ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΑΝΑΛΥΣΗ ΚΑΙ ΠΡΟΒΛΕΨΗ ΧΡΟΝΟΣΕΙΡΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΡΥΤΗΟΝ

υπό

ΑΠΟΣΤΟΛΟΥ ΑΔΡΑΚΤΑ

Μεταπτυχιακή Εργασία

Υπεβλήθη για την εκπλήρωση μέρους των
απαιτήσεων για την απόκτηση του
Μεταπτυχιακού Διπλώματος Ειδίκευσης

Βόλος, 2022

Εγκρίθηκε από τα Μέλη της Τριμελούς Εξεταστικής Επιτροπής:

Πρώτος Εξεταστής (Επιβλέπων)	Δρ. Γεώργιος Λυμπερόπουλος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας
Δεύτερος Εξεταστής	Δρ. Παντελής Δημήτριος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Τρίτος Εξεταστής	Δρ. Σαχαρίδης Γιώργος Αναπληρωτής Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας

Ευχαριστίες

Θα ήθελα αρχικά να ευχαριστήσω τον επιβλέποντα της διπλωματικής μου εργασίας Δρ. Γεώργιο Λυμπερόπουλο ο οποίος μου έδωσε το την ευκαιρία να ασχοληθώ με το συγκεκριμένο αντικείμενο, καθώς και για την συνεννόηση και την βοήθεια του σε όλη τη διάρκεια εκπόνησης της διπλωματικής μου εργασίας. Επίσης ένα θερμό ευχαριστώ οφείλω σε όλους τους διδάσκοντες του για την ευκαιρία που μου έδωσαν να συμμετάσχω στο ΠΜΣ και να συνεργαστώ μαζί τους.

K.M

ΑΝΑΛΥΣΗ ΚΑΙ ΠΡΟΒΛΕΨΗ ΧΡΟΝΟΣΕΙΡΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ PYTHON

ΑΔΡΑΚΤΑΣ ΑΠΟΣΤΟΛΟΣ

Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας, 2022

Επιβλέπων Καθηγητής: Δρ. Γεώργιος Λυμπερόπουλος

Καθηγητής Στοχαστικών Μεθόδων στην Διοίκηση Παραγωγής

Περίληψη

Σήμερα, υπάρχουν πολλά διαφορετικά μοντέλα πρόβλεψης για χρονικές σειρές, με το καθένα να απαιτεί κατάλληλη προ επεξεργασία και ανάλυση δεδομένων για να παρέχει μια αξιοποιήσιμη πρόβλεψη. Ο στόχος αυτής της εργασίας είναι να διεξαχθεί μια συγκριτική μελέτη σχετικά με τους πιο συχνά χρησιμοποιούμενα μοντέλα χρονοσειρών προκειμένου να συγκριθεί η απόδοσή τους όπως ARIMA, SARIMA, Holt Winter’s κλπ. και τα συγκρίνουμε τα με το λογισμικό που αναπτύχθηκε από την IBM που ονομάζεται SPSS Statistics. Όλα τα μοντέλα που εφαρμόζονται είναι αυτοματοποιημένα, καθιστώντας την αναζήτηση παραμέτρων μέρος του μοντέλου, αυτό γίνεται έτσι ώστε να μπορεί να χρησιμοποιηθεί χωρίς προηγούμενη γνώση των μοντέλων ή των συνόλων δεδομένων στα οποία θα εφαρμοστούν.

TIME SERIES ANALYSIS AND FORECASTING IN PYTHON

APOSTOLOS ADRAKTAS

Department of Mechanical Engineering, University of Thessaly, 2021

Supervisor: Dr Liberopoulos George

Professor of Stochastic Methods in Production Management

Abstract

Today, there are plenty of various forecasting models for Time Series with each one requiring proper data preprocessing and analysis to provide a usable prediction. The aim of this report is to conduct a comparative study on the most commonly used Time Series models in order to benchmark their performance like ARIMA, SARIMA, Holt winter's and compare them to the in-house estimator developed by IBM called SPSS Statistics. All the implemented models are automated, making hyper-parameter search a part of the model, this is done so that it can be used without any prior knowledge of the models, or the datasets on which they will be applied on.

Περιεχόμενα

Κατάλογος Πινάκων

Κατάλογος Σχημάτων

Λίστα Συντομογραφιών

ACF	AutoCorrelation function
AIC	Akaike's Information Criterion
AR	Auto Regression
ARIMA	AutoRegressive Integrated-Moving Average
ARMA	Auto Regressive Moving Average
BIC	Bayesian Information Criterion
MA	Moving Average
MAD	Mean Absolute Deviation
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ME	Mean Error
MLP	Multilayer Perceptron
MPE	Mean Percentage Error
MSD	Mean Square Deviation
MSE	Mean Square Error
PACF	Partial Autocorrelation Coefficient
DES	Double Exponential Smoothing
ES	Exponential Smoothing
MSE	Mean Squared Error
RMSE	Root Mean Square Error
SARIMA	Seasonal Auto Regressive Integrated Moving Average
SES	Simple Exponential Smoothing

Κεφάλαιο 1. ΕΙΣΑΓΩΓΗ

1.1 Κίνητρο και Υπόβαθρο

Η μοντελοποίηση χρονοσειρών είναι ένας δυναμικός ερευνητικός τομέας που έχει προσελκύσει την προσοχή της ερευνητικής κοινότητας τις τελευταίες δεκαετίες. Ο κύριος στόχος της μοντελοποίησης χρονοσειρών είναι η συλλογή, η ανάλυση και η αυστηρή μελέτη του παρελθόντος και η ανάπτυξη ενός κατάλληλου μοντέλου που περιγράφει την εγγενή δομή της σειράς. Αυτό το μοντέλο χρησιμοποιείται στη συνέχεια για να εξηγήσει τη χρονοσειρά και να προβλέψει τις μελλοντικές τιμές για αυτήν, δηλαδή για την πρόβλεψη.

Η πρόβλεψη χρονοσειρών είναι μια από τις πιο κοινές και χρησιμοποιούμενες εργασίες μάθησης, καθημερινά οι επιχειρήσεις χρησιμοποιούν πρόβλεψη χρονοσειρών για μια μεγάλη ποικιλία σκοπών, όπως πρόβλεψη ημερήσιων τιμών μετοχών, πρόβλεψη συναλλαγματικών ισοτιμιών, πρόβλεψη ποσοστών ανεργίας. Οι μετεωρολόγοι το χρησιμοποιούν για να παρέχουν μια εκτίμηση της ταχύτητας του ανέμου, τις ημερήσιες μέγιστες και ελάχιστες θερμοκρασίες και κατά προσέγγιση τις βροχοπτώσεις. Όλα αυτά και πολλά άλλα καθήκοντα δείχνουν τη σημασία των χρονοσειρών και τη σημασία της καλής εκτίμησης του μέλλοντος, καθώς μπορεί να είναι υψίστης σημασίας για τις επιχειρήσεις να προετοιμαστούν για πιθανή άνοδο/πτώση των πωλήσεών τους και να προετοιμαστούν για αυτό ή να αποφύγουν την καταστροφή όταν παρατηρούν μετεωρολογικά δεδομένα.

Η πρόβλεψη χρονοσειρών είναι η πράξη πρόβλεψης του μέλλοντος μετά από προσεκτική εξέταση και ανάλυση του παρελθόντος, λόγω της απαραίτητης σημασίας αυτού του έργου σε πολλούς τομείς όπως τα οικονομικά, η μετεωρολογία, οι επιχειρήσεις, η επιστήμη και η μηχανική. Η προετοιμασία ενός κατάλληλου μοντέλου για να ταιριάζει και στη συνέχεια να προβλέψει τη σειρά δεν είναι μια προφανής εργασία, καθώς κάθε σήμα/σειρά έχει τις δικές του ιδιότητες και εξαρτήσεις από εξωγενείς παραμέτρους που δεν μπορούν εύκολα να αναπαρασταθούν στο μοντέλο. Με τα χρόνια, έχει προταθεί πληθώρα ερευνών και μοντέλων από ακαδημαϊκούς, στατιστικολόγους και οικονομολόγους για να

βελτιωθεί η ακρίβεια των προβλέψεων. Ως αποτέλεσμα, διάφορα μοντέλα σειρών Time έχουν τεθεί σε λειτουργία/βελτιωθεί, αλλά αυτή η αφθονία μοντέλων δεν σημαίνει απαραίτητα ότι αυτά τα μοντέλα είναι πανταχού παρόντα. Οι πιο δημοφιλείς και χρησιμοποιούμενες προσεγγίσεις εξακολουθούν να είναι τα στατιστικά μοντέλα όπως το ARIMA και το ES και τα μοντέλα παλινδρόμησης μηχανικής μάθησης (που εφαρμόζονται κατάλληλα στις χρονοσειρές) όπως το SVR.. Μπορούν να χρησιμοποιηθούν με μια μεγάλη ποικιλία δεδομένων (εξωγενών) που μπορούν να εμπλουτίσουν τη γνώση που παρέχουν οι χρονοσειρές.

