_depth: 64
epochs: 350
```



Εικόνα 16:lstm διαγραμμα train results

Και το accuracy που πετύχαμε στο test είναι:

tf.Tensor(47.031124, shape=(), dtype=float32) άρα 100-47 = 53%

2.3.3 lstm 3

Στο παράδειγμα αυτό τροποποιούμε την μεταβλητή train_test_split από 0.2 σε 0.3 θέτουμε ότι το 70% των δεδομένων χρησιμοποιούνται για να τροφοδοτήσουμε(Feeding) την εκπαίδευση του νευρωνικού μας συστήματος και το 30% για να ελέχξουμε την αποδοτικότητα του νευρωνικού και να κάνουμε την πρόβλεψη.

TRAIN TEST SPLIT 0.2

Στο σημείο αυτό ορίζουμε ότι το 70% των δεδομένων χρησιμοποιούνται για να τροφοδοτήσουμε(Feeding) την εκπαίδευση του νευρωνικού μας συστήματος και το 30% για να ελέχξουμε την αποδοτικότητα του νευρωνικού και να κάνουμε την πρόβλεψη

LAG 1

Με βάση την παραπάνω ανάλυση θέτουμε το Lag στο 1 δηλαδή αναζητούμε την αυτοσυσχέτιση της χρονοσειράς σε χρονική μετατόπιση 1 μονάδων.

LSTM_LAYER_DEPTH 128

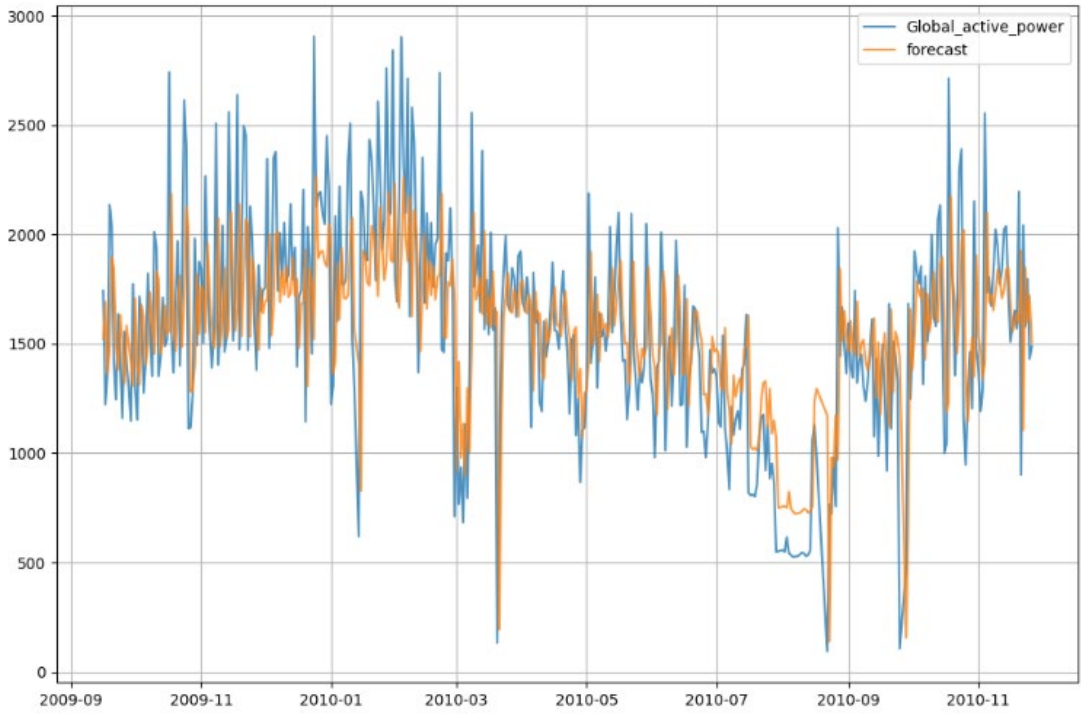
Η συγκεκριμένη εκδοχή αφορά νευρωνικό με 128 νευρώνες στο επίπεδο (Layer)

EPOCHS 350

Όπως προαναφέρθηκε η μεταβλητή epochs εκφράζει το πλήθος των φορών που τα δεδομένα θα περάσουν μέσα στο νευρωνικό για εκπαίδευση(πλήθος επανατροφοδοτήσεων) στο παράδειγμα είναι 350.

```
train_test_split: 0.30
lag: 1
LSTM_layer_depth: 128
epochs: 350
```

ενώ βλέπουμε το train μας δίνει κάλο διάγραμμα



Εικόνα 17:lstm διαγραμμα train results

Παρατηρούμε ότι το αλγεβρικό αποτέλεσμα του δείκτη MAPE μας δίνει accuracy (αποδοτικότητα)

tf.Tensor(41.208466, shape=(), dtype=float32)

άρα 100-41 = 59%

3.2.4 lstm 4

Με μια τελευταία προσπάθεια που κάναμε τροποποιήσαμε τις παρακάτω μεταβλητές ως εξής:

TRAIN TEST SPLIT 0.2

Στο σημείο αυτό ορίζουμε ότι το 80% των δεδομένων χρησιμοποιούνται για να τροφοδοτήσουμε(Feeding) την εκπαίδευση του νευρωνικού μας συστήματος και το 20% για να ελέχξουμε την αποδοτικότητα του νευρωνικού

LAG 10

Με βάση την παραπάνω ανάλυση θέτουμε το Lag στο 10 δηλαδή αναζητούμε την αυτοσυσχέτιση της χρονοσειράς σε χρονική μετατόπιση 10 μονάδων.

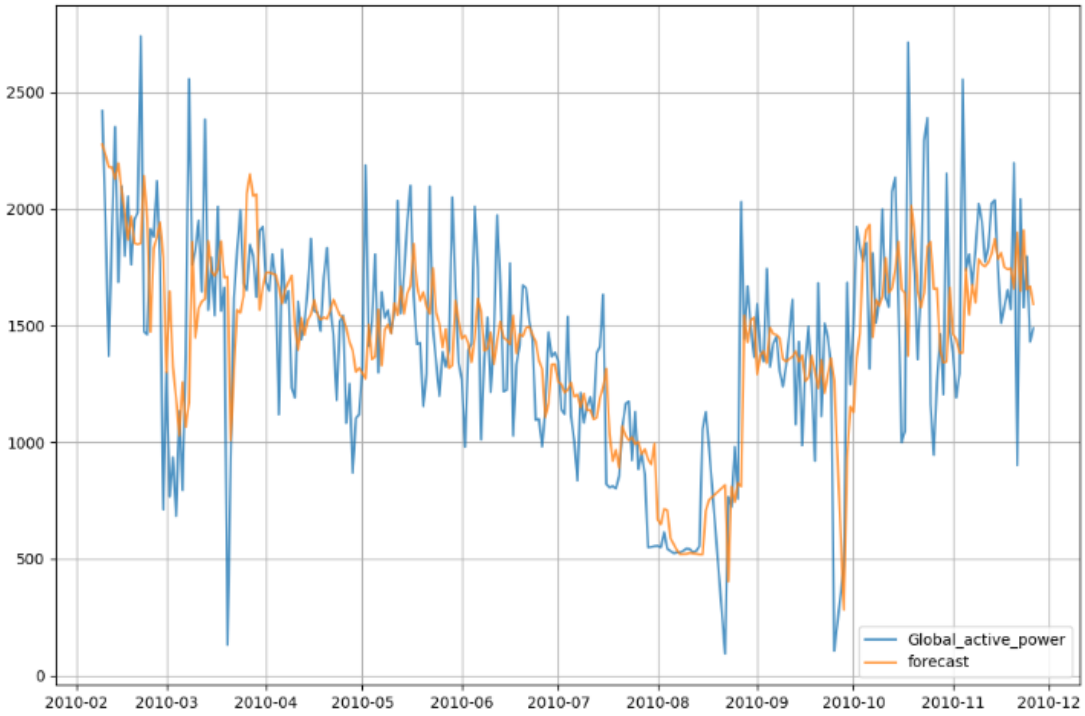
LSTM_LAYER_DEPTH 32

Η συγκεκριμένη εκδοχή αφορά νευρωνικό με 32 νευρώνες στο επίπεδο (Layer)

EPOCHS 50

Όπως προαναφέρθηκε η μεταβλητή epochs εκφράζει το πλήθος των φορών που τα δεδομένα θα περάσουν μέσα στο νευρωνικό για εκπαίδευση(πλήθος επανατροφοδοτήσεων) στο παράδειγμα είναι 50.