Κεφάλαιο 2. `ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΠΡΟΒΛΕΨΕΙΣ

Η πρόβλεψη αποτελεί μια από τις σημαντικότερες ενέργειες που καλείται ένα άτομο να κάνει σε πολλούς και διαφορετικούς τομείς της ζωής του, αυτό μπορεί να αφορά την προσωπική ζωή, την επαγγελματική, την επιχειρησιακή, την οικονομική κ.α. Στόχος του κάθε ατόμου είναι η λήψη σημαντικών και καθοριστικών αποφάσεων για το μέλλον του είτε σε προσωπικό επίπεδο είτε σε κοινωνικό. Η πρόβλεψη διακρίνεται σε βραχυπρόθεσμη, μεσοπρόθεσμη ή μακροπρόθεσμη ,ανάλογα με το χρονικό ορίζοντα στον οποίο απευθύνεται.

2.1 Χαρακτηριστικά των προβλέψεων

Οι προβλέψεις, όταν καθοριστούν, αντιμετωπίζονται συχνά ως γνωστές πληροφορίες. Οι απαιτήσεις πόρων και τα χρονοδιαγράμματα παραγωγής ενδέχεται να απαιτούν τροποποιήσεις εάν η πρόβλεψη της ζήτησης αποδειχθεί ανακριβής. Ας δούμε τα κύρια χαρακτηριστικά των προβλέψεων:

1. Το σύστημα σχεδιασμού πρέπει να είναι αρκετά ισχυρό ώστε να μπορεί να αντιδρά σε απρόβλεπτα σφάλματα πρόβλεψης.

2. Μια καλή πρόβλεψη είναι κάτι παραπάνω από έναν μόνο αριθμό. Δεδομένου ότι οι προβλέψεις είναι γενικά λάθος, μια καλή πρόβλεψη περιλαμβάνει επίσης κάποιο μέτρο του αναμενόμενου σφάλματος πρόβλεψης. Αυτό θα μπορούσε να έχει τη μορφή ενός εύρους ή ενός μέτρου σφάλματος όπως η διακύμανση της κατανομής του σφάλματος πρόβλεψης.

3. Οι συνολικές προβλέψεις είναι πιο ακριβείς. Θυμηθείτε από τη στατιστική ότι η διακύμανση του μέσου όρου μιας συλλογής ανεξάρτητων ταυτόσημων κατανεμημένων τυχαίων μεταβλητών είναι χαμηλότερη από τη διακύμανση κάθε μιας από τις τυχαίες μεταβλητές. Δηλαδή, η διακύμανση του μέσου δείγματος είναι μικρότερη από τη διακύμανση του πληθυσμού. Το ίδιο φαινόμενο ισχύει και για τις προβλέψεις. Σε ποσοστιαία βάση, το σφάλμα που έγινε στις προβλέψεις πωλήσεων

για μια ολόκληρη σειρά προϊόντων είναι γενικά μικρότερο από το σφάλμα που έγινε στις προβλέψεις πωλήσεων για ένα μεμονωμένο στοιχείο.

4. Όσο μεγαλύτερος είναι ο ορίζοντας πρόβλεψης, τόσο λιγότερο ακριβής θα είναι η πρόβλεψη.

5. Οι προβλέψεις δεν πρέπει να χρησιμοποιούνται για τον αποκλεισμό γνωστών πληροφοριών. Μια συγκεκριμένη τεχνική μπορεί να οδηγήσει σε αρκετά ακριβείς προβλέψεις στις περισσότερες περιπτώσεις. Ωστόσο, ενδέχεται να υπάρχουν διαθέσιμες πληροφορίες σχετικά με τη μελλοντική ζήτηση που δεν παρουσιάζεται στην προηγούμενη ιστορία της σειράς. Για παράδειγμα, η εταιρεία μπορεί να σχεδιάζει μια ειδική προωθητική πώληση για ένα συγκεκριμένο προϊόν, έτσι ώστε η ζήτηση πιθανότατα να είναι υψηλότερη από την κανονική. Αυτές οι πληροφορίες πρέπει να συμπεριληφθούν χειροκίνητα στην πρόβλεψη.

2.2 Βασικά στάδια στη διαδικασία πρόβλεψης

Τα βασικά στάδια σε μια διαδικασία πρόβλεψης είναι τα εξής:

1. *Καθορισμός Προβλήματος (Problem Definition)*

Τις περισσότερες φορές είναι το πιο δύσκολο μέρος στη διαδικασία πρόβλεψης και ταυτόχρονα το πιο σημαντικό. Αυτό συμβαίνει γιατί θα πρέπει να γίνουν σαφή και κατανοητά ορισμένα θέματα, όπως το πώς θα χρησιμοποιηθούν οι προβλέψεις και από ποιους.

2. *Συγκέντρωση Πληροφοριών (Gathering Information)*

Στο δεύτερο βήμα απαιτούνται τουλάχιστον δύο είδη πληροφοριών. Στο πρώτο είναι τα στατιστικά (συνήθως αριθμητικά) δεδομένα και το δεύτερο η κρίση, η πείρα και η εμπειρία του προσωπικού που ασχολούνταν με αυτή τη συλλογή για αυτό το χρονικό διάστημα. Επίσης

οι παραπάνω πληροφορίες πρέπει να συλλεχθούν πριν ξεκινήσει η διαδικασία της πρόβλεψης.

3. *Προκαταρτική Ανάλυση (Exploratory Analysis)*

Στο βήμα αυτό μας απασχολεί το είδος της πληροφορίας που αποκομίζουμε από τα ακατέργαστα ιστορικά δεδομένα. Αρχικά, αναπαριστούμε γραφικά τα δεδομένα και στη συνέχεια, υπολογίζουμε κάποιους βασικούς στατιστικούς δείκτες, όπως η μέση τιμή, η τυπική απόκλιση, ελάχιστο, μέγιστο και γραμμική τάση. Οι παραπάνω δείκτες αναδεικνύουν κάποια δευτερεύοντα χαρακτηριστικά της χρονοσειράς. Σκοπός μας είναι να αποκτήσουμε μία αίσθηση των δεδομένων, δίνοντας απαντήσεις σε ερωτήματα όπως αν υπάρχουν λανθασμένα πρότυπα, αν υπάρχει σημαντική τάση ή εποχικότητα και τέλος, αν υπάρχουν ασυνήθιστες τιμές (outliers).Η ανάλυση αυτή μας οδηγεί στην οικογένεια μοντέλων πρόβλεψης που λογικά αναμένεται να δώσει ικανοποιητικές προβλέψεις.

4. *Επιλογή και Προσαρμογή Μοντέλου (Choosing & Fitting models).*

Εδώ γίνεται η επιλογή και καθορισμός των παραμέτρων διάφορων ποσοτικών μοντέλων πρόβλεψης που έχουν επιλεγεί στο προηγούμενο βήμα

Στο τελικό στάδιο, αφού ένα μοντέλο έχει επιλεγεί υποκειμενικά και οι παράμετροι του έχουν, προηγουμένως, καθοριστεί, χρησιμοποιείται ώστε να παραχθούν προβλέψεις. Κατά την εξέλιξη της διαδικασίας, γίνεται αποτίμηση των πλεονεκτημάτων και μειονεκτημάτων του μοντέλου και, εφόσον κριθεί απαραίτητο, επαναλαμβάνονται κάποια βήματα στη διαδικασία.

2.3 Εισαγωγή στις Χρονοσειρές

Η Χρονοσειρά είναι γενικά δεδομένα που συλλέγονται με την πάροδο του χρόνου και εξαρτώνται από αυτήν. Όταν τα δεδομένα μας είναι ανεξάρτητα από το χρόνο, τότε δεν μπορούμε να τα βάλουμε σε χρονοσειρές. Ας δούμε τώρα τον επίσημο ορισμό της χρονοσειράς.

Μια σειρά σημείων δεδομένων που συλλέγονται σε χρονική σειρά είναι γνωστή ως χρονολογική σειρά. Οι περισσότεροι επιχειρηματικοί οίκοι εργάζονται σε δεδομένα χρονολογικών σειρών για να αναλύσουν τον αριθμό πωλήσεων για το επόμενο έτος, την επισκεψιμότητα ισοτόπων, τον αριθμό κυκλοφορίας, τον αριθμό των κλήσεων που ελήφθησαν κ.λπ. Τα δεδομένα μιας χρονοσειράς μπορούν να χρησιμοποιηθούν για την πρόβλεψη. Δεν αντιπροσωπεύουν όλα τα δεδομένα που συλλέγονται σχετικά με το χρόνο μια χρονολογική σειρά.

Αρχικά, είναι σημαντικό να καθορίσουμε γρήγορα τους τυπικούς όρους που χρησιμοποιούνται κατά την περιγραφή δεδομένων Χρονοσειρών. Ο τρέχων χρόνος ορίζεται ως (t) και μια παρατήρηση την τρέχουσα ώρα ορίζεται ως obs(t). Μας ενδιαφέρει συχνά οι παρατηρήσεις που έγιναν σε προηγούμενες ώρες, που ονομάζονται χρόνοι καθυστέρησης ή καθυστερήσεις. Οι ώρες στο παρελθόν είναι αρνητικές σε σχέση με την τρέχουσα ώρα. Για παράδειγμα, η προηγούμενη ώρα είναι t-1 και η ώρα πριν από αυτή είναι t-2. Οι παρατηρήσεις αυτές τις στιγμές είναι obs (t-1) και obs (t-2) αντίστοιχα. Οι χρόνος στο μέλλον είναι αυτό που μας ενδιαφέρει να προβλέψουμε και είναι θετικός σε σχέση με την τρέχουσα ώρα. Για παράδειγμα, η επόμενη χρονική στιγμή είναι t + 1 και η επόμενη στιγμή είναι t + 2. Οι παρατηρήσεις αυτές τις στιγμές είναι obs (t + 1) και obs (t + 2) αντίστοιχα. Για απλότητα, συχνά αφήνουμε τη σημείωση obs (t) και χρησιμοποιούμε το t + 1 και υποθέτουμε ότι μιλάμε για παρατηρήσεις κατά καιρούς και όχι για τους ίδιους τους δείκτες χρόνου.

Συνοψίζοντας έχουμε:

- t-n: Προηγούμενη ή καθυστέρηση (π.χ. t-1 για την προηγούμενη ώρα).
- t: Τρέχουσα ώρα και σημείο αναφοράς.
- t + n: Μελλοντικός χρόνος ή πρόβλεψη (π.χ. t + 1 για την επόμενη φορά).

2.3.1 Συστατικά μιας χρονοσειράς

Οι χρονοσειρές είναι ιστορικά δεδομένα που απαρτίζονται από διαδοχικές παρατηρήσεις μέσα σε ένα χρονικό διάστημα. Οι παρατηρήσεις γίνονται ανά σταθερό

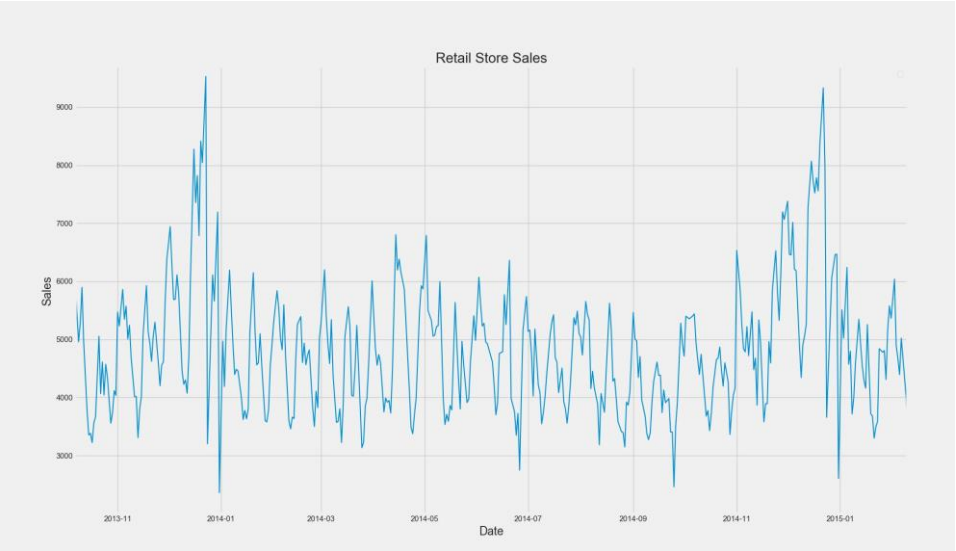
χρονικό βήμα και μπορούν να είναι ετήσιες, τριμηνιαίες, μηνιαίες, εβδομαδιαίες, ημερήσιες κτλ. Τα ποιοτικά χαρακτηριστικά αυτών των παρατηρήσεων είναι τα εξής:

- Στασιμότητα (Stationary), όταν οι τιμές κυμαίνονται γύρω από μία μέση τιμή.
- Τάση (Trend) όταν υπάρχει μια μακροπρόθεσμη αύξηση ή μείωση του επιπέδου των τιμών.
- Εποχικότητα (Seasonal) όταν η χρονοσειρά επηρεάζεται από εποχιακούς παράγοντες.
- Κυκλικότητα(Cyclical) όταν οι τιμές αυξομειώνονται, αλλά όχι σε σταθερές περιόδους.
- Τυχαιότητα (Irregular-Random), όταν έχουμε διακυμάνσεις λόγω τυχαίων γεγονότων.

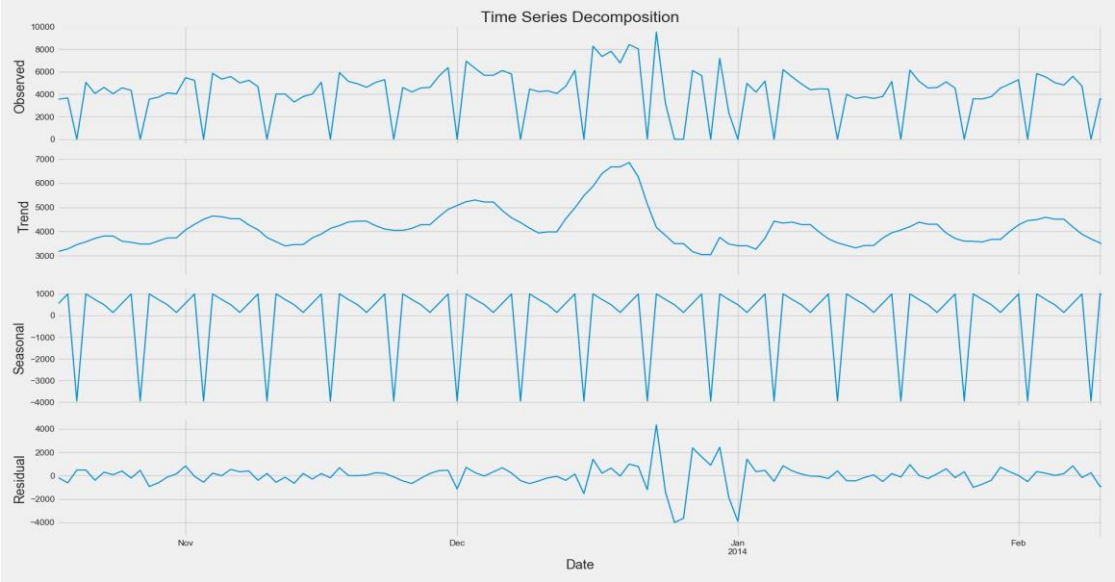
$$y(t) = T(t) + S(t) + C(t) + \varepsilon(t) \quad (2.1)$$

$$y(t) = T(t) * S(t) * C(t) * \varepsilon(t) \quad (2.2)$$

Το y (t) αντιπροσωπεύει το μέτρο που καταγράφηκε στο βήμα t, T (t) είναι η συνολική τάση του η σειρά, S (t) περιγράφει την εποχική πτυχή της χρονοσειράς, C (t) είναι η κυκλική συνιστώσα της παρατήρησης και ε(t) αντιπροσωπεύει τα ακανόνιστα μοτίβα στο σειρά, υπολείμματα.



Εικόνα 1 - Πωλήσεις Καταστήματος Λιανικής



Εικόνα 2 *Αποσύνθεση των συστατικών των πωλήσεων του καταστήματος λιανικής*

2.4 Διαφορά μια Χρονοσειράς από ένα πρόβλημα παλινδρόμησης

Εδώ μπορείτε να σκεφτείτε ότι η μεταβλητή που έχουμε ως στόχο είναι αριθμητική μπορεί να προβλεφθεί με τη χρήση τεχνικών παλινδρόμησης, αλλά ένα πρόβλημα χρονολογικές σειρές είναι διαφορετικό από ένα πρόβλημα παλινδρόμησης με τους εξής τρόπους:

- Η κύρια διαφορά είναι ότι μια χρονική σειρά εξαρτάται από το χρόνο.
- Έτσι, η βασική υπόθεση ενός μοντέλου γραμμικής παλινδρόμησης ότι οι παρατηρήσεις είναι ανεξάρτητες δεν ισχύει σε αυτήν την περίπτωση.
- Μαζί με μια αυξανόμενη ή φθίνουσα τάση, οι περισσότερες Time Series έχουν κάποια μορφή εποχιακών τάσεων, δηλαδή. παραλλαγές συγκεκριμένες για ένα συγκεκριμένο χρονικό πλαίσιο.

Έτσι, η πρόβλεψη μιας χρονοσειράς χρησιμοποιώντας τεχνικές παλινδρόμησης δεν είναι καλή προσέγγιση. Η ανάλυση Χρονοσειρών περιλαμβάνει μεθόδους για την ανάλυση δεδομένων Χρονοσειρών με σκοπό την εξαγωγή σημαντικών στατιστικών και άλλων χαρακτηριστικών των δεδομένων. Η πρόβλεψη Χρονοσειρών είναι η χρήση ενός μοντέλου για την πρόβλεψη μελλοντικών τιμών βάσει των τιμών που παρατηρήθηκαν προηγουμένως.

Κεφάλαιο 3. ΑΝΑΛΥΣΗ ΧΡΟΝΟΣΕΙΡΑΣ

Ο κυριότερος στόχος στην ανάλυση Χρονοσειρών είναι η επιλογή και προσαρμογή του κατάλληλου μοντέλου που να προσεγγίζει ικανοποιητικά τα δεδομένα και να περιγράφει το μηχανισμό της χρονοσειράς από την οποία προέκυψε η συγκεκριμένη σειρά, καθώς και η χρησιμοποίηση του μοντέλου για πρόβλεψη.