```
train_test_split: 0.2
lag: 10
LSTM_layer_depth: 32
epochs: 50
```



Εικόνα 18:lstm διαγραμμα train results

Τα αποτελέσματα που βγαίνουν είναι : tf.Tensor(39.795856, shape=(), dtype=float32) αρα 61%

2.3.5 Αποτελεσματα lstm (days)

Άρα καταλαβαίνουμε ότι πρέπει να βρούμε τις εποχές στο σημείο όπου δεν μας κανει overfeed και το κατάλληλο είναι στις 350 άρα το αφήνουμε εκει στις 350 εποχές. Ακόμα το lag πρέπει να είναι στο 1 γιατι όσο αυξάνουμε το lag είδαμε ότι τα αποτελέσματα μας δεν είναι αυτά που περιμένουμε.

Και αυτό το βλέπουμε σε αυτό το τεστ που κάναμε με αρχικοποιήσεις:

TRAIN TEST SPLIT 0.2

Στο σημείο αυτό ορίζουμε ότι το 70% των δεδομένων χρησιμοποιούνται για να τροφοδοτήσουμε(Feeding) την εκπαίδευση του νευρωνικού μας συστήματος και το 30% για να ελέχξουμε την αποδοτικότητα του νευρωνικού και να κάνουμε την πρόβλεψη

LAG 10

Με βάση την παραπάνω ανάλυση θέτουμε το Lag στο 10 δηλαδή αναζητούμε την αυτοσυσχέτιση της χρονοσειράς σε χρονική μετατόπιση 10 μονάδων.

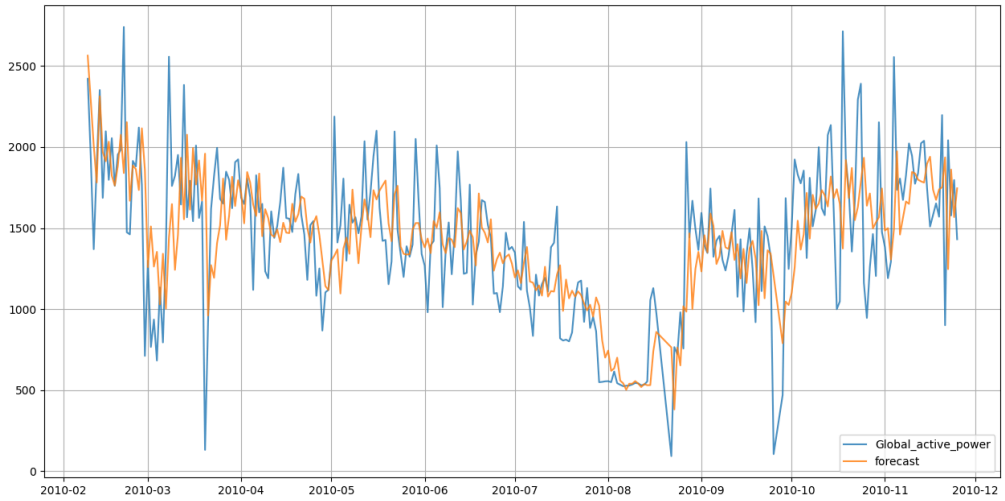
LSTM_LAYER_DEPTH 128

Η συγκεκριμένη εκδοχή αφορά νευρωνικό με 128 νευρώνες στο επίπεδο (Layer)

EPOCHS 350

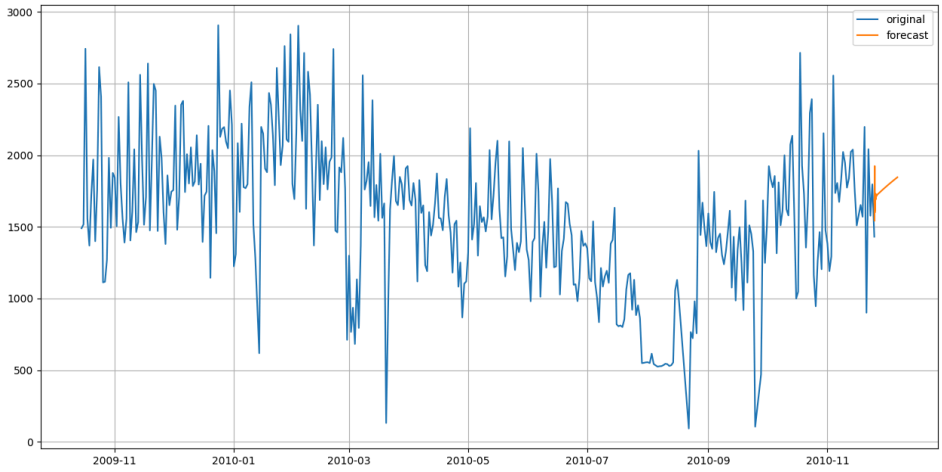
Όπως προαναφέρθηκε η μεταβλητή epochs εκφράζει το πλήθος των φορών που τα δεδομένα θα περάσουν μέσα στο νευρωνικό για εκπαίδευση(πλήθος επανατροφοδοτήσεων) στο παράδειγμα είναι 350.

Κάτωθι βλέπουμε τα αποτελεσματα απο το train του μοντούλου μας και απο το plot είναι υποσχόμενο:



Εικόνα 19: lstm διαγραμμα train results

Αλλά στην συνέχεια απο τα αποτελέσματα που λάβαμε (βλέπουμε και το plot) και απο τα metrics τα αποτελέσματα δεν είναι αυτά που περιμέναμε:



Εικόνα 20:lstm διαγραμμα test results

tf.Tensor(61.46233, shape=(), dtype=float32) άρα ~38.54

2.3.6 lstm months

Η επόμενη μας προσπάθεια θα είναι να κάνουμε grouping σε μήνες για να προσπαθήσουμε να κατασκευάσουμε το ιδιο μοντέλο με τα ίδια χαρακτηριστικά και να το βελτιώσουμε με κατάλληλες αρχικοποιήσεις.

TRAIN TEST SPLIT 0.25

Στο σημείο αυτό ορίζουμε ότι το 75% των δεδομένων χρησιμοποιούνται για να τροφοδοτήσουμε(Feeding) την εκπαίδευση του νευρωνικού μας συστήματος και το 25% για να ελέχξουμε την αποδοτικότητα του νευρωνικού

LAG 1

Με βάση την παραπάνω ανάλυση θέτουμε το Lag στο 1 δηλαδή αναζητούμε την αυτοσυσχέτιση της χρονοσειράς σε χρονική μετατόπιση 1 μονάδων.

LSTM_LAYER_DEPTH 128

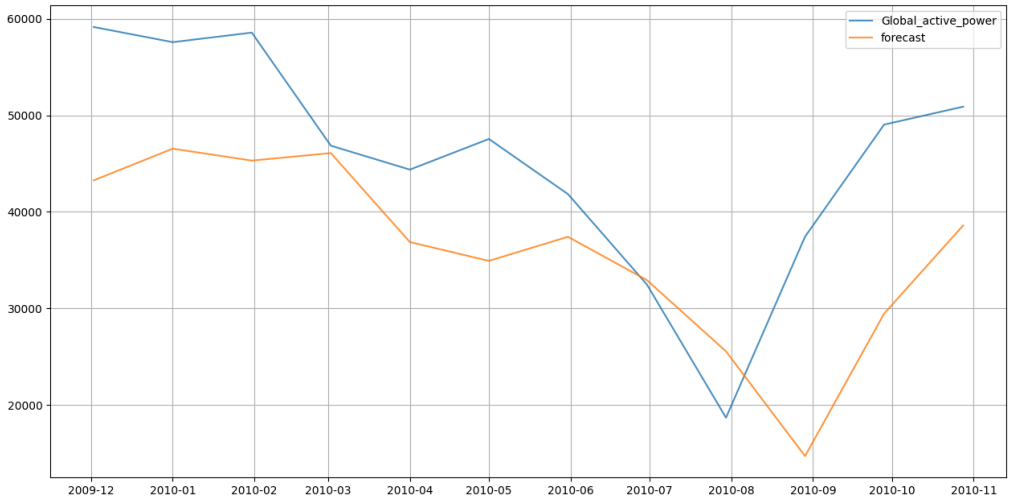
Η συγκεκριμένη εκδοχή αφορά νευρωνικό με 128 νευρώνες στο επίπεδο (Layer)

EPOCHS 300

Όπως προαναφέρθηκε η μεταβλητή epochs εκφράζει το πλήθος των φορών που τα δεδομένα θα περάσουν μέσα στο νευρωνικό για εκπαίδευση(πλήθος επανατροφοδοτήσεων) στο παράδειγμα είναι 50.

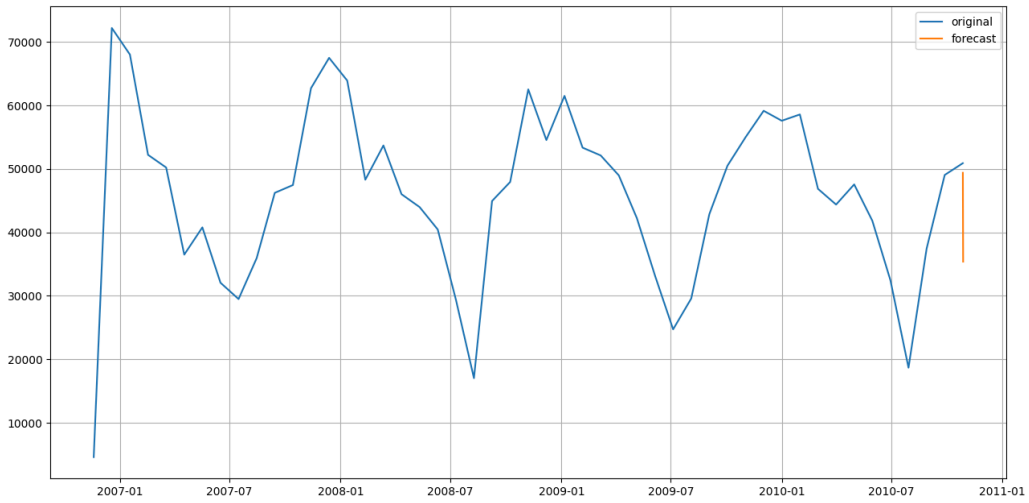
```
train_test_split: 0.25
lag: 1
LSTM_layer_depth: 128
epochs: 300
```

Το διάγραμμα μας της πρόβλεψης τών train δεδομένων μας είναι το ακόλουθες:



Εικόνα 21:lstm διαγραμμα train results

Βλέπουμε στην προσπάθεια μας ότι το μοντέλο μας δίνει ότι δεν ειναι αποδοτικό απο μια πρώτη είκονα που κάναμε Plot και στο επόμενο θα δούμε το διάγραμμα μας με τα αποτελέσματα και το forecast που θα κάνουμε και θα βγάλουμε τις μετρικές μας.



Εικόνα 22:lstm διαγραμμα test results

με βάση τα αποτελέσματα μας βλέπουμε ότι τα αποτελέσματα με grouping σε months tf.Tensor(21.54472, shape=(), dtype=float32) άρα ~ 78.5%

2.3.7lstm months

TRAIN TEST SPLIT 0.25

Στο σημείο αυτό ορίζουμε ότι το 75% των δεδομένων χρησιμοποιούνται για να τροφοδοτήσουμε(Feeding) την εκπαίδευση του νευρωνικού μας συστήματος και το 25% για να ελέχξουμε την αποδοτικότητα του νευρωνικού

LAG 3

Με βάση την παραπάνω ανάλυση θέτουμε το Lag στο 3 δηλαδή αναζητούμε την αυτοσυσχέτιση της χρονοσειράς σε χρονική μετατόπιση 3 μονάδων.

LSTM_LAYER_DEPTH 128

Η συγκεκριμένη εκδοχή αφορά νευρωνικό με 128 νευρώνες στο επίπεδο (Layer)

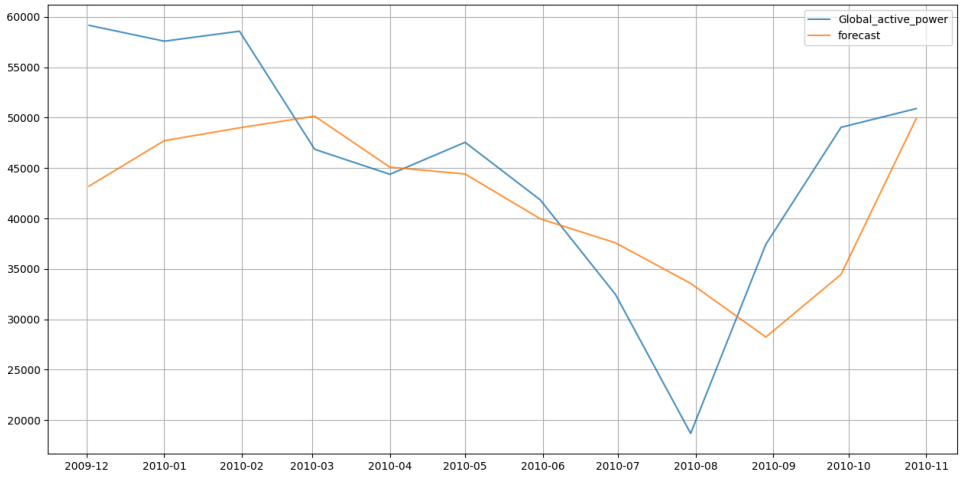
EPOCHS 300

Όπως προαναφέρθηκε η μεταβλητή epochs εκφράζει το πλήθος των φορών που τα δεδομένα θα περάσουν μέσα στο νευρωνικό για εκπαίδευση(πλήθος επανατροφοδοτήσεων) στο παράδειγμα είναι 50.

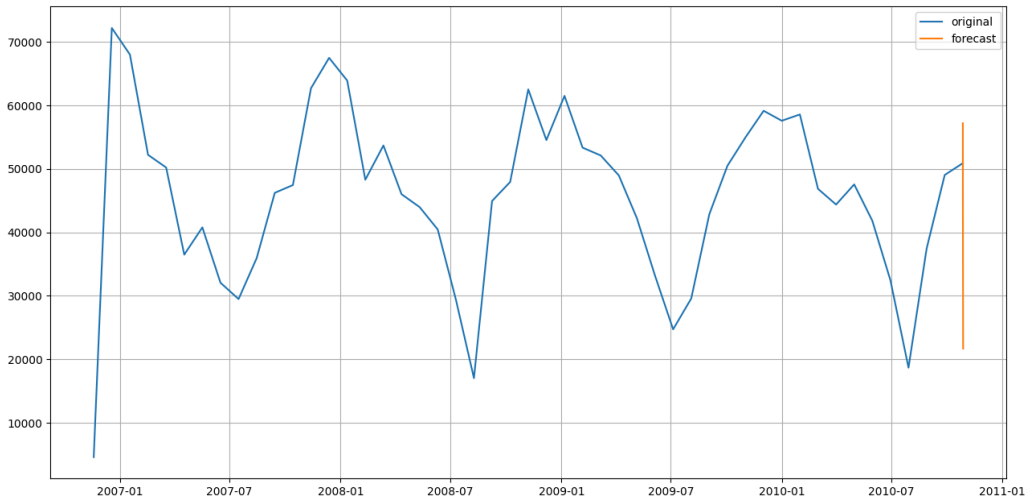
lag=3

```
train_test_split: 0.25
lag: 3
LSTM_layer_depth: 128
epochs: 300
```

Βλέπουμε στην προσπάθεια μας ότι το μοντέλο μας δίνει ότι δεν ειναι αποδοτικό απο μια πρώτη είκονα που κάναμε Plot και στο επόμενο θα δούμε το διάγραμμα μας με τα αποτελέσματα και το forecast που θα κάνουμε και θα βγάλουμε τις μετρικές μας.



Εικόνα 23:lstm διαγραμμα train results



Εικόνα 24:lstm διαγραμμα test results

tf.Tensor(26.419964, shape=(), dtype=float32)άρα ~73,6%

2.3.8lstm months

TRAIN TEST SPLIT 0.25

Στο σημείο αυτό ορίζουμε ότι το 75% των δεδομένων χρησιμοποιούνται για να τροφοδοτήσουμε(Feeding) την εκπαίδευση του νευρωνικού μας συστήματος και το 25% για να ελέχξουμε την αποδοτικότητα του νευρωνικού

LAG 6

Με βάση την παραπάνω ανάλυση θέτουμε το Lag στο 6 δηλαδή αναζητούμε την αυτοσυσχέτιση της χρονοσειράς σε χρονική μετατόπιση 6 μονάδων.

LSTM_LAYER_DEPTH 128

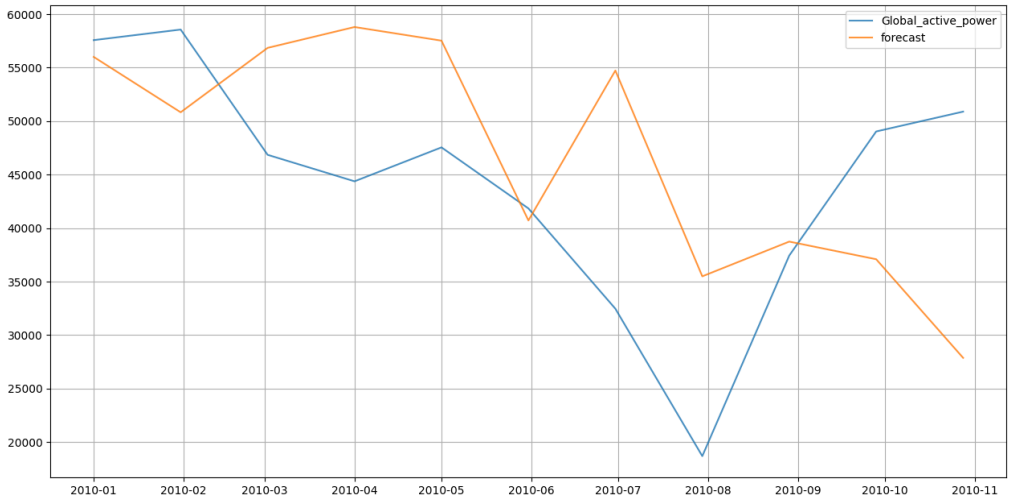
Η συγκεκριμένη εκδοχή αφορά νευρωνικό με 128 νευρώνες στο επίπεδο (Layer)

EPOCHS 300

Όπως προαναφέρθηκε η μεταβλητή epochs εκφράζει το πλήθος των φορών που τα δεδομένα θα περάσουν μέσα στο νευρωνικό για εκπαίδευση(πλήθος επανατροφοδοτήσεων) στο παράδειγμα είναι

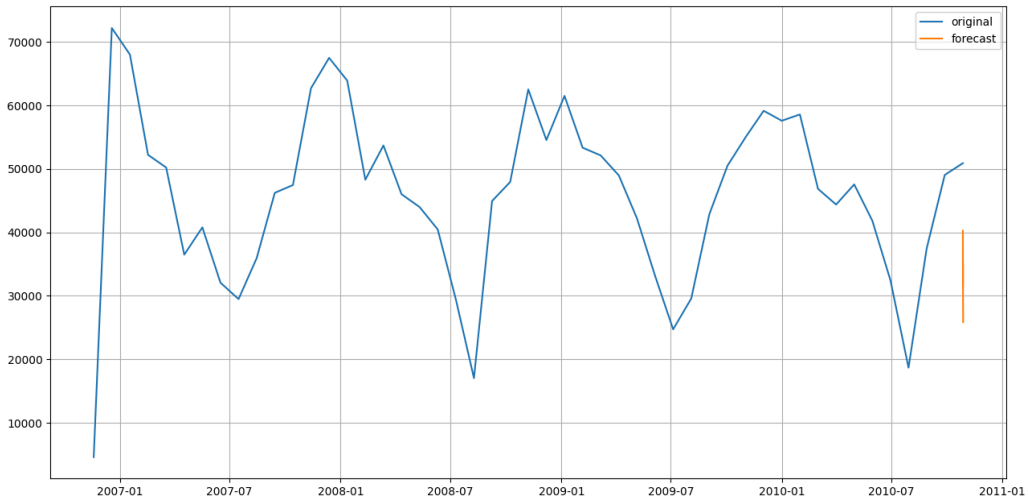
Lag=6

```
train_test_split: 0.25
lag: 6
LSTM_layer_depth: 128
epochs: 300
```



Εικόνα 25:lstm διαγραμμα train results

Στο συγκεκριμένο μοντέλο βλέπουμε μετά τις αλλαγές που κάναμε το διάγραμμα μας του train δεν έχει βελτιωθεί, το αντίθετο είναι χειρότερο άρα βλέπουμε και σε αυτό το παράδειγμα η αλλγή του lag μας δίνει χειρότερα αποτελέσματα. Παμε να δουμε και το τεστ για να μπορούμε να κάνουμε συγκριση.



Εικόνα 26:lstm διαγραμμα test results

tf.Tensor(33.348663, shape=(), dtype=float32) άρα 66,65 και επιβεβαιώνουμε ότι το lag μας πρέπει να ειναι στο 1 και όσο ειναι και η χρονική μας περίοδος και να μην το αλλάξουμε.

2.3.9 lstm seasonal

TRAIN TEST SPLIT 0.25

Στο σημείο αυτό ορίζουμε ότι το 75% των δεδομένων χρησιμοποιούνται για να τροφοδοτήσουμε(Feeding) την εκπαίδευση του νευρωνικού μας συστήματος και το 25% για να ελέχξουμε την αποδοτικότητα του νευρωνικού

LAG 12

Με βάση την παραπάνω ανάλυση θέτουμε το Lag στο 12 δηλαδή αναζητούμε την αυτοσυσχέτιση της χρονοσειράς σε χρονική μετατόπιση 12 μονάδων.

LSTM_LAYER_DEPTH 128

Η συγκεκριμένη εκδοχή αφορά νευρωνικό με 128 νευρώνες στο επίπεδο (Layer)

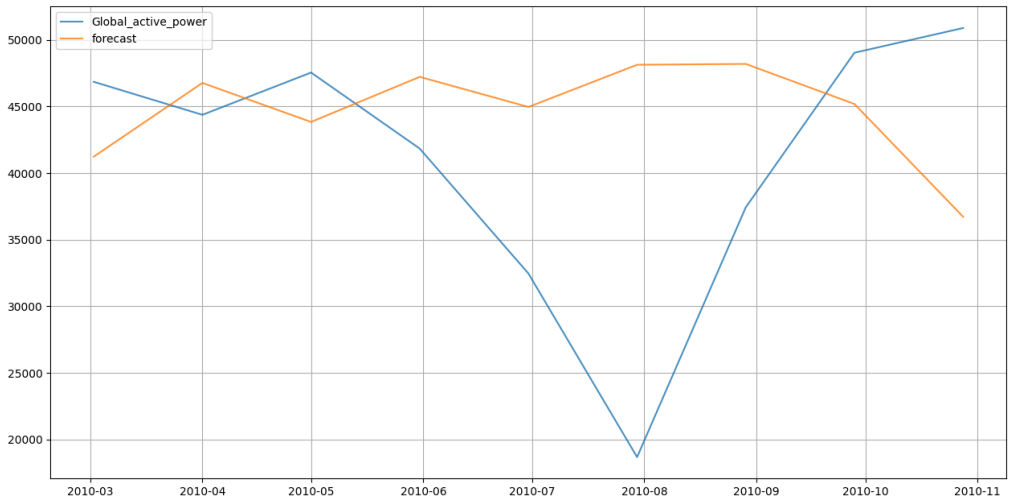
EPOCHS 300

Όπως προαναφέρθηκε η μεταβλητή epochs εκφράζει το πλήθος των φορών που τα δεδομένα θα περάσουν μέσα στο νευρωνικό για εκπαίδευση(πλήθος επανατροφοδοτήσεων) στο παράδειγμα είναι

Lag=12