Η μεγαλύτερη πρόκληση στην ανάλυση Χρονοσειρών είναι η πρόβλεψη, δηλαδή πως η ακολουθία των παρατηρήσεων θα συνεχιστεί στο μέλλον. Το ζητούμενο είναι να ακολουθεί μια διαδικασία που θα εξασφαλίσει ότι θα παραχθούν όσο το δυνατόν πιο ακριβείς προβλέψεις, αξιοποιώντας στο έπακρο όλη την διαθέσιμη ιστορική πληροφορία .

Στο παρόν κεφάλαιο θα παρουσιαστούν τα στοχαστικά μοντέλα του λευκού θορύβου, της αυτοπαλίνρομης διαδικασίας (AR), του κινητού μέσου (MA) και του αυτοπαλίνδρομου κινητού μέσου (ARMA) τα οποία αναφέρονται όλα σε στάσιμες διαδικασίες. Αυτό σημαίνει ότι ο μέσος όρος, η διακύμανση και οι αυτοσυνδιακυμάνσεις δεν εξαρτώνται από το χρόνο t. Υπάρχουν, όμως σειρές που δεν μπορούν να προσαρμοστούν σε κανένα από αυτά τα μοντέλα. Τέλος, θα αναλυθούν οι μη στάσιμες χρονοσειρές του τυχαίου περιπάτου και των ολοκληρωμένων αυτοπαλίνδρομων μοντέλων κινητού μέσου ARIMA.

3.1 Λευκός θόρυβος

Η πιο απλή στάσιμη χρονοσειρά που θα εξετάσουμε ονομάζεται λευκός θόρυβος (white noise). Συνιστά δομική μονάδα για όλες τις υπόλοιπες χρονοσειρές που θα μελετηθούν στη συνέχεια [14] και ορίζεται από την σχέση:

$$Y_t = E_t$$

Οι παρατηρήσεις μιας χρονοσειράς λευκού θορύβου et, έχουν μέση τιμή ίση με μηδέν και σταθερή διακύμανση όλες τις χρονικές στιγμές. Ακόμη, όλες οι παρατηρήσεις είναι ασυσχέτιστες μεταξύ τους, δηλαδή η αυτοσυνδιακύμανση της j-οστής τάξης γjt είναι μηδενική για κάθε j ≠ 0. Οι τρεις βασικές συνθήκες είναι οι εξής:

$$\mu_t = E(\varepsilon_t) = 0 \text{ , για κάθε t} \qquad 3,1$$

$$\gamma_{0t} = E(Y_t - \mu_t)^2 = E(\varepsilon_t^2) = \sigma^2 \text{ , για κάθε t} \qquad 3,2$$

$$\gamma_{jt} = E(Y_t - \mu_t)(Y_{t-j} - \mu_{t-j}) = E(\varepsilon_t \varepsilon_{t-j}) = 0 \text{ , για κάθε t, j} \neq 0. \qquad 3,3$$

3.2 Διάσπαση Χρονοσειρών

Η διάσπαση χρονοσειρών χρησιμοποιείται κυρίως για ανάλυση χρονοσειρών, και ως εργαλείο ανάλυσης μπορεί να χρησιμοποιηθεί για την ενημέρωση μοντέλων πρόβλεψης στο πρόβλημά μας. Παρέχει έναν δομημένο τρόπο σκέψης για ένα πρόβλημα πρόβλεψης χρονολογικών σειρών, τόσο γενικά όσον αφορά την πολυπλοκότητα μοντελοποίησης και συγκεκριμένα όσον και στον τρόπο καλύτερης λήψης κάθε ενός από αυτά τα στοιχεία σε ένα δεδομένο μοντέλο.

Κάθε ένα από αυτά τα στοιχεία είναι κάτι που ίσως χρειαστούμε να σκεφτούμε και να αντιμετωπίσουμε κατά την προετοιμασία δεδομένων, την επιλογή μοντέλου και τον παραμετροποίηση του μοντέλου. Μπορείτε να την αντιμετωπίσουμε όσον αφορά τη μοντελοποίηση της τάσης και την αφαίρεσή της από τα δεδομένα μας, ή σιωπηρά παρέχοντας αρκετό ιστορικό για έναν αλγόριθμο για τη μοντελοποίηση μιας τάσης εάν υπάρχει.

Τα προβλήματα του πραγματικού κόσμου είναι ακατάστατα και θορυβώδη. Μπορεί να υπάρχουν πρόσθετα (additive) και πολλαπλασιαστικά (multiplicative) συστατικά. Μπορεί να υπάρχει μια αυξανόμενη τάση ακολουθούμενη από μια φθίνουσα τάση. Μπορεί να υπάρχουν μη επαναλαμβανόμενοι κύκλοι αναμεμιγμένοι με τα επαναλαμβανόμενα συστατικά εποχικότητας. Ωστόσο, αυτά τα αφηρημένα μοντέλα παρέχουν ένα απλό πλαίσιο που μπορούμε να χρησιμοποιήσουμε για να αναλύσουμε τα δεδομένα μας και να εξερευνήσουμε τρόπους σκέψης και πρόβλεψης του προβλήματός μας

. Μια δεδομένη χρονική σειρά θεωρείται ότι αποτελείται από τρία συστηματικά στοιχεία, όπως επίπεδο, τάση, εποχικότητα και ένα μη συστηματικό στοιχείο που ονομάζεται θόρυβος. Αυτά τα στοιχεία ορίζονται ως εξής:

- Level: Η μέση τιμή στη σειρά..
- Trend: Η αυξανόμενη ή φθίνουσα τιμή στη σειρά.
- Seasonality: Ο επαναλαμβανόμενος βραχυπρόθεσμος κύκλος στη σειρά.
- Noise: Η τυχαία παραλλαγή της σειράς

Μια σειρά θεωρείται ότι είναι ένα σύνολο ή συνδυασμός αυτών των τεσσάρων συστατικών. Όλες οι σειρές έχουν επίπεδο και θόρυβο. Τα στοιχεία τάσης και εποχικότητας είναι προαιρετικά. Είναι χρήσιμο να σκεφτούμε τα συστατικά ως συνδυασμό είτε πρόσθετα είτε πολλαπλασιαστικά.

3.2.1 Additive Model

Ένα πρόσθετο μοντέλο υποδηλώνει ότι τα συστατικά προστίθενται μαζί ως εξής:

$$y(t) = Level + T rend + Seasonality + Noise \quad 3,4$$

$$Y(t) = T(t) + S(t) + C(t) + I(t) \quad 3,5$$

Ένα πρόσθετο μοντέλο είναι γραμμικό όπου οι αλλαγές με την πάροδο του χρόνου γίνονται με συνέπεια με την ίδια ποσότητα. Μια γραμμική τάση είναι μια ευθεία γραμμή. Μια γραμμική εποχικότητα έχει την ίδια συχνότητα (πλάτος κύκλων) και πλάτος (ύψος κύκλων).

3.2.2 Multiplicative Model

Ένα πολλαπλασιαστικό μοντέλο υποδηλώνει ότι τα στοιχεία πολλαπλασιάζονται μαζί ως εξής:

$$y(t) = Level \times Trend \times Seasonality \times Noise \quad 3,6$$

$$Y(t) = T(t) \cdot S(t) \cdot C(t) \cdot I(t) \quad 3,7$$

Ένα πολλαπλασιαστικό μοντέλο είναι μη γραμμικό, όπως τετραγωνικό ή εκθετικό. Οι αλλαγές αυξάνονται ή μειώνονται με την πάροδο του χρόνου. Μια μη γραμμική τάση είναι μια καμπύλη γραμμή. Μια μη γραμμική εποχικότητα έχει μια αυξανόμενη ή μειωμένη συχνότητα και / ή πλάτος με την πάροδο του χρόνου.

3.2.3 Ανάλυση και διαχωρισμός της εποχικότητας

Μια τάση είναι μια συνεχής αύξηση ή μείωση μεσα στην σειρά με την πάροδο του χρόνου. Μπορεί να υπάρχει όφελος στον εντοπισμό, τη μοντελοποίηση και ακόμη και την κατάργηση πληροφοριών τάσεων από το σύνολο των δεδομένων των Χρονοσειρών μας. Ο εντοπισμός και η κατανόηση των πληροφοριών τάσεων μπορούν να βοηθήσουν στη βελτίωση της απόδοσης του μοντέλου. Παρακάτω είναι μερικοί λόγοι:

- Ταχύτερη μοντελοποίηση: Ίσως η γνώση μιας τάσης ή η έλλειψη τάσης μπορεί να προτείνει μεθόδους και να κάνει την επιλογή και αξιολόγηση του μοντέλου πιο αποτελεσματική.
- Απλούστερο πρόβλημα: Ίσως μπορούμε να διορθώσουμε ή να αφαιρέσουμε την τάση για απλοποίηση της μοντελοποίησης και βελτίωση της απόδοσης του μοντέλου
- Περισσότερα δεδομένα: Ίσως μπορούμε να χρησιμοποιήσουμε τις πληροφορίες τάσης, άμεσα ή ως περίληψη, για να παρέχουμε πρόσθετες πληροφορίες στο μοντέλο και να βελτιώσουμε την απόδοση του μοντέλου.

Τύποι τάσεων Υπάρχουν όλες οι τάσεις. Δύο είναι γενικά οι τάξεις για τις οποίες μπορούμε να σκεφτούμε είναι:

- Ντετερμινιστικές τάσεις: Πρόκειται για τάσεις που αυξάνονται ή μειώνονται συνεχώς.
- Στοχαστικές τάσεις: Αυτές είναι τάσεις που αυξάνονται και μειώνονται με συνέπεια.

Σε γενικές γραμμές, οι ντετερμινιστικές τάσεις είναι πιο εύκολο να εντοπιστούν και να αφαιρεθούν, αλλά οι μέθοδοι που συζητούνται σε αυτό το σεμινάριο εξακολουθούν να είναι χρήσιμες για στοχαστικές τάσεις. Μπορούμε να σκεφτούμε τις τάσεις όσον αφορά το εύρος των παρατηρήσεών τους.

• Παγκόσμιες τάσεις: Αυτές είναι τάσεις που ισχύουν για ολόκληρες τις χρονοσειρές.

• Τοπικές τάσεις: Αυτές είναι τάσεις που ισχύουν για τμήματα ή ακολουθίες μιας χρονοσειράς.

Γενικά, οι παγκόσμιες τάσεις είναι πιο εύκολο να εντοπιστούν και να αντιμετωπιστούν Χρησιμοποιώντας τις τάσεις των Χρονοσειρών στη μηχανική εκμάθηση από την άποψη της μηχανικής μάθησης, μια τάση στα δεδομένα σας αντιπροσωπεύει δύο ευκαιρίες:

• Κατάργηση πληροφοριών: Για να αφαιρέσετε συστηματικές πληροφορίες που στρεβλώνουν τη σχέση μεταξύ των μεταβλητών εισόδου και εξόδου.

• Προσθήκη πληροφοριών: Για να προσθέσετε συστηματικές πληροφορίες για τη βελτίωση της σχέσης μεταξύ των μεταβλητών εισόδου και εξόδου

Συγκεκριμένα, μια τάση μπορεί να αφαιρεθεί από τα δεδομένα των Χρονοσειρών σας (και τα δεδομένα στο μέλλον) ως προετοιμασία και καθαρισμό δεδομένων. Αυτό είναι συνηθισμένο όταν χρησιμοποιείτε στατιστικές μεθόδους για πρόβλεψη Χρονοσειρών, αλλά δεν βελτιώνει πάντα τα αποτελέσματα όταν χρησιμοποιείτε μοντέλα μηχανικής μάθησης. Εναλλακτικά, μια τάση μπορεί να προστεθεί, είτε άμεσα είτε ως περίληψη, ως μια νέα μεταβλητή εισαγωγής στο εποπτευόμενο μαθησιακό πρόβλημα για την πρόβλεψη της μεταβλητής εξόδου

3.2.3.1 Detrend by Differencing

Ίσως η απλούστερη μέθοδος αποτροπής μιας χρονοσειράς είναι η διάκριση. Συγκεκριμένα, κατασκευάζεται μια νέα σειρά όπου η τιμή στο τρέχον βήμα χρόνου

υπολογίζεται ως η διαφορά μεταξύ της αρχικής παρατήρησης και της παρατήρησης στο προηγούμενο χρονικό βήμα.

τιμή (t) = παρατήρηση (t) - παρατήρηση (t - 1) *3,8*

Αυτό έχει ως αποτέλεσμα την κατάργηση μιας τάσης από ένα σύνολο δεδομένων Χρονοσειρών

3.2.3.2 Detrend by Model Fitting

Μια τάση συχνά απεικονίζεται εύκολα ως γραμμή μέσω των παρατηρήσεων. Οι γραμμικές τάσεις μπορούν να συνοψιστούν με ένα γραμμικό μοντέλο και οι μη γραμμικές τάσεις μπορούν να συνοψιστούν καλύτερα χρησιμοποιώντας μια πολυώνυμή ή άλλη μέθοδο προσαρμογής καμπύλης. Λόγω του υποκειμενικού χαρακτήρα και του συγκεκριμένου τομέα της ταυτοποίησης των τάσεων, αυτή η προσέγγιση μπορεί να βοηθήσει στον προσδιορισμό του εάν υπάρχει μια τάση. Ακόμη και η προσαρμογή ενός γραμμικού μοντέλου σε μια τάση που είναι σαφώς υπεργραμμική ή εκθετική μπορεί να είναι χρήσιμη. Εκτός από το ότι χρησιμοποιούνται ως εργαλείο αναγνώρισης τάσεων, αυτά τα μοντέλα προσαρμογής μπορούν επίσης να χρησιμοποιηθούν για να καταστρέψουν μια χρονοσειρά

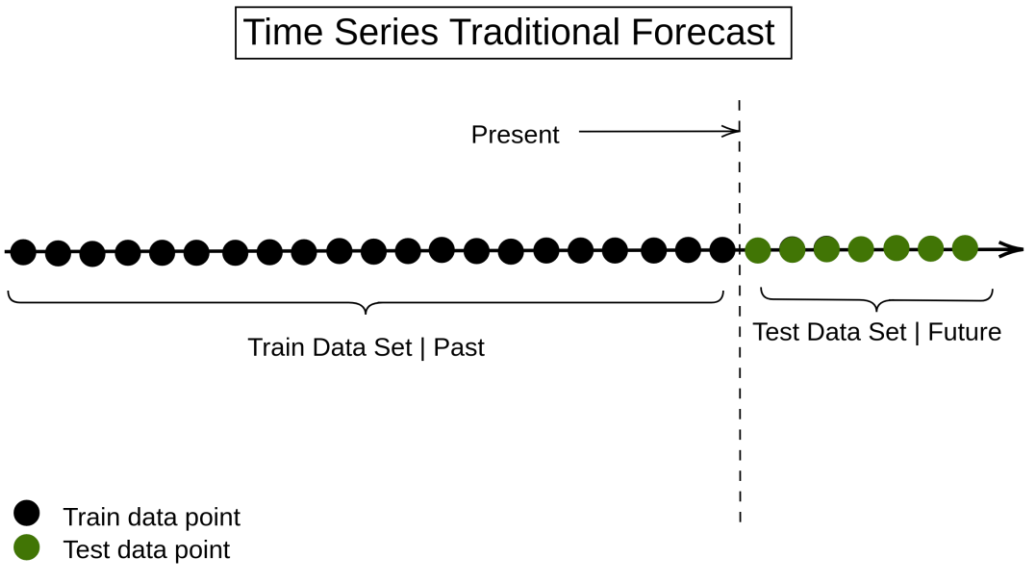
3.1.4 Cross Validation for Time Series

Η παραδοσιακή προσέγγιση πρόβλεψης για χρονικές σειρές είναι να εκπαιδεύσει το μοντέλο/εκτιμητή σε ένα μέρος των δεδομένων (σύνολο δεδομένων τρένου) στη συνέχεια να προβλέψει σε έναν ορίζοντα σταθερού μεγέθους και στη συνέχεια να υπολογίσει την απόδοση του εκτιμητή στο σύνολο δεδομένων δοκιμής.

Η προαναφερθείσα στρατηγική δεν είναι επαρκής για χρονοσειρές ειδικά για εκείνες με τεράστιο αριθμό σημείων δεδομένων. Αυτό προέρχεται από το γεγονός ότι δεν μπορούμε να υποθέσουμε ότι η διανομή της σειράς δεν θα αλλάξει με την πάροδο του χρόνου.
Από την άλλη πλευρά, το γεγονός ότι ορισμένες εξωτερικές/εξωγενείς μεταβλητές έχουν μια επίδραση που δεν μπορούμε απαραίτητα να μοντελοποιήσουμε/χρησιμοποιήσουμε όταν

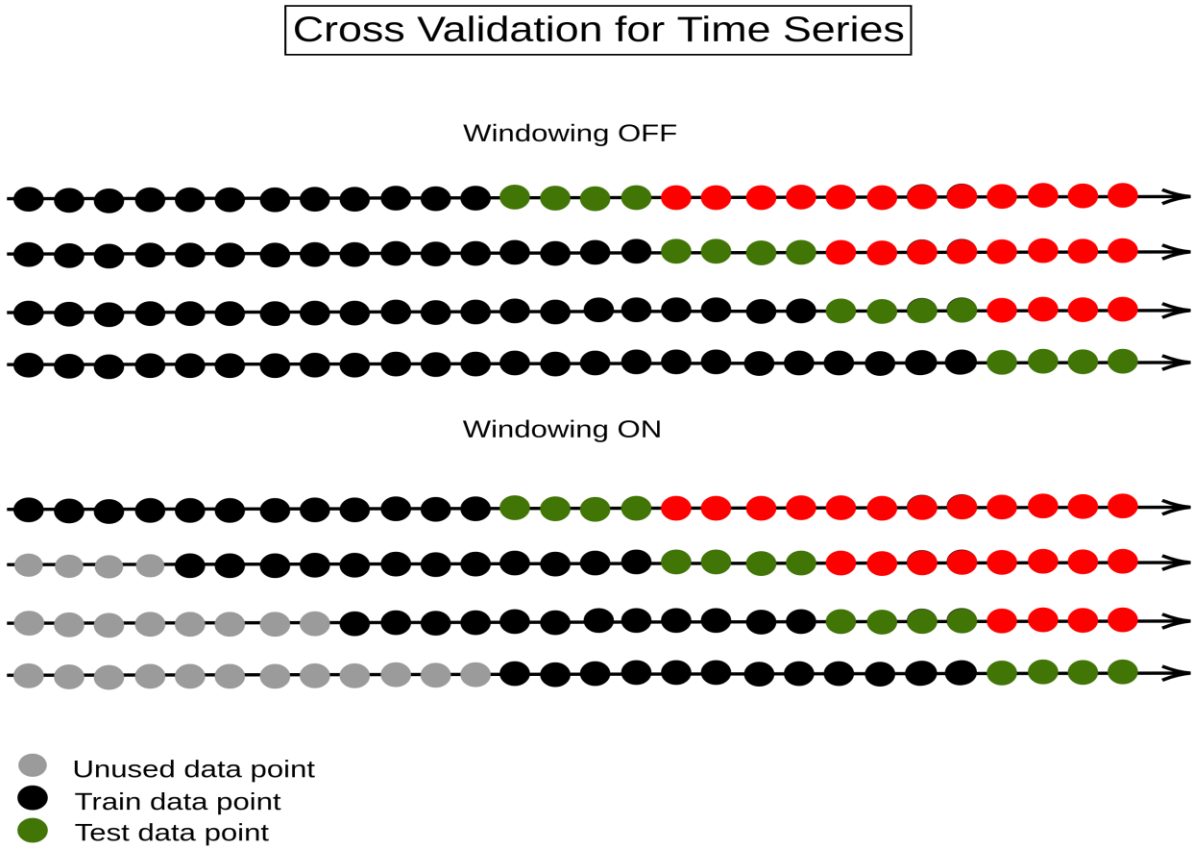
κάνουμε τις προβλέψεις μας, καθιστώντας αναξιόπιστο το έργο της πρόβλεψης σε σχετικά ευρύ ορίζοντα.

Για να αποτρέψουμε, λοιπόν, τέτοια ζητήματα, χρησιμοποιούμε μια προσέγγιση που ονομάζεται: Walk Forward Validation.



Εικόνα 3 Train Test Forecasting Approach

Αυτή η στρατηγική λαμβάνει υπόψη την ευαισθησία/συχνότητα της χρονοσειράς κατά την πρόβλεψη μη ορατών τιμών. Από επιχειρηματική/πρακτική άποψη, αυτή είναι η πιο σύγχρονη προσέγγιση/εφαρμογή του προβλήματος προβλέψεων χρονοσειρών καθώς μπορεί να παρέχει καλύτερες προβλέψεις (χρησιμοποιώντας τη μέγιστη ποσότητα πληροφοριών που έχει μέχρι σήμερα). Καθώς υποθέτουμε ότι μπορούμε να έχουμε καλές επιδόσεις από τους εκτιμητές μας μέχρι έναν συγκεκριμένο υπο-ορίζοντα, που μπορεί να συναχθεί από την ευαισθησία του συνόλου δεδομένων, και στη συνέχεια να επανεκπαιδεύσουμε το μοντέλο στο ίδιο σύνολο δεδομένων αμαξοστοιχίας και τις παρατηρήσεις που περιλαμβάνονται σε αυτόν τον υποορίζοντα που έχουμε ήδη προβλέψει. Αυτό αυξάνει τη γνώση των εκτιμητών καθώς κάνουν τις προβλέψεις τους και αντιμετωπίζουν ένα ζήτημα που είναι αχαλίνωτο στην παραδοσιακή προσέγγιση, η οποία προσαρμόζει το μοντέλο στην αλλαγή της συνολικής τάσης της σειράς.



Εικόνα 4 Walk Forward Validation.

Κεφάλαιο 4. ΜΕΘΟΔΟΙ ΠΡΟΒΛΕΨΕΩΝ

4.1 Μέθοδοι εξομάλυνσης

Μέθοδοι εξομάλυνσης (smoothing methods) είναι οι τεχνικές με τις οποίες προσδιορίζονται οι θα δοθούν σε μια μεταβλητή βάση τον τρόπο εφαρμογής τους. Τις τεχνικές αυτές τις ονομάζουμε μέθοδοι εξομάλυνσης, διότι η δημιουργία των προβλέψεων βασίζεται από την εξομάλυνση της διαχρονικής εξέλιξης των τιμών της μεταβλητής, ώστε να μπορούμε να αναγνωρίσουμε καλύτερα το τρόπος συμπεριφοράς της. Ορισμένες από αυτές τις μεθόδους μπορούν να εφαρμοστούν και σε περιπτώσεις μικρού αριθμού παρατηρήσεων της μεταβλητής

4.1.1 Η μέθοδος του κινητού μέσου (Moving average)

Η εξομάλυνση είναι μια τεχνική που εφαρμόζεται σε χρονοσειρές για την αφαίρεση της διακύμανσης μεταξύ βήματα χρόνου. Η ελπίδα της εξομάλυνσης είναι να αφαιρέσετε τον θόρυβο και να εκθέσετε καλύτερα το σήμα του υποκείμενες αιτιώδεις διαδικασίες. Οι κινούμενοι μέσοι όροι είναι ένας απλός και κοινός τύπος εξομάλυνσης χρησιμοποιείται στην ανάλυση Χρονοσειρών και στην πρόβλεψη Χρονοσειρών. Ο υπολογισμός ενός κινούμενου μέσου όρου περιλαμβάνει μια νέα σειρά όπου οι τιμές αποτελούνται από τον μέσο όρο των ακατέργαστων παρατηρήσεων στη χροσνοσειρά.

Ένας κινητός μέσος όρος απαιτεί να καθορίσετε ένα μέγεθος παραθύρου που ονομάζεται πλάτος παραθύρου. Αυτό ο αριθμός των ανεπεξέργαστων παρατηρήσεων που χρησιμοποιήθηκαν για τον υπολογισμό της τιμής της κίνησης. Υπάρχουν δύο βασικά τύποι κινούμενου μέσου όρου που χρησιμοποιούνται: Κεντρικός και κινούμενος μέσος όρος.

Η μέθοδος του κινητού μέσου όρου, σε αλγεβρική μορφή, διατυπώνεται ως εξής:

$$F_{t+1}=\frac{X_t+X_{t+1}+\ldots+X_{t-n+1}}{n}=\frac{1}{n}(\sum_{i=t-n+1}^{t}X_i) \qquad 4,1$$

όπου t είναι η πιο πρόσφατη παρατήρηση και t+1 είναι η επόμενη περίοδος. Ο τύπος αυτός απαιτεί ο προσβλέπων να διαθέτει τις τιμές των παρατηρήσεων του παρελθόντος. Με την προσθήκη μιας νέας παρατήρησης και την εξάλειψη της παλαιότερης, μπορούμε να επαναδιατυπώσουμε τον τύπο ως εξής:

$$F_{t+1}=\frac{1}{n}(\sum_{i=t-n}^{t-1}X_i)+\frac{1}{n}(X_t-X_{t-n})=F_t+\frac{X_t}{n}-\frac{X_{t-n}}{n} \qquad 4,2$$

4.1.2 Απλή εκθετική εξομάλυνση (Simple exponential smoothing)

Υπάρχουν σοβαροί περιορισμοί στη χρήση του κινητού μέσου όρου. Πρώτον, για τους σχετικούς υπολογισμούς απαιτούνται οι παρελθούσες n παρατηρήσεις του δείγματος. Αν πρόκειται να προβλεφθεί ένας μεγάλος αριθμός μεγεθών (μεταβλητών), τα δεδομένα απαιτούν μεγάλο αποθηκευτικό χώρο. Δεύτερον, δίνεται ίση στάθμιση σε όλες τις παρατηρήσεις που αφορούν το παρελθόν, χωρίς να δίνεται καμία στάθμιση στις παρατηρήσεις που είναι πριν από την περίοδο (t-n+ 1). Οι πρόσφατες παρατηρήσεις, εν τούτοις, μπορεί να περιέχουν περισσότερες πληροφορίες από τις παλαιότερες, πράγμα που έχει σημασία για τις μελλοντικές προβλέψεις. Έτσι, κάποιος μπορεί να θέλει να αποδώσει μεγαλύτερη στάθμιση στην πρόβλεψη η οποία βασίζεται στις πιο πρόσφατες παρατηρήσεις παρά στις παλαιότερες.

Μια μέθοδος που απλοποιεί τους υπολογισμούς της πρόβλεψης και έχει μικρές απαιτήσεις σε δεδομένα, ονομάζεται εκθετική εξομάλυνση. Επίσης, η μέθοδος αυτή δίνει αυτοδιορθούμενες προβλέψεις αφού διαθέτει μηχανισμό ενσωματωμένων προσαρμογών, ο οποίος ρυθμίζει τις τιμές αλλάζοντας τες προς την αντίθετη κατεύθυνση από εκείνη που κινήθηκαν τα λάθη προηγούμενων περιόδων:

$$F_{t+1}=aX_t+(1-a)F_t \qquad 4.3$$

όπου t είναι η τρέχουσα περίοδος, τα Ft+1 και F, είναι τιμές πρόβλεψης για την επόμενη και την τρέχουσα περίοδο και Xt είναι η τιμή που παρατηρήθηκε την τρέχουσα περίοδο. Το α ονομάζεται σταθερά εξομάλυνσης και παίρνει τιμές από 0 έως 1. Αφού η παραπάνω εξίσωση περιλαμβάνει μόνο μία σταθερά, το μοντέλο αυτό είναι μοντέλο εκθετικής εξομάλυνσης μιας παραμέτρου.

Η εκθετική εξομάλυνση μιας παραμέτρου είναι πολύ απλή μέθοδος, αφού μόνο μια τιμή, η πρόβλεψη της τελευταίας περιόδου, είναι αυτή που πρέπει να διασωθεί. Στην ουσία, ολόκληρη η χρονοσειρά εμπεριέχεται σ' αυτή την πρόβλεψη. Εάν εκφράσουμε το Ft σε όρους της προηγηθείσας παρατήρησης Xt-1 και των τιμών της πρόβλεψης Ft-1, τότε το ισοδύναμο για την πρόβλεψη της επόμενης περιόδου γίνεται:

$$F_{t+1} = aX_t + a(1-a)X_{t-1} + (1-a)^2 F_{t-1} \qquad 4,4$$

Η νέα αυτή εξίσωση είναι μοντέλο δευτεροβάθμιας εκθετικής εξομάλυνσης μιας παραμέτρου. Μπορούμε να συνεχίσουμε έτσι για έναν αριθμό προηγούμενων περιόδων, πράγμα που δείχνει ότι όλες οι προηγηθείσες τιμές του X αντανακλώνται στην τρέχουσα πρόβλεψη. Έτσι, το όνομα αυτής της διαδικασίας προέρχεται από τις διαδοχικές σταθμίσεις α, α(1-α), α(1-α) 2, α(1-α) 3,..., οι οποίες μειώνονται εκθετικά. Οι πιο πρόσφατες περίοδοι στη χρονοσειρά λαμβάνουν μεγαλύτερη στάθμιση στον υπολογισμό της πρόβλεψης. Πρακτικά, οι αρκετά παλιές τιμές της X εξαιρούνται. Η διαδικασία πρόβλεψης μπορεί να τροποποιηθεί οποιαδήποτε στιγμή με τη μεταβολή της τιμής της α. Μπορούμε να ξαναγράψουμε την εξίσωση (3.5) ως εξής:

$$F_{t+1} = F_t + ae_t \qquad 4,5$$

όπου t ae το σφάλμα πρόβλεψης για την περίοδο t, είναι η πραγματική τιμή μείον την τιμή της πρόβλεψης. Επομένως, βλέπουμε ότι η πρόβλεψη που δίνεται από την εκθετική εξομάλυνση είναι η παλαιά πρόβλεψη συν μια προσαρμογή για το σφάλμα που έγινε στην τελευταία πρόβλεψη. Όταν το α βρίσκεται πλησίον του 1, η νέα πρόβλεψη περιέχει μια ουσιώδη προσαρμογή για το σφάλμα της προηγούμενης πρόβλεψης. Αντίθετα, εάν το α

βρίσκεται πολύ κοντά στο 0, η νέα πρόβλεψη θα περιέχει μικρή μόνο προσαρμογή για το σφάλμα. Επομένως, το αποτέλεσμα του μεγέθους του α είναι όμοιο με τα αποτελέσματα των διαφόρων τιμών για τον αριθμό των παρατηρήσεων του δείγματος όταν υπολογίζουμε τον κινητό μέσο όρο. Τέλος, η εκθετική εξομάλυνση μιας παραμέτρου δίνει προβλέψεις που ακολουθούν το πρότυπο στα δεδομένα ενός δείγματος. Αυτό συμβαίνει γιατί η διαδικασία προσαρμόζει μόνο την επόμενη πρόβλεψη ως προς κάποιο ποσοστό τού πιο πρόσφατου σφάλματος πρόβλεψης, και δεν μπορεί να προβλέψει μεταβολές στην κατεύθυνση της χρονοσειράς.

Για να λύσουμε το πρόβλημα της επιλογής εφαρμόζουμε την ανάλυση ευαισθησίας στις ιστορικές χρονοσειρές, χρησιμοποιώντας διαφορετικές τιμές για τη σταθερά της εξομάλυνσης. Για κάθε τιμή πάνω από ένα εύρος τιμών, προετοιμάζεται ένα πρότυπο πρόβλεψης με τη μέθοδο της εκθετικής εξομάλυνσης και υπολογίζεται το κατάλληλο μέτρο της ακριβείας της πρόβλεψης. Στην πράξη, μελέτες που έγιναν, δείχνουν ότι οι τιμές από 0.05 έως 0.30 ταιριάζουν πολύ καλά στα πρότυπα της εκθετικής εξομάλυνσης. Τιμές της α που είναι μεγαλύτερες από 0.30 συνήθως δείχνουν ότι κάποιο εναλλακτικό μοντέλο πρόβλεψης θα είναι περισσότερο κατάλληλο

4.1.3 Εκθετική εξομάλυνση με προσαρμογή στην τάση (Exponential smoothing adjusted for trend)

Ονομάζεται επίσης μέθοδος Brown. Χρησιμοποιείται κυρίως σε σειρές που παρουσιάζουν τάση. Η βασική φιλοσοφία της μεθόδου είναι σχεδόν ίδια με αυτήν του διπλού κινητού μέσου, δηλαδή η εξομάλυνση των παρατηρήσεων της χρονοσειράς γίνεται δύο φορές, ενώ στη διαμόρφωση των προβλέψεων λαμβάνεται υπόψη και η τάση. Η εφαρμογή της μεθόδου στηρίζεται στην ακόλουθη διαδικασία[4]:

- Εξομαλύνονται οι αρχικές παρατηρήσεις της χρονοσειράς με τη μέθοδο της απλή εκθετικής εξομάλυνσης ως ακολούθως:

$$A_t = aX_t + (1-a)A_{t-1} \qquad 4,6$$

όπου α είναι η σταθερά εξομάλυνσης, γ ια 0 ≤ α ≤ 1, At οι εξομαλυνθείσες τιμές της χρονοσειράς που προκύπτουν από την εξομάλυνση, για t=2, 3, ..., n ενώ για t=1 ορίζεται ως αρχική συνθήκη A1 = X1.

- Εξομαλύνονται οι εξομαλυνθείσες τιμές At της χρονοσειράς με τη μέθοδο της απλής εκθετικής εξομάλυνσης ως ακολούθως:

$$A'_t = aA_t + (1\ a)A'_{t-1} \qquad 4,7$$

όπου A′t οι εξομαλυνθείσες τιμές της χρονοσειράς που προκύπτουν από την δεύτερη εξομάλυνση, για t=2, 3, ..., n ενώ για t=1, A'1 = x1. Υπολογίζεται η διαφορά a, t ως:

$$\alpha_t = aA_t - A'_t \qquad 4,9$$

4.1.4 Εκθετική εξομάλυνση με προσαρμογή στην τάση και στην εποχικότητα (Exponential smoothing adjusted for trend and seasonality

Ο Winters ανέπτυξε μια μέθοδο για την προσαρμογή της εποχικής ή περιοδικής κίνησης μέσα στο πλαίσιο της γραμμικής εκθετικής εξομάλυνσης με ή χωρίς τάση. Επομένως, η διαδικασία του Winters μπορεί να εφαρμοστεί για προβλέψεις με βάση μια χρονοσειρά που εμφανίζει και τάση και εποχικό πρότυπο. Πρόκειται ουσιαστικά για μία επέκταση της μεθόδου διπλής εκθετικής εξομάλυνσης (double exponential smoothing), η οποία εφαρμόζεται σε δεδομένα που χαρακτηρίζονται από την ύπαρξη τάσης αλλά όχι εποχικότητας. Η επέκταση συνίσταται στην ύπαρξη επιπλέον εξίσωσης για τον υπολογισμό της εποχικής συνιστώσας της χρονοσειράς.

Για την ανάπτυξη της μεθόδου αυτής θεωρούμε το Πολλαπλασιαστικό Μοντέλο για την αναπαράσταση των δεδομένων μέσω των συνιστωσών. Y = L x S Μέσα στο L "κρύβεται" η τάση, η κυκλικότητα και η τυχαιότητα. Προκειμένου να κάνουμε πρόβλεψη με αυτό το μοντέλο χρειαζόμαστε τέσσερις εξισώσεις:

- Η εκθετικά εξομαλυνθείσα σειρά είναι η εξής:

$$S_t = a\frac{X_t}{I_{t-L}} + (1-a)(S_{t-1} + b_{t-1}) \qquad 4,10$$

- Η εκτίμηση της εποχικότητας είναι η εξής:

$$I_t = \beta \frac{X_t}{S_t} + (1-\beta) I_{t-L} \qquad 4,11$$

όπου I είναι ο παράγοντας προσαρμογής της εποχικότητας και το s είναι το μήκος της εποχικότητας.

- Η εκτίμηση της τάσης παραμένει η ίδια με εκείνη της μεθόδου διπλής εκθετικήςεξομάλυνσης:

$$Tt = \beta\ (Lt - Lt{-}1\ ) + (1 - \beta) Tt{-}1 \qquad 4,12$$

- Προβλέποντας p περιόδους στο μέλλον έχουμε:

$$Y_{t}{+}p = (L_{t} + p\ T_{t}\ ) St{-}s{+}p \qquad 4,13$$

Με την πρώτη εξίσωση γίνονται επίκαιρες οι εξομαλυνθείσες τιμές της σειράς. Το L δεν εμπεριέχει την εποχικότητα. Στην εξίσωση αυτή το Yt διαιρείται δια του St-s που προσαρμόζει τις αρχικές παρατηρήσεις για την εποχικότητα και αναιρεί τις επιδράσεις της, όσο καλύτερα αυτές μπορεί να μετρηθούν από τη χρονολογική σειρά.

Η δεύτερη εξίσωση δίνει την εκτίμηση της εποχικής συνιστώσας Yt/Lt πολλαπλασιασμένη επί τη σταθερά γ συν την παλαιά εποχική εκτίμηση St-s πολλαπλασιασμένη επί (1–γ). Επομένως, η επικαιροποίηση των εποχικών εκτιμήσεων είναι από μόνη της μια διαδικασία εκθετικής εξομάλυνσης. Επίσης, το Yt διαιρείται με Lt, προκειμένου να εκφραστεί η τιμή ως δείκτης παρά ως απόλυτο μέγεθος. Αυτό επιτρέπει την εύρεση του μέσου όρου των νέων εποχικών εκτιμήσεων με βάση τον εποχικό δείκτη της προηγούμενης περιόδου.

Η τρίτη εξίσωση εκφράζει τη σύγχρονη τιμή της συνιστώσας της τάσης, που επιτυγχάνεται με τη συνηθισμένη διαδικασία εκθετικής εξομάλυνσης. Τέλος, μετά από αυτή την εξίσωση λαμβάνουμε την εξίσωση για τις μελλοντικές περιόδους. Η διαφορά είναι ότι αυτή η εκτίμηση για τη μελλοντική περίοδο t+p πολλαπλασιάζεται επί St-s+p. Αυτός είναι ο

τελικός διαθέσιμος εποχικός δείκτης και αποτελεί την προσαρμογή της πρόβλεψης για εποχικότητα. Οι αρχικές τιμές είναι S1 = 1, T1 = 0, L1 = Y1. Οι παράμετροι α, β, γ ορίζονται με τέτοιο τρόπο, ώστε το MSE να γίνει ελάχιστο.

4.2 Box-Jenkins Method

Η μέθοδος Box-Jenkins προτάθηκε από τους George Box και Gwilym Jenkins στο βιβλίο Time Series Analysis: Forecasting and Control. Η προσέγγιση ξεκινά με την υπόθεση ότι η διαδικασία που δημιούργησε τις χρονοσειρές μπορεί να προσεγγιστεί χρησιμοποιώντας ένα μοντέλο ARMA εάν είναι στατικό ή ένα μοντέλο ARIMA εάν είναι μη στατικό. Η διαδικασία για την κατασκευη ενός στοχαστικού μοντέλου και ότι πρόκειται για μια επαναληπτική προσέγγιση που αποτελείται από τα ακόλουθα 3 βήματα:

1. Ταυτοποίηση: Χρησιμοποιήστε τα δεδομένα και όλες τις σχετικές πληροφορίες για να επιλέξετε μια υποκατηγορία μοντέλου που μπορεί να συνοψίσει καλύτερα τα δεδομένα.

2. Εκτίμηση: Χρησιμοποιήστε τα δεδομένα για να εκπαιδεύσετε τις παραμέτρους του μοντέλου (δηλαδή τους συντελεστές).

3. Διαγνωστικός έλεγχος: Αξιολογήστε το προσαρμοσμένο μοντέλο στο πλαίσιο των διαθέσιμων δεδομένων και ελέγξτε για περιοχές όπου το μοντέλο μπορεί να βελτιωθεί.

Είναι μια επαναληπτική διαδικασία, έτσι ώστε καθώς αποκτώνται νέες πληροφορίες κατά τη διάρκεια της διάγνωσης, μπορείτε να επιστρέψετε στο βήμα 1 και να τις ενσωματώσετε σε νέες κατηγορίες μοντέλων. Ας ρίξουμε μια ματιά σε αυτά τα βήματα με περισσότερες λεπτομέρειες.

4.2.1 Ταυτοποίηση

Το βήμα αναγνώρισης κατανέμεται περαιτέρω σε: Αξιολογήστε εάν οι χρονοσειρές είναι στάσιμες και, αν όχι, πόσες διαφορές απαιτούνται για να γίνει στάσιμη. Προσδιορίστε τις παραμέτρους ενός μοντέλου ARMA για τα δεδομένα

4.2.2 Differencing

Ακολουθούν μερικές συμβουλές κατά την αναγνώριση. Δοκιμές ρίζας μονάδας. Χρησιμοποιήστε στατιστικές δοκιμές μονάδας ρίζας στις χρονοσειρές για να προσδιορίσετε αν είναι σταθερή ή όχι. Επαναλάβετε μετά από κάθε γύρο διαφοροποίησης. Αποφύγετε την υπερβολική διάκριση. Η διαφοροποίηση των Χρονοσειρών περισσότερο από ό, τι απαιτείται μπορεί να έχει ως αποτέλεσμα την προσθήκη επιπλέον σειριακού συσχετισμού και πρόσθετης πολυπλοκότητας.

4.2.3 Configuring AR and MA

Μπορούν να χρησιμοποιηθούν δύο διαγνωστικά διαγράμματα για να βοηθήσουν στην επιλογή των παραμέτρων p και q των ARMA ή ARIMA. Είναι:

- Λειτουργία αυτοσυσχέτισης (ACF). Το διάγραμμα συνοψίζει τη συσχέτιση μιας παρατήρησης με τις τιμές υστέρησης. Ο άξονας x δείχνει την υστέρηση και ο άξονας y δείχνει τον συντελεστή συσχέτισης μεταξύ -1 και 1 για αρνητική και θετική συσχέτιση.

- Λειτουργία μερικής αυτοσυσχέτισης (PACF). Το διάγραμμα συνοψίζει τους συσχετισμούς για μια παρατήρηση με τιμές υστέρησης που δεν λαμβάνονται υπόψη από προηγούμενες παρατηρήσεις με καθυστέρηση. Και τα δύο γραφήματα σχεδιάζονται ως διαγράμματα ράβδων που δείχνουν τα διαστήματα εμπιστοσύνης 95% και 99% ως οριζόντιες γραμμές. Οι μπάρες που διασχίζουν αυτά τα διαστήματα εμπιστοσύνης είναι επομένως πιο σημαντικές και αξίζει να σημειωθούν.

4.2.4 Εκτίμηση

Περιλαμβάνει τη χρήση αριθμητικών μεθόδων για την ελαχιστοποίηση ενός όρου απώλειας ή σφάλματος. Δεν θα αναφερθούμε στις λεπτομέρειες της εκτίμησης των παραμέτρων του μοντέλου, καθώς αυτές οι λεπτομέρειες αντιμετωπίζονται από την επιλεγμένη βιβλιοθήκη ή το εργαλείο. Θα συνιστούσα να αναφερθώ σε ένα βιβλίο για να κατανοήσουμε καλύτερα το πρόβλημα βελτιστοποίησης που θα λυθεί από τα μοντέλα ARMA και ARIMA και μεθόδους βελτιστοποίησης όπως το BFGS περιορισμένης μνήμης που χρησιμοποιείται για την επίλυσή του.

4.2.5 Diagnostic Checking

Η ιδέα των διαγνωστικών ελέγχων είναι να ψάξουν για στοιχεία που αποδεικνύουν ότι το μοντέλο δεν είναι ένα κατάλληλο για τα δεδομένα. Δύο χρήσιμοι τομείς για τη διερεύνηση των διαγνωστικών είναι το Overfitting και τα Residuals Errors. Ας τα δουμε παρακάτω.

4.2.5.1 Overfitting

Ο πρώτος έλεγχος είναι να ελέγξετε αν το μοντέλο υπερβαίνει τα δεδομένα. Γενικά, αυτό σημαίνει ότι το μοντέλο είναι πιο περίπλοκο από ό, τι χρειάζεται και εχει τυχαίο θόρυβο στα δεδομένα του. Αυτό είναι ένα πρόβλημα για τις προβλέψεις χρονολογικών σειρών, επειδή επηρεάζει αρνητικά την ικανότητα του μοντέλου να γενικεύσει, με αποτέλεσμα την κακή απόδοση των προβλέψεων σε δεδομένα εκτός δείγματος. Πρέπει να δοθεί ιδιαίτερη προσοχή τόσο στα δεδομενα που είναι εντος δειγματος όσο και εκτός και αυτό απαιτεί τον προσεκτικό σχεδιασμό μιας στιβαρής δοκιμαστικής πλεξούδας για την αξιολόγηση μοντέλων

4.2.5.2 Residual Errors

Τα υπολείμματα πρόβλεψης παρέχουν μια μεγάλη ευκαιρία