```
train_test_split: 0.25
lag: 12
LSTM_layer_depth: 128
epochs: 300
```

Πάμε να δούμε το train μας πως θα συμπεριφερθεί τώρα, και βλεπουμε ότι με μεγάλο lag το train forecasting δεν λειτουργεί όπως θα περιμέναμε ή τουλάχιστον να μας δίνει καλα αποτελέσματα , άρα περιμένουμε και το test forecasting να μην είναι αποδοτικό.



Εικόνα 27:lstm διαγραμμα train results

Όπως και έχει γίνει το forecasting μας βλέπουμε ότι το error μας είναι πολύ μεγαλο άρα και το ποσοστό επιτυχια μας ειναι μικρό tf.Tensor(64.45427, shape=(), dtype=float32) άρα 35.5 μη αποδοτικό μοντέλο, παμε να κάνουμε αλλαγές τωρα να δούμε την αποδοση του με διαφορετικό Lag

2.3.10 lstm seasonal

TRAIN TEST SPLIT 0.25

Στο σημείο αυτό ορίζουμε ότι το 75% των δεδομένων χρησιμοποιούνται για να τροφοδοτήσουμε(Feeding) την εκπαίδευση του νευρωνικού μας συστήματος και το 25% για να ελέχξουμε την αποδοτικότητα του νευρωνικού

LAG 12

Με βάση την παραπάνω ανάλυση θέτουμε το Lag στο 12 δηλαδή αναζητούμε την αυτοσυσχέτιση της χρονοσειράς σε χρονική μετατόπιση 12 μονάδων.

LSTM_LAYER_DEPTH 128

Η συγκεκριμένη εκδοχή αφορά νευρωνικό με 128 νευρώνες στο επίπεδο (Layer)

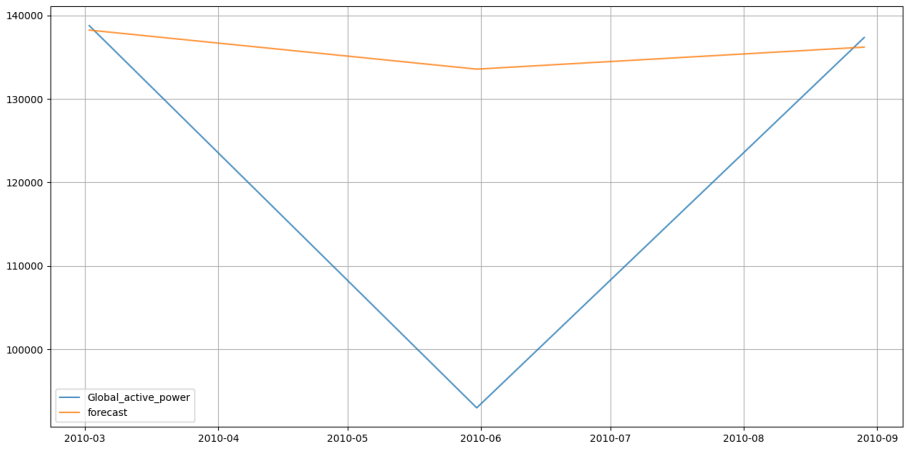
EPOCHS 300

Όπως προαναφέρθηκε η μεταβλητή epochs εκφράζει το πλήθος των φορών που τα δεδομένα θα περάσουν μέσα στο νευρωνικό για εκπαίδευση(πλήθος επανατροφοδοτήσεων) στο παράδειγμα είναι

Lag=1

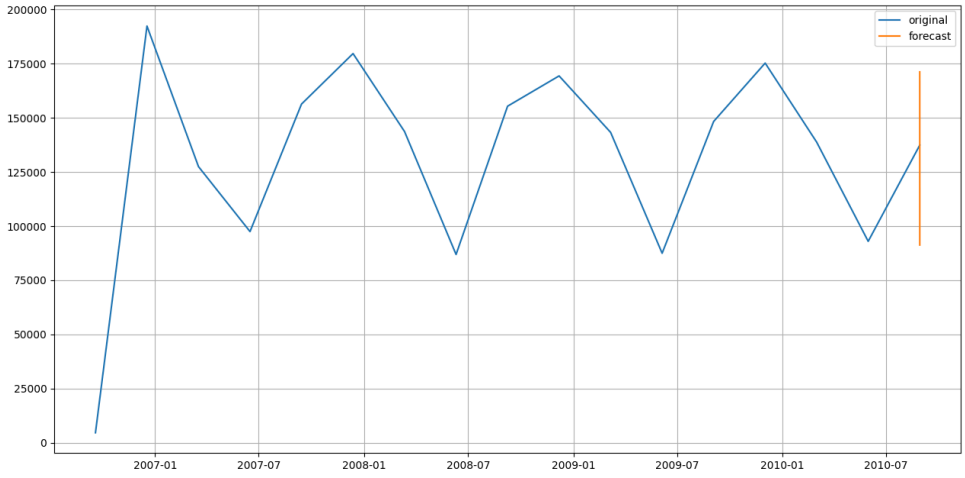
```
train_test_split: 0.25
lag: 1
LSTM_layer_depth: 128
epochs: 300
```

Το plot μας βλεπουμε κάτωθι είναι προβληματικό μας δεν βλέπουμε αναμενόμενα αποτελέσματα στο χρονικό διάστημα που έχουμε κανει plot ενδεικτικά, άλλα πάμε να εξετάσουμε και το test forecasting για να δουμε την απόδοση του μοντέλου μας.



Εικόνα 28:lstm διαγραμμα train results

Το ποσοστό απο τα αποτελέματα μας metrics MAPE είναι tf.Tensor(25.264925, shape=(), dtype=float32) άρα 74.34 είναι πολυ πιο αποδοτικό όταν κάναμε την αλλαγή στο lag όπως και περιμέναμε με βάση και τα προηγούμενα μοντέλα μας όπου είχαμε εξετάσει βάση μηνών grouping data. Αρα πιο αποδοτικό μοντέλο.



Εικόνα 29:lstm διαγραμμα test results

2.3.3 Συμπεράσματα για LTSM

Συνοψίζοντας για το μοντέλο LSTM με βάση τα μοντέλα και τα διαφορετικά testing που κάναμε με διαφορετικά configuration είδα ότι : Καλύτερα αποτελέσματα πήραμε με τον configuration ειναι το 2.3.6 όπου έχουμε κάνει grouping σε Months και το configuration μας ειναι lag=1 ,layers=128 & epochs =300 άρα καταλαβαίνουμε απο τα αποτελέσματα ότι το lag παιζεί πολυ σημαντικό ρόλο στα αποτελέσματα μας ακόμα ισοσταθμισει της εποχες που πρέπει να τρέξει το μοντέλο μας έτσι ώστε να μήν κάνει overfeed όπως και για το layer depth είδαμε βάση πειραμάτων πιο αποδοτικά μοντέλα όταν το depth layer είναι στο 128 σε όλα μας τα πειράματα και tests.

Το seasonality μας θα πρέπει να κάνουμε πιο εκτενέστερη και σε βάθος ανάλυση. Έχουμε προσθέσει στην θεωρία μας σχετικά με την εποχικότητα και στα αποτελέσματα μας δεν βρήκαμε κάποια συσχέτηση άρα θα πρέπει να περάσουμε περισσότερο χρόνο αναλύοντας άλλα πηραμε αρκετα καλά αποτελέσματα όταν κάναμε grouping σε seasons.

3.Fbprophet

3.1 Γιατί το Facebook Prophet;

Το Facebook ανέπτυξε ένα open source Prophet, ένα εργαλείο πρόβλεψης διαθέσιμο τόσο σε Python όσο και σε R. Παρέχει έξυπνες παραμέτρους που είναι εύκολο να συντονιστούν. Ακόμη και κάποιος που δεν έχει βαθιά εξειδίκευση στα μοντέλα πρόβλεψης χρονοσειρών μπορεί να το χρησιμοποιήσει για να δημιουργήσει ουσιαστικές προβλέψεις για μια ποικιλία προβλημάτων σε επιχειρηματικά σενάρια και όχι μονο.

Η παραγωγή προβλέψεων υψηλής ποιότητας δεν είναι εύκολο πρόβλημα ούτε για τις μηχανές ούτε για τους περισσότερους αναλυτές. Παρατηρήσαμε δύο βασικά θέματα στην πρακτική δημιουργίας ποικίλων επιχειρηματικών προβλέψεων:

- Οι εντελώς αυτόματες τεχνικές πρόβλεψης μπορεί να είναι εύθραυστες και συχνά είναι πολύ άκαμπτες για να ενσωματώσουν χρήσιμες υποθέσεις ή ευρετικές μεθόδους.
- Οι αναλυτές που μπορούν να παράγουν προβλέψεις υψηλής ποιότητας είναι αρκετά σπάνιοι επειδή η πρόβλεψη είναι μια εξειδικευμένη δεξιότητα επιστήμης δεδομένων που απαιτεί σημαντική εμπειρία.

Το Fbprophet χρησιμοποιεί ένα μοντέλο αποσυνθέσιμης χρονοσειράς με τρία κύρια στοιχεία του μοντέλου: trend, seasonality, and holidays. Συνδυάζονται στην ακόλουθη εξίσωση:

$$y(t)= g(t) + s(t) + h(t) + \varepsilon t$$

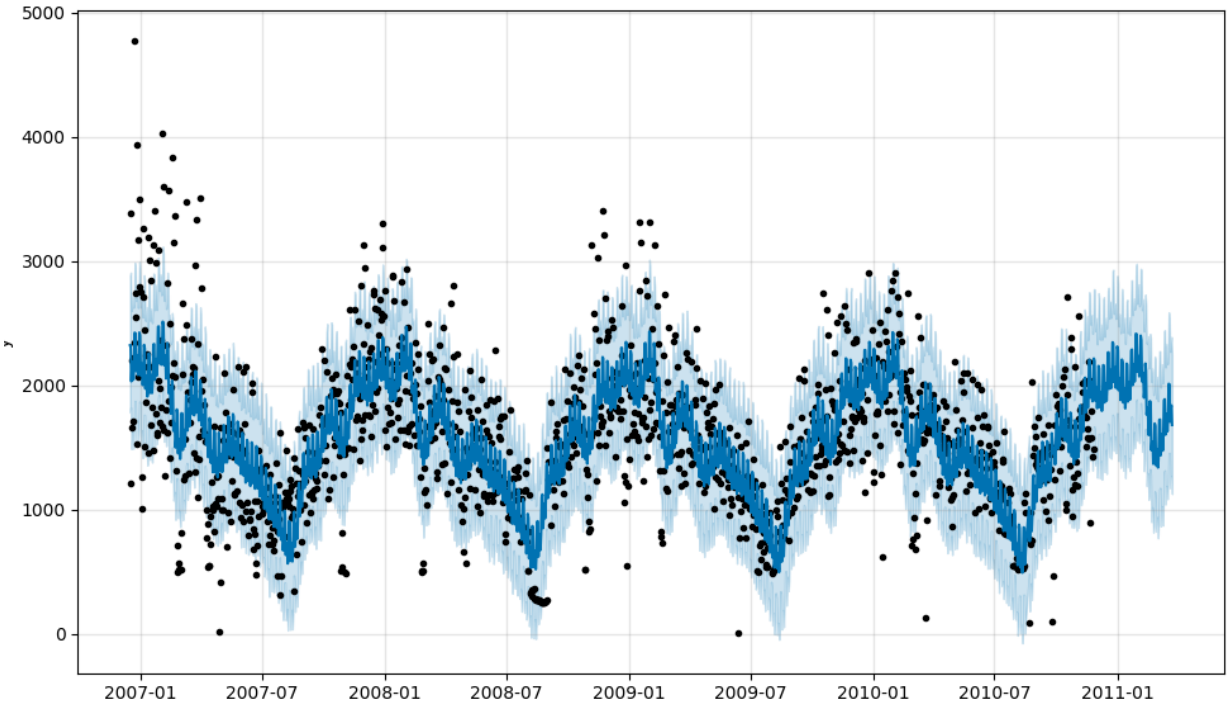
- g(t): τμηματικά γραμμική ή λογιστική καμπύλη ανάπτυξης για μοντελοποίηση μη περιοδικών αλλαγών σε χρονοσειρές
- s(t): περιοδικές αλλαγές (π.χ. εβδομαδιαία/ετήσια εποχικότητα)
- h(t): επιπτώσεις των αργιών (παρέχεται από τον χρήστη) με ακανόνιστα δρομολόγια
- εt: ο όρος σφάλματος λαμβάνει υπόψη τυχόν ασυνήθιστες αλλαγές που δεν περιλαμβάνονται στο μοντέλο

Χρησιμοποιώντας το χρόνο ως οπισθοδρόμηση, ο Fbprophet προσπαθεί να προσαρμόσει πολλές γραμμικές και μη γραμμικές συναρτήσεις του χρόνου ως συνιστώσες. Η μοντελοποίηση της εποχικότητας ως πρόσθετου συστατικού είναι η ίδια προσέγγιση που ακολουθείται από την εκθετική εξομάλυνση στην τεχνική Holt-Winters. Ο Fbprophet πλαισιώνει το πρόβλημα της πρόβλεψης ως άσκηση προσαρμογής καμπύλης αντί να εξετάζει ρητά τη χρονική εξάρτηση κάθε παρατήρησης μέσα σε μια χρονολογική σειρά.

Στην συνέχεια θα αναλύσουμε και θα τρέξουμε το μοντέλο μας με διαφορετικές ρυθμίσεις και με διαφορετικά grouping για τα δεδομένα μας. Δλδ θα σπάσουμε σε days-months-seasons-raw data έτσι ώστε να κάνουμε μια σύγκριση μεταξύ του ίδου μοντέλου και των υπόλοιπον μοντέλουν που έχουμε δημιουργήσει στα προηγουμενα\επόμενα κεφάλαια .

3.2 Fbprophet per day

Τα δεδομένα μας σε αυτή την περίπτωση είναι grouping με βάση την ημέρα. Αρα έχουμε κανει grouping κάθε μία μερα τα δεδομένα και τα εξετάζουμε per day. Στην εικόνα που ακολουθεί μπορούμε να δούμε τα δεδομένα μας τυπωμένα και βλέπουμε και το forecasting.



Εικόνα 30: Data plot diagram (days)

Οι κώδικες υπάρχουν σε μορφή py per_day. Οι ρυθμίσεις που έχουμε βάλει για αυτό το μοντέλο μας είναι οι ακόλουθες:

Per day dataset

```
dataset = pd.read_csv(r'C:\Users\antreas\PycharmProjects\ML\dataset_by_day.csv',
                      usecols=['dayofyear','Global_active_power'])
```

period 1 day:

```
df_cross_val=cross_validation(model,initial=str(round(len(dataset.index[:])*0.7))+ " days", period='1
days', horizon=str(round(len(dataset.index[:])*0.3)+1)+" days")
```

Αποτελέσματα μοντέλου:

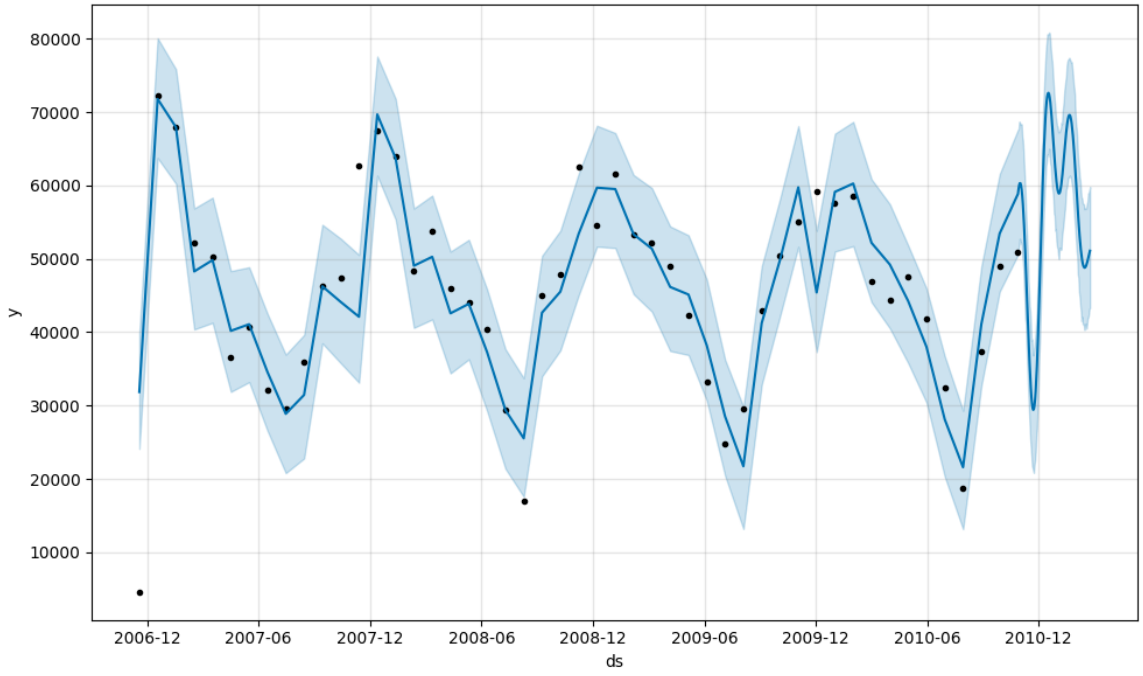
```
   horizon          mse       rmse  ...      mape     mdape  coverage
0   43 days  118123.521815  343.691027  ...  0.163750  0.146060  0.914454
1   44 days  120292.449727  346.832019  ...  0.164997  0.146060  0.908555
2   45 days  122588.470616  350.126364  ...  0.165994  0.144778  0.902655
3   46 days  125350.586018  354.048847  ...  0.167420  0.144778  0.896755
4   47 days  126377.839183  355.496609  ...  0.167780  0.142848  0.893805
..      ...           ...         ...  ...       ...       ...       ...
```

```
384 427 days  194734.519134  441.287343  ...  0.203880  0.141367  0.858407
385 428 days  203621.288575  451.244156  ...  0.210046  0.141843  0.852507
386 429 days  213588.830700  462.156717  ...  0.217133  0.144156  0.846608
387 430 days  223050.433341  472.282154  ...  0.223716  0.144187  0.840708
388 431 days  232884.878077  482.581473  ...  0.230788  0.145058  0.831858

MAPE: 0.29460260593486334
```

Αρα από τα αποτελέματα μας βλέπουμε ότι το μοντέλο μας έχει έχει ποσοστό επιτυχίας 71% στο forecasting. Από τα αποτελέσματα που έχουμε τυπώσει βλέπουμε ότι σε πολλές περιπτώσεις φτάνει κοντά στο φτάνει κοντά στο 84%.αλλα στο σύνολο είναι σε αυτό που αναφέραμε στο 71%.

3.3 Fbprophet per month

Τα δεδομένα μας σε αυτή την περίπτωση είναι χωρισμένα σε μήνες. Άρα έχουμε κάνει grouping όλες την ήμερες και λεπτά σε μήνες για να εξετάσουμε per_month. Στην εικόνα που ακολουθεί μπορούμε να δουμε τα δεδομένα μας τυπωμένα και το forecasting



Εικόνα 31: Data plot diagram (months)

Οι κώδικες υπάρχουν σε μορφή py per_month. Οι ρυθμίσεις που έχουμε βάλει για αυτό το μοντέλο μας είναι οι ακόλουθες:

Per month dataset:

```
dataset = pd.read_csv(r'C:\Users\antreas\PycharmProjects\ML\dataset_by_day.csv',
                      usecols=['dayofyear','Global_active_power'])
```

Period 30days:

```
df_cross_val=cross_validation(model,initial=str(round(len(dataset.index[:])*0.7))+ " days", period='30
days',
                              horizon=str(round(len(dataset.index[:])*0.3)+1)+" days")
```

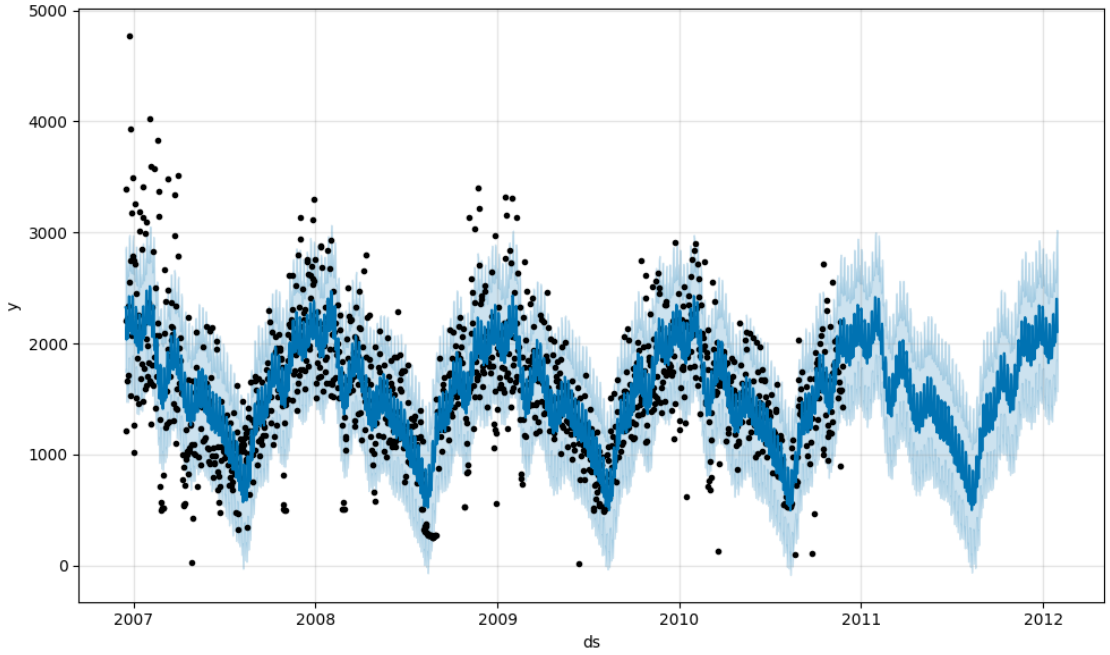
Αποτελέσματα:

```
   horizon          mse       rmse  ...      mape     mdape  coverage
0    42 days  118726.511365  344.567136  ...  0.167138  0.153239  0.904762
1    43 days  119236.331428  345.306142  ...  0.168288  0.153239  0.904762
2    44 days  132282.704441  363.706894  ...  0.172073  0.153239  0.880952
3    45 days  133718.828635  365.675852  ...  0.173626  0.159696  0.880952
4    46 days  134275.258192  366.435886  ...  0.174800  0.159696  0.880952
..       ...            ...         ...  ...       ...       ...       ...
378 427 days  254586.132981  504.565291  ...  0.250299  0.152929  0.785714
379 428 days  254661.837240  504.640305  ...  0.250362  0.152929  0.785714
380 429 days  264819.101138  514.605773  ...  0.257307  0.161253  0.761905
381 430 days  267981.063556  517.668874  ...  0.260159  0.168932  0.761905
382 431 days  271834.856224  521.377844  ...  0.265316  0.168932  0.738095

MAPE: 0.303545054768725
```

Αρα από τα αποτελέματα μας βλέπουμε ότι το μοντέλο μας έχει έχει ποσοστό επιτυχίας 70% στο forecasting. Από τα αποτελέσματα που έχουμε τυπώσει βλέπουμε ότι σε πολλές περιπτώσεις φτάνει κοντά στο φτάνει κοντά στο 84%.αλλα στο σύνολο είναι σε αυτό που αναφέραμε στο 70%. Δεν βλέπουμε να έχει μεγάλη διαφορά το shorting day σε σύγκρισή με το months άρα δεν βλέπουμε ότι κανει κάποια μεγάλη μεταβολή στο μοντέλο μας όταν από day forecasting πάμε se month forecasting.Να δούμε και σε years αν και τα δεδομένα που έχουμε δειν είναι πάρα πολλά. Περισσότερο για να κάνουμε συγκριση και ψάχνουμε να βρούμε seasonality ή σε κάποιο time period όπου τα δεδομένα μας μας δίνουν καλύτερα αποτελέσματα.

3.4 Fbprophet per year

Όπως είπαμε και πριν το επόμενο μοντέλο μας θα δεχτεί δεδομένα χωρισμένα σε έτη. Άρα έχουμε κάνει grouping τους μήνες, τις ήμερες και λεπτά σε χρονολογικά έτη για να εξετάσουμε per_year. Στην εικόνα που ακολουθεί μπορούμε να δούμε τα δεδομένα μας τυπωμένα και το forecasting:



Εικόνα 32:Data plot diagram (years)

Οι κώδικες υπάρχουν σε μορφή py per_year. Οι ρυθμίσεις που έχουμε βάλει για αυτό το μοντέλο μας είναι οι ακόλουθες:

Per year dataset:

```
dataset = pd.read_csv(r'C:\Users\antreas\PycharmProjects\ML\dataset_by_day.csv',
                      usecols=['dayofyear','Global_active_power'])
```

Period 365days:

```
df_cross_val=cross_validation(model,initial=str(round(len(dataset.index[:])*0.7))+ " days", period='365
days',
                              horizon=str(round(len(dataset.index[:])*0.3)+1)+" days")
```

Αποτελέσματα:

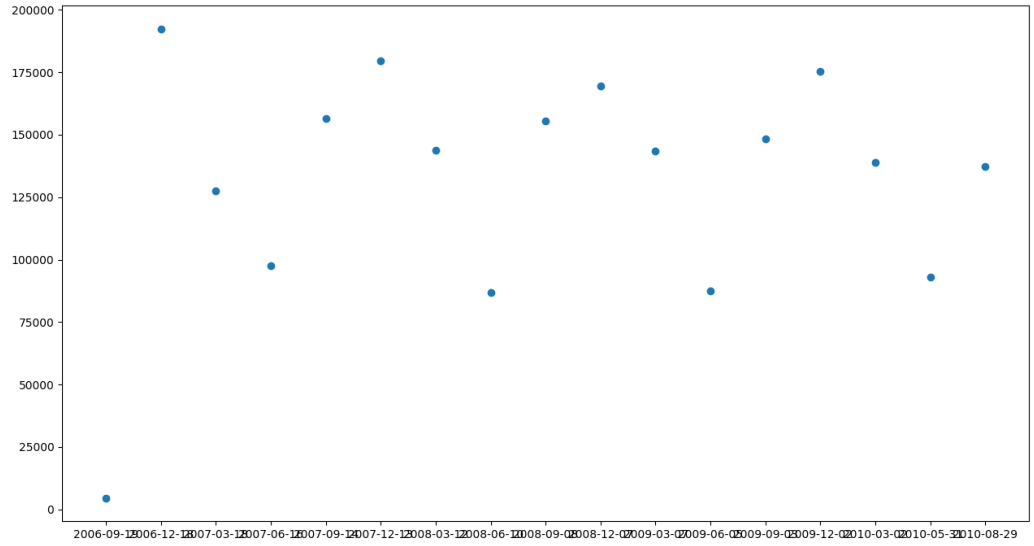
```
     horizon          mse       rmse  ...      mape     mdape  coverage
0   42 days  118726.511365  344.567136  ...  0.167138  0.153239  0.904762
1   43 days  119236.331428  345.306142  ...  0.168288  0.153239  0.904762
2   44 days  132282.704441  363.706894  ...  0.172073  0.153239  0.880952
3   45 days  133718.828635  365.675852  ...  0.173626  0.159696  0.880952
4   46 days  134275.258192  366.435886  ...  0.174800  0.159696  0.880952
..      ...           ...         ...  ...       ...       ...       ...
378 427 days  254586.132981  504.565291  ...  0.250299  0.152929  0.809524
```

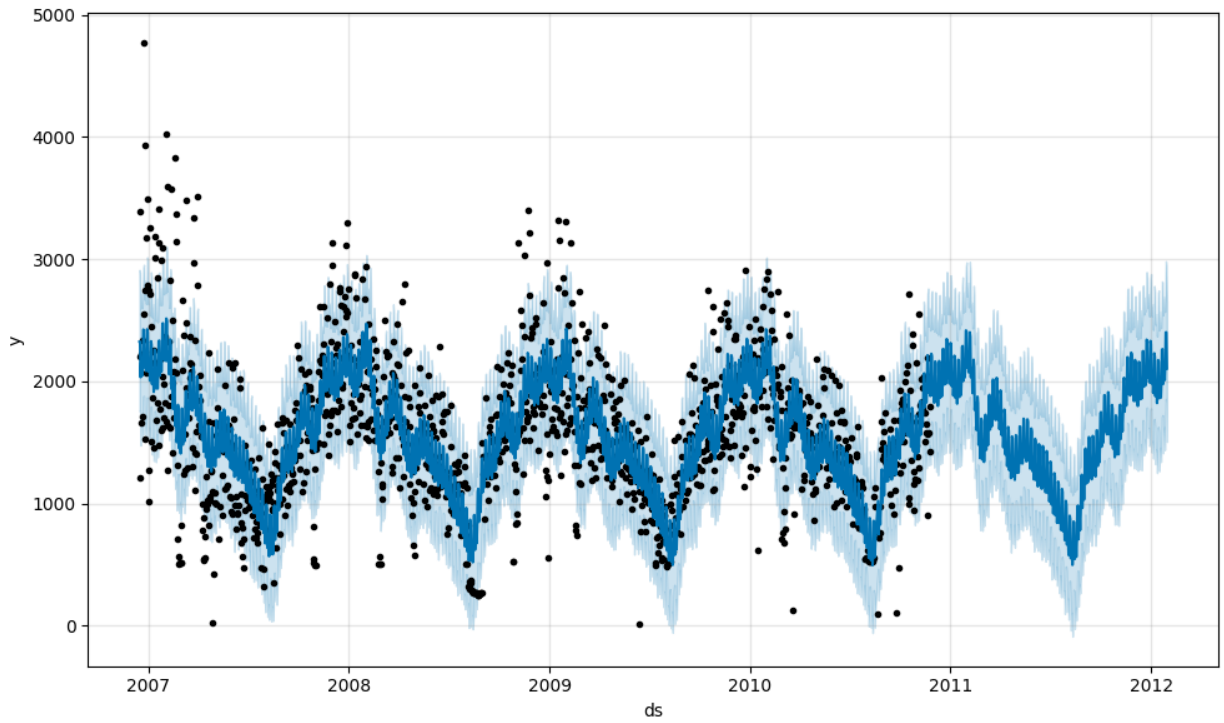
```
379 428 days  254661.837240  504.640305  ...  0.250362  0.152929  0.809524
380 429 days  264819.101138  514.605773  ...  0.257307  0.161253  0.785714
381 430 days  267981.063556  517.668874  ...  0.260159  0.168932  0.785714
382 431 days  271834.856224  521.377844  ...  0.265316  0.168932  0.761905

MAPE: 0.303545054768725
```

Τα αποτελέσματα μας με στρογγυλοποίηση βλέπουμε ότι δεν διαφέρουν από το per_month . Το αποτέλεσμα είναι στο 70%. Άρα από τα αποτελέσματα μας καταλαβαίνουμε ότι δεν βοηθάει όταν κάνουμε grouping στα δεδομένα μας. Λαβαμε κάλα αποτελέσματα στο grouparισμα που κάναμε ανα day. Λογο αυτών τον αποτελεσμάτων θα εξετάσουμε και για seasonality να δούμε πως θα συμπεριφερθεί το μοντέλο μας αλλα και στα raw data. Σε αυτό το σημειο αποφασίσαμε να παίξουμε με τα raw data γιατι είδαμε ότι μας φέρνει καλύτερα αποτελέσματα όταν χρησιμοποιούμε πιθανότατα μεγαλύτερο όγκο δεδομένων και όχι λόγο day-month. Αυτό είναι κατι που θα το εξετάσουμε στα επόμενα μοντελα που θα φτιάξουμε και θα ξεκαθαρίσουμε καλύτερα τι συμβαίνει με τα δεδομένα μας.

3.5 Fbprophet seasonality

Δεδομένου τον προηγούμενων αποτελεσμάτων που μας έχουν δώσει τα μοντέλα μας δεν περιμένουμε καλα αποτελέσματα. Στο seasonality ουσιαστικά κάνουμε grouping ανα 3 μήνες. Κάτωθι βλέπουμε πιλοταρισμένα τα δεδομένα μας και στην επόμενη εικόνα βλέπουμε πιλοταρισμένα τα δεδομένα μας και το forecasting.





Εικόνα 33: Data plot diagram (years-seasonality)

Οι κώδικες υπάρχουν σε μορφή py per_season. Οι ρυθμίσεις που έχουμε βάλει για αυτό το μοντέλο μας είναι οι ακόλουθες:

Per season dataset:

```
dataset = pd.read_csv(r'C:\Users\antreas\PycharmProjects\ML\dataset_by_day.csv',
                      usecols=['dayofyear','Global_active_power'])
```

Period 90days:

```
df_cross_val=cross_validation(model,initial=str(round(len(dataset.index[:])*0.7))+ " days", period='90
days',
                              horizon=str(round(len(dataset.index[:])*0.3)+1)+" days")
```

Αποτελέσματα:

```
   horizon          mse        rmse  ...      mape     mdape  coverage
0   42 days  118726.511365  344.567136  ...  0.167138  0.153239  0.904762
1   43 days  119236.331428  345.306142  ...  0.168288  0.153239  0.904762
```

```
2   44 days  132282.704441  363.706894  ...  0.172073  0.153239  0.880952
3   45 days  133718.828635  365.675852  ...  0.173626  0.159696  0.880952
4   46 days  134275.258192  366.435886  ...  0.174800  0.159696  0.880952
..      ...          ...         ...  ...       ...       ...       ...
378 427 days  254586.132981  504.565291  ...  0.250299  0.152929  0.785714
379 428 days  254661.837240  504.640305  ...  0.250362  0.152929  0.785714
380 429 days  264819.101138  514.605773  ...  0.257307  0.161253  0.761905
381 430 days  267981.063556  517.668874  ...  0.260159  0.168932  0.761905
382 431 days  271834.856224  521.377844  ...  0.265316  0.168932  0.738095

MAPE:0.303545054768725
```

Άρα από τα αποτελέσματα μας βλέπουμε αυτό που ακριβός περιμέναμε καμία διαφορά με τα προηγούμενα αρα δεν έχουμε κάτι να σχολιάσουμε, δεν έχουμε καταφέρει να βρούμε seasonality στα δεδομένα μας. Μια σκέψη μας είναι ότι το seasonality μπορεί να χωρίζεται σε day-night ή ακόμα καλύτερα σε διαστήματα πρωί\μεσημέρι\απόγευμα\βράδυ. Άρα πάμε να εξετάσουμε στα raw data μας το μοντέλο για να δούμε συγκριτικά την συμπεριφορά του.

3.6 Fbprophete raw data

Οι κώδικες υπάρχουν σε μορφή py original_data. Οι ρυθμίσεις που έχουμε βάλει για αυτό το μοντέλο μας είναι οι ακόλουθες:

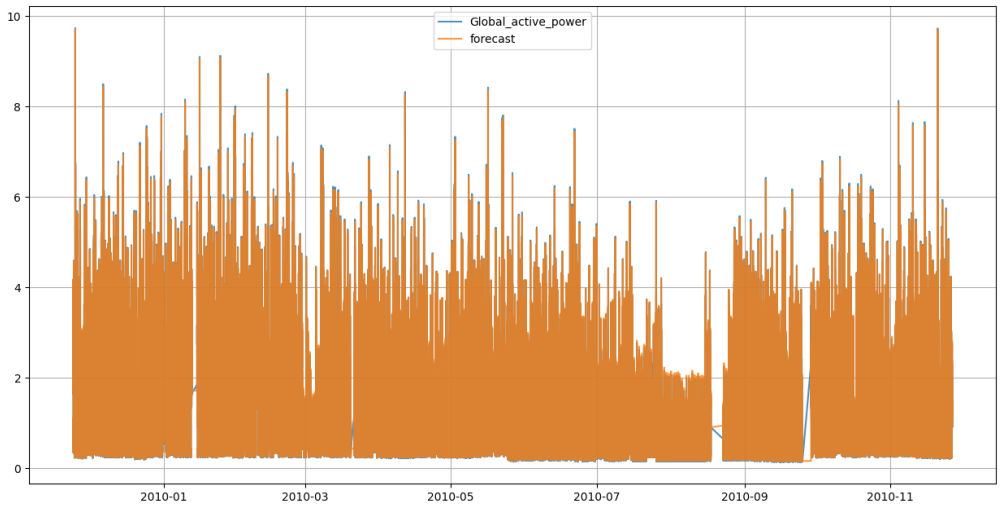
Raw data dataset:

```
dataset =
pd.read_csv(r'C:\Users\antreas\PycharmProjects\ML\household_power_consumption_Global_active_
power.csv',
                usecols=['datetime','Global_active_power'])
```

Period 1hour:

```
df_cross_val=cross_validation(model,initial=str(round(len(dataset.index[:])*0.7))+ " days", period='',
                      horizon=str(round(len(dataset.index[:])*0.3)+1)+" days")
```

Αποτελέσματα:



Εικόνα 34: Raw data diagram

Το μοντέλο μας μας δίνει plot αλλα δεν τελειώνει ποτέ τα raw data χωρις καμία επεξεργασία στο fbprohet δεν είναι αποδοτικά και δεν παίρνουμε αποτέλεσμα μονο το train plot μας όπου βλέπουμε ότι τα αποτελέσματα μας του train forecast είναι πολύ κοντα στο data set μας άλλα δεν μας βγάζει κάποιο αποτέλεσμα MAPE. Θα πρέπει να εξετάσμουμε σε μεγαλύτερο βάθος γιατι δεν μας λειτουργεί μεχρι τον βαθμό όπου προσπαθήσαμε δεν καταφέραμε κάτι και να το κάνουμε να λειτουργήσει και με βάση το documentation απο το official site και απο βοηθητικά articles που βηρήκαμε στο διαδίκτυο και με δικές μας ενέργειες

3.7 Συμπεράσματα

Στα μοντελα FBprophet δεν καταφέραμε να πετύχουμε MAPE καλύτερο απο τα προηγούμενα μοντέλα μας απο ότι στο lstm το FBprophet χρειάζετε επεξεργασία στα δεδομένα μας πρίν τα βάλουμε μεσα στο μοντέλο μας διότι ένα πείραμα με τα raw data που καναμε δεν μας έδωσε αποτελέσματα. Τα υπόλοιπα μοντέλα μας στο fbprophet μας έδωσαν αποτελέσματα και το πιο αποδοτικό ειναι το grouping per day που δημιουργήσαμε. Μπορειτε να δείτε αναλυτικά τι έχουμε κάνει στο κεφάλαιο και στον κώδικα μας πιο αποδοτικά μοντέλα ειναι βαση 1 ημέρας καταωάλωσης αν και τα υπόλοιπα μοντέλα per month , year δεν μας έβλαλαν πολυ χειρότερα αποτελέσματα αλλα το πιο αποδοτικό ειναι per day.

4.WeKa

Επόμενη σκέψη μας για να δούμε γρήγορα αποτελέσματα είναι να χρησιμοποιήσουμε το Weka.

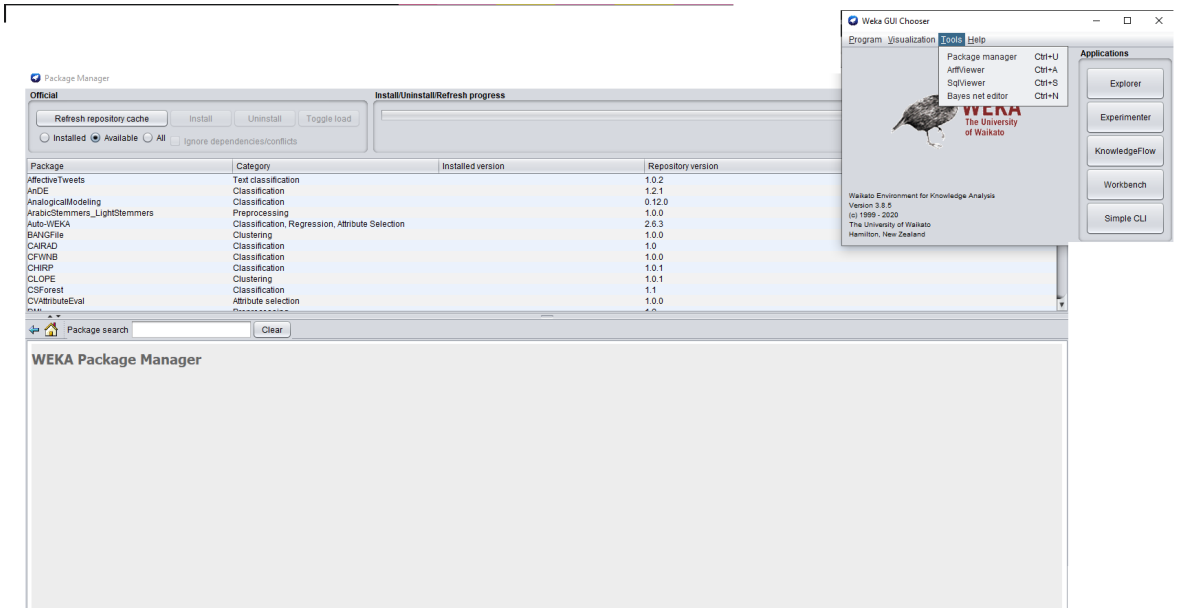
4.1Τι είναι το WeKa?

Είαι ένα λογισμικό ανοιχτού κώδικα παρέχει εργαλεία για προεπεξεργασία δεδομένων, εφαρμογή πολλών αλγορίθμων μηχανικής μάθησης και εργαλεία απεικόνισης, ώστε να μπορείτε να αναπτύξετε τεχνικές μηχανικής εκμάθησης και να τις εφαρμόσετε σε προβλήματα εξόρυξης δεδομένων σε πραγματικό κόσμο.

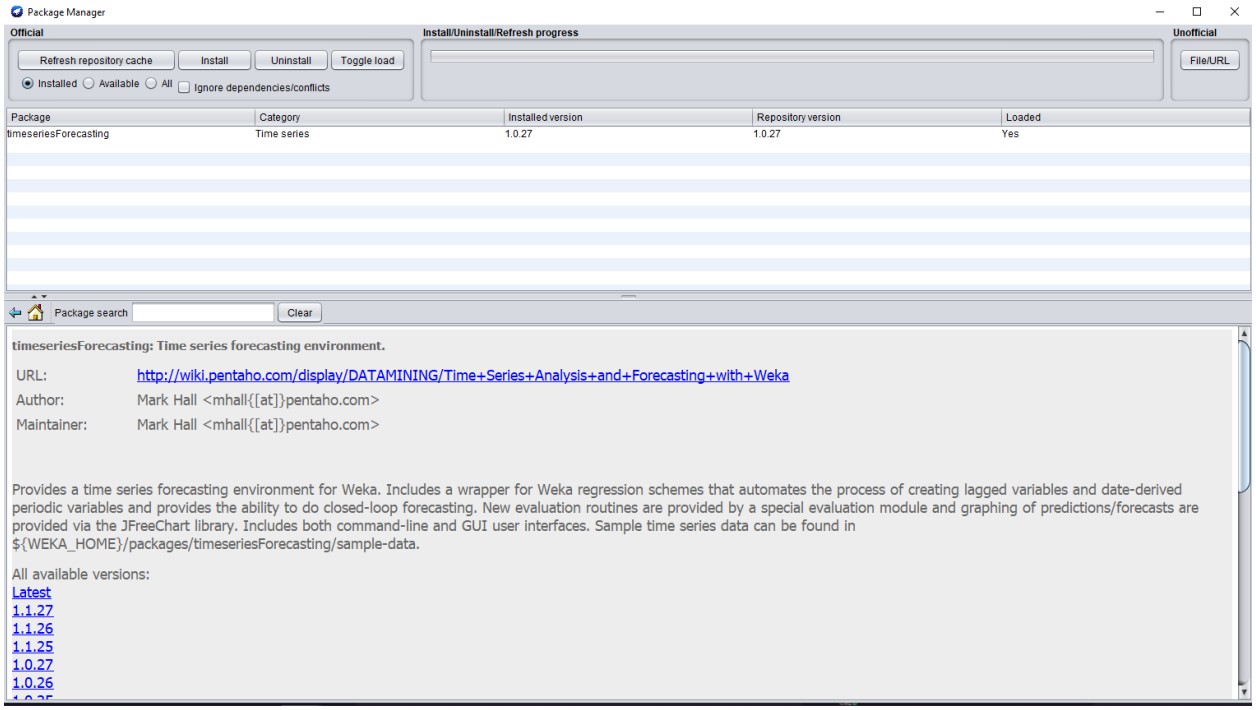
4.2Εξέταση των δεδομένων μας με το Weka

Για να περάσουμε τα δεδομένα μας στο Weka θα πρέπει να τα μετατρέψουμε σε arff αρχείο. Για αυτό το λόγο φτιάξαμε το script csv2arff.py (Ο κώδικάς είναι στα επισυναπτόμενα αρχεία) και παράγαγε το αρχείο household_power_consumption.arff έτσι ώστε τώρα να χρησιμοποιήσουμε το weka.

Τώρα στο weka καναμε install το package timeseriesforecasting

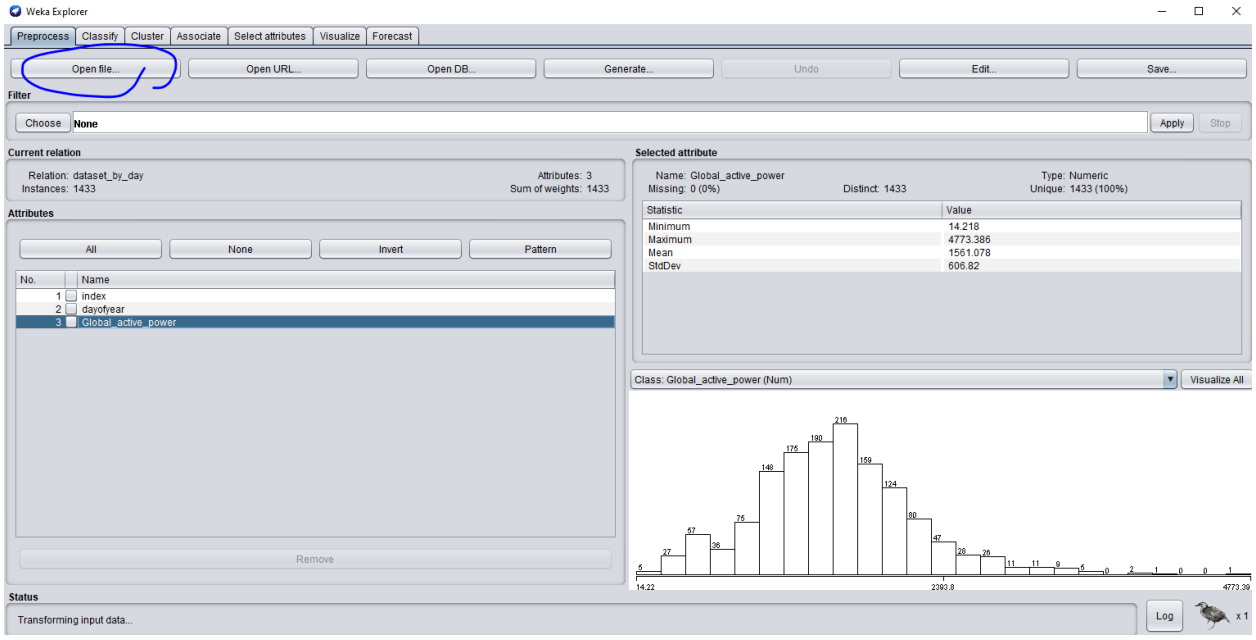


Εικόνα 35:Weka interface



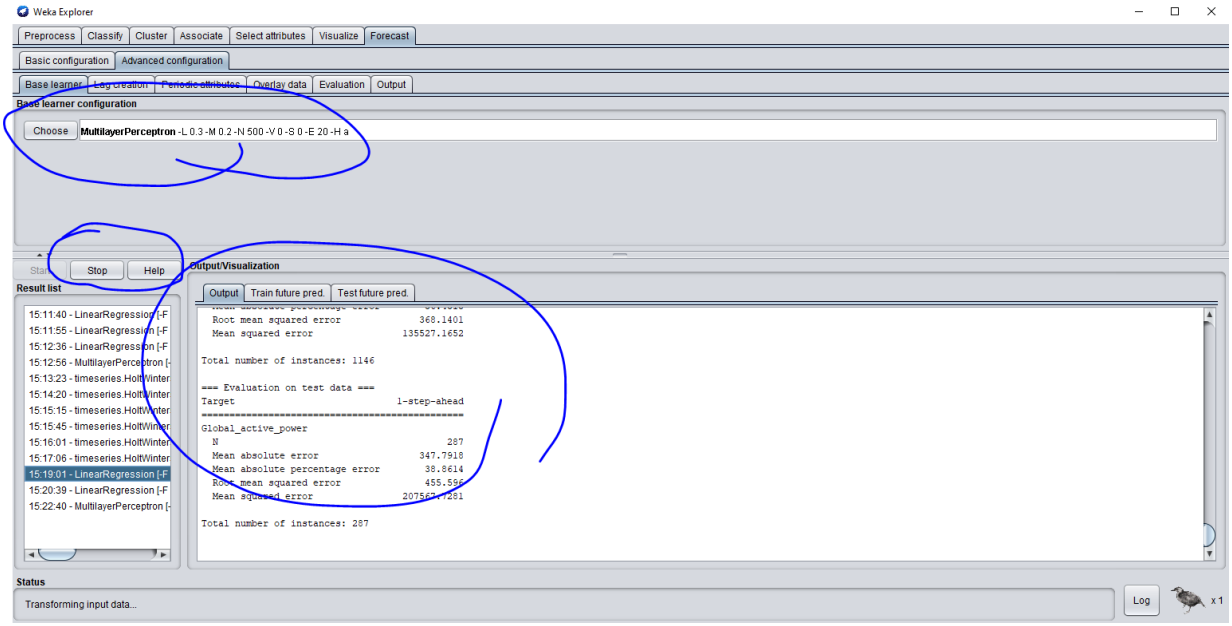
Εικόνα 36: Models installation

Στην συνέχεια βάζουμε το dataset μας



Εικόνα 37: data insert weka

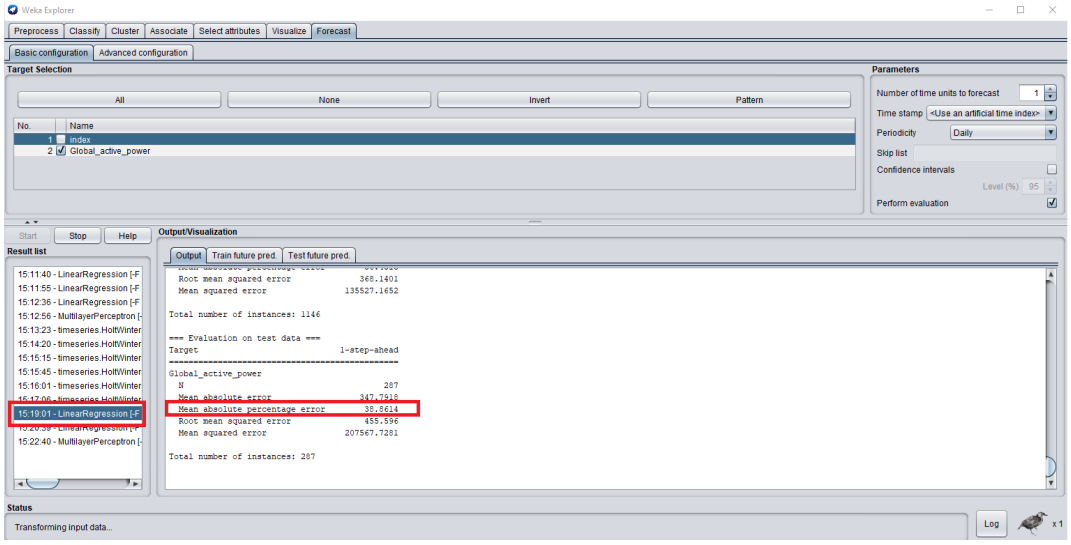
Και ανοίγουμε το forecasting tab και όπως βλέπουμε στο base learning στο Advanced configuration βαζουμε το μοντελο που θέλουμε και τις υπόλοιπες ρυθμίσεις και τρέχουμε τα μοντέλα μας.



Εικόνα 38: Model results

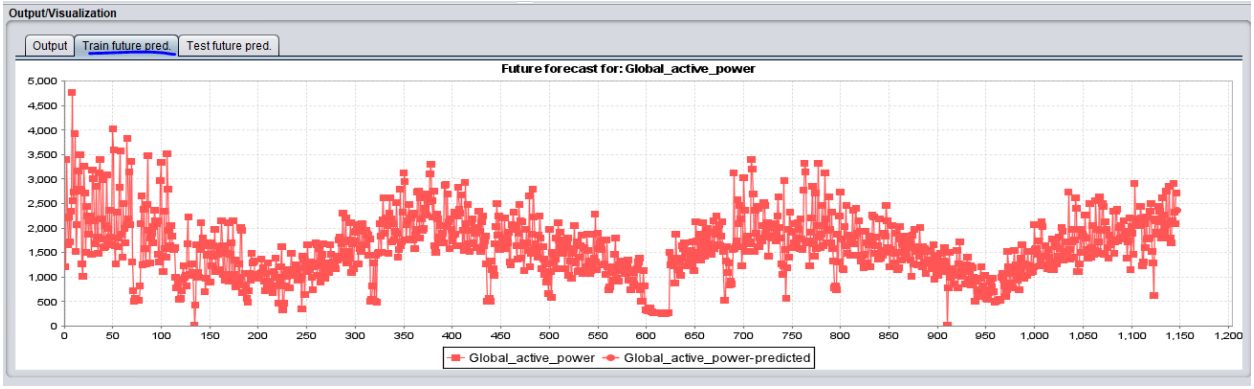
Τώρα ας εξετάσουμε τα μοντέλα που θέλουμε να δοκιμάσουμε

Για να δούμε πως αντιλαμβάνεται το weka και για να δούμε εάν το μοντελο που έχουμε φτιάξει επαληθεύετε θα κάνουμε ένα μοντελο με linear Regression

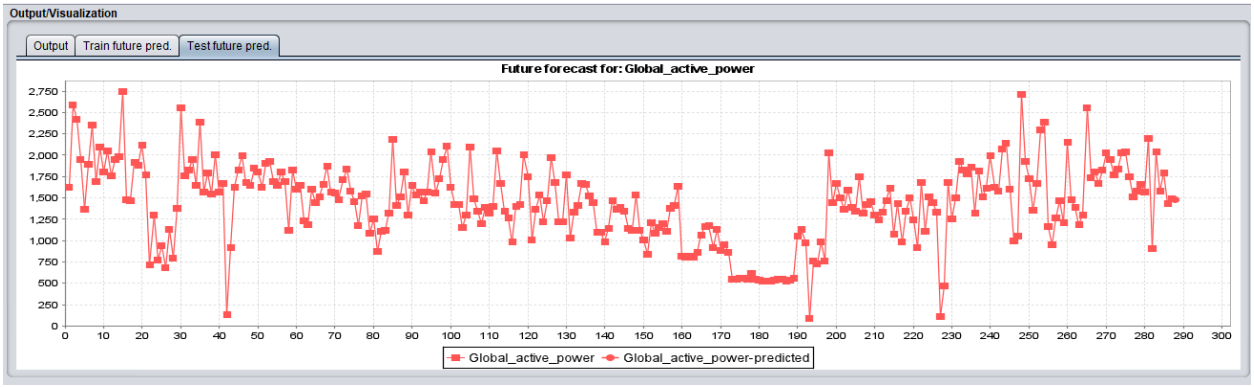


Εικόνα 39: Weka model results

Μας έβγαλε το μοντέλο μας στις καλύτερη version που πετύχαμε στο 100- 38.8=61.2% accuracy



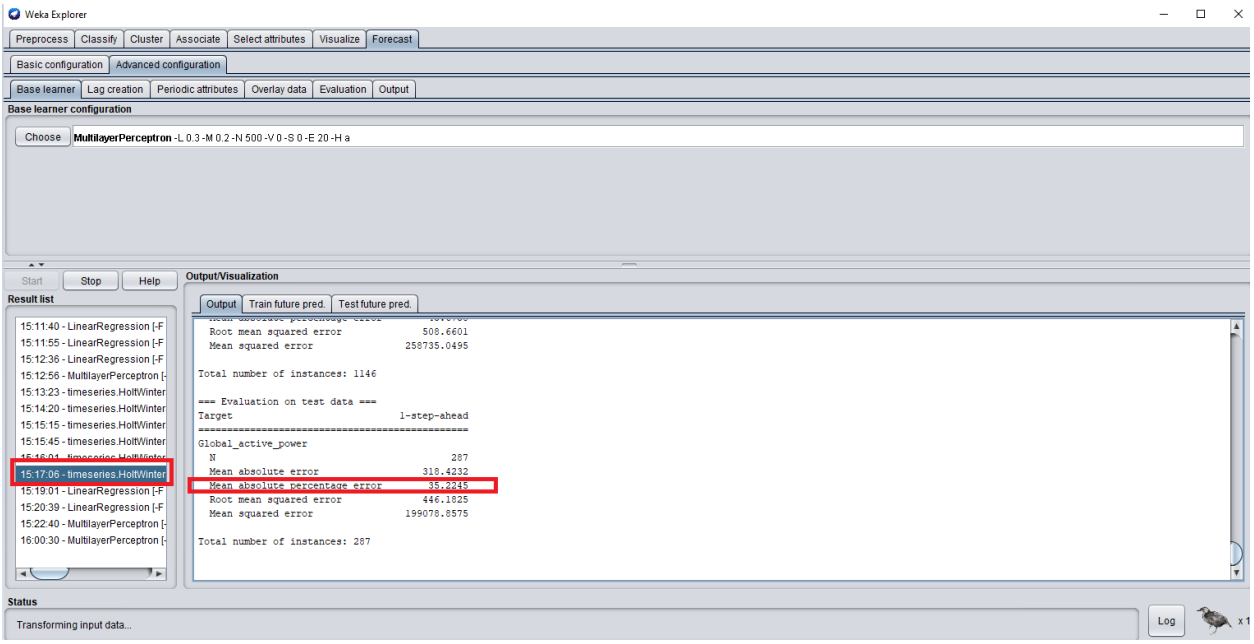
Εικόνα 40: Train Forecasting results



Εικόνα 41: Test Forecasting results

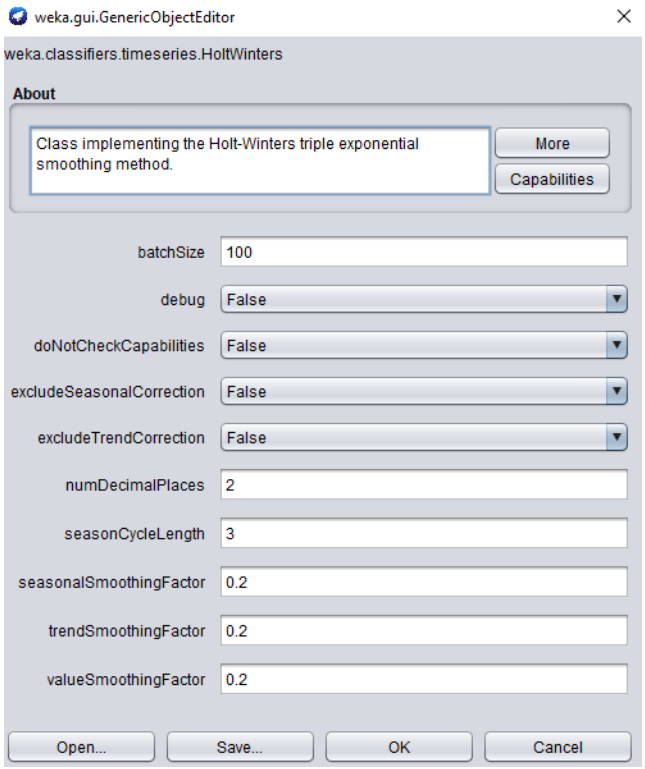
Άρα το μοντέλο που έχουμε φτιάξει επαληθεύετε και με το weka και ήταν το καλύτερο που καταφέραμε να φτάσουμε. Τωρα σκοπός μας είναι να προσπαθήσουμε και νέα μοντέλα αρχικα στο weka όπου θα έχουμε γρήγορα αποτελέσματα και στην συνέχεια να φτιάξουμε το μοντέλο με την βοήθεια τις python.

Στην συνέχεια χρησιμοποιούμε το μοντέλο Holtwinters

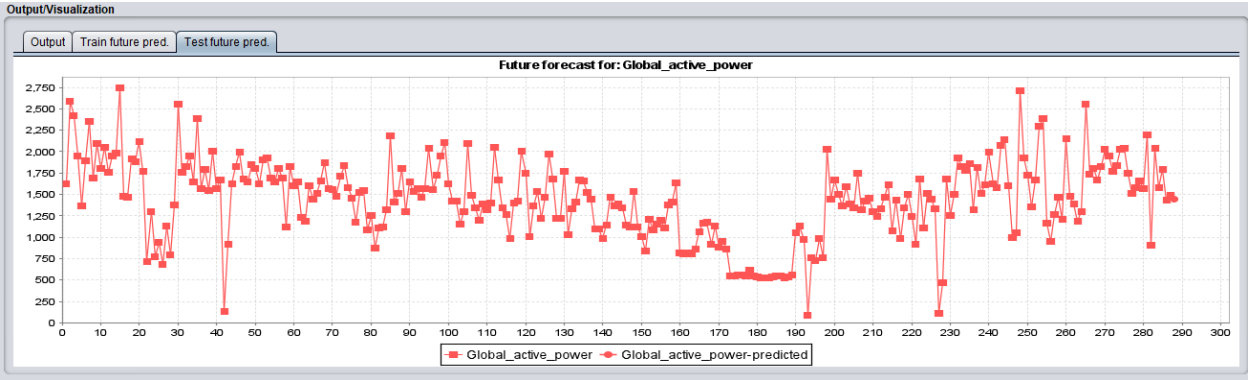


Εικόνα 42:Weka model results

Με ποσοστό 65% στην πρώτη προσπάθεια, μετα την αλλαγή του seasonCycleLenght στο 3 και καταφέραμε να το φτάσουμε στο 67%



Εικόνα 43: Weka model configuration



Εικόνα 44: weka model plot

4.3 Συμπέρασμα σχετικά με το Weka

Σκοπός μας είναι να εφαρμόσουμε τα ίδια μοντέλα με την χρήση python έτσι ώστε να δουμε ότι τα αποτελέσματα (ανεξάρτητα εάν έχουμε ίδια αποσοστα) με την χρήση Python θα μας φέρουν τα ίδια αποτελεσματα σαν συνολική εικόνα έτσι ώστε να εξετάσουμε σε αλλαγή παραμέτρων και να κάνουμε συγκρίσεις με διαφορετικές περιόδικες χρονοσειρές για τα ίδια δεδομένα (days months years etc) Η αρχική μας πρόβλεψη όπως βλεπουμε για τα μοντέλα μας μέχρι τώρα συμπίπτει με τα αποτελέσματα του weka που μας κάνει να προσπαθήσουμε να καταλάβουμε με ποιον τρόπο θα κάνουμε το forecast μας καλύτερο. Μια σκέψη είναι ότι τα δεδομένα που έχουμε δεν είναι κατάλληλα για να εξετάσουμε ενεργειακά δεδομένα ή ακόμα καλύτερα δεν εχουμε καταφέρει να βρουμε seasonality στα δεδομένα μας. Σε ένα global scale όπως είναι τα δεδομένα που έχουμε το πρόβλημα είναι ότι θα πρέπει να εξετάσουμε πχ day-night για να έχει μια βάση όλη η πρόβλεψη που προσπαθούμε να κάνουμε και όχι με βάση την ημέρα ή τον μήνα. Καλύτερα αποτελέσματα θα πάρουμε εάν εξετάσουμε ανα 12ώρες ή ακόμα και ανα grouping σε 6 ή 4 ώρες τα δεδομένα μας πιθανότατα. Το weka μας βοήθησε να εξετάσουμε γρηγορά εύκολα και χωρις να χρειαζτεί να γράψουμε κώδικά ή να επεξεργαστούμε πολυ τα δεδομένα μας. Είναι ένα tool που έχει μεγάλο βάθος άλλα θέλει τον χρόνο του να εξοικειωθείς και να καταλάβεις τα μοντέλα καλύτερα και να μπορεις να τα λειτουργήσεις. Πάντως με τις λίγες γνώσεις μας πάνω στο tool τα αποτελέσματα που είδαμε ήταν αυτά που περιμέναμε. Είδαμε απο το μοντέλο linear ότι τα ποσοστά είναι πολυ μικρά άρα καταλαβαίνουμε ότι θα πρέπει να κανουμε καλύτερο μετασχηματισμό στα δεδομένα μας και στο Holt-winters βράλαμε καλύτερα αποτελέσματα αλλα και παλι οχι τα επιθυμητά.Εξετάσαμε το weka για να δουμε τα έτοιμα tool κατα πόσο μπορούν να λειτουργήσουν και τι αποτελέσματα μας δίνουν.

Γ. Συμπεράσματα

Το πιο αποδοτικό μας μοντέλο είδαμε ότι ειναι το lstm και πιο συγκεκριμένα το lstm per month data grouping, τα δεδομένα μας χρειάστικαν μια πρώτη επεξεργασία σε όλα τα μοντέλα που χρησιμοποιήσαμε και σε όλα τα πειράματα\δοκιμές που κάναμε. Μπορειτε να δείτε αναλυτικά στο προηγούμενα κεφάλαια τα σθυμπεράσματα για τα μοντελα συνοπτικά αλλα και ένα ενα τα αποτελέσματα τις γραφικές και τα διαγράμματα μας για κάθε ενα ξεχωριστά αυτό που καταλάβαμε για τα δεδομένα μας απο όλη την επεξεργασία και όλη την μελέτη πάνω στο γνωστικό αντικείμενο είναι ότι θα πρέπει να κάνουμε pre-process\ cleansing. Cleansing στα δεδομένα μας κάναμε μονο για να αφαιρέσουμε τα missing values και τπτ περισσότερο. Αλλα στην διαδικαάσια pre-process σκεφτήκαμε να ομοδοποιήσουμε τα δεδομένα μας για να καταφέρουμε να τα εξετάσουμε για την περιοδικότητα τους αλλα και για να θέσουμε time period με στατικες περιόδουμε (days ,months, years etc). Αυτό μας έκανε τα δεδομένα μας πιο αποδοτικά εν τελή στα αποτελέσματα που βγάλαμε απο τα μοντέλα μας.

Τα metrics που χρησιμοποιήσαμε για να κάνουμε συγκριση στα μοντέλα μας ειναι το MAPE (Πληροφορίες στην θεωρία μας) για να έχουμε ίδια μέτρα συγκρισής. Για το κάθε μοντέλο βγάλαμε διαφορετικά αποτελέσματα και εχουμε κανει τις μεταξύ του συγκρίσεις. Το τελικό μας αποτέλεσμα είναι ότι δεν μπορέσαμε να βρούμε συγκεκριμένη περιοδικότητα στα δεδομένα μας σε κανένα απο τα μοντέλα μας. Ακόμα ειδαμε ότι όσο περισσότερο χρόνο και προσπαθειες\πειράματα κάναμε σε καθε ενα απο τα μοντέλα μας βοηθούσε να κατανοήσουμε και να κάνουμε αλλαγές στον τρόπο προσεγγισης του κάθε πειράματος μας. Η εμπειρία και οι δοκιμές πάνω σε ενα τομέα παιζει τον μεγαλύτερο ρόλο για την καλύτερη εκβαση τον αποτελεσμάτων μας. Καταφέραμε να φτάσουμε σε ενα ποσοστο επυτιχιας 78.5% περιμέναμε στην αρχη ενα ποσοστό μεγαλύτερο πάνω απο 80%, απο τον τρόπο επεξεργασίας και ανάλυσης τον δεδομένων μας καταλάβαμε ότι δεν έχουν και την καλύτερη δομή για να μας βοηθήσει να ξεπεράσουμε αυτό το ποσοστό. Με βάση τα αποτελέσματα μας είμαστε ευχαριστήμενοι για την εμπειρία που αποκτήσαμε στον κλάδο ενέργειας και της μηχανικής μάθησης πάνω σε χρονοσειρές! Σκεψεις για περαιτέρω αναλυσης και διαφορετικών τρόπον προσέγγισης είναι αυτό το οποιο εκ κατακλείδι συμπεράνουμε. Αυτό μας έδειξαν τα αποτελέσματα μας.

Βιβλιογραφία

- https://machinelearningmastery.com/multi-step-time-series-forecasting-with-machine-learning-models-for-household-electricity-consumption/?fbclid=IwAR1KyDeCpvvEXHfYlVcMYrDa23mI5LXSBEZJRY05lqiwLZujD-TbCtoNjCg
- http://home.iitj.ac.in/~parmod/document/introduction%20time%20series.pdf?fbclid=IwAR2eenTrAln-rYLx5w2MesK2wA2Hv9_iCIkSkx1SUtPNPsjEXUTUt1gzRlU
- https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption?fbclid=IwAR1PIWedAuW8cG8kKJlwLrL1FwgxU6w-30mtPjeskwgSV-eipPvt6ssbuKE
- https://jakevdp.github.io/PythonDataScienceHandbook/?fbclid=IwAR29bOEcSP4oEZ5Pds_6uuZDmiS6dwOLBPfiv1V5ylCAdRFy9Kn_m_i1sZU
- https://el.wikipedia.org/wiki/%CE%9C%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CE%AE_%CE%BC%CE%AC%CE%B8%CE%B7%CF%83%CE%B7#%CE%9F%CF%81%CE%B9%CF%83%CE%BC%CF%8C%CF%82
- Layout image credit: Cube3D
- https://el.wikipedia.org/wiki/%CE%9C%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CE%AE_%CE%BC%CE%AC%CE%B8%CE%B7%CF%83%CE%B7
- Machine Learning Strategies for Time Series Forecasting (Gianluca Bontempi, Souhaib Ben Taieb, and Yann-A¨el Le Borgne)
- https://www.cs.waikato.ac.nz/ml/weka/
- https://towardsdatascience.com/a-quick-start-of-time-series-forecasting-with-a-practical-example-using-fb-prophet-31c4447a2274
- An Introductory Study on Time Series Modeling and Forecasting- Ratnadip Adhikari
- Forecasting at scale – fbprophet paper Facebook, Menlo Park, California, United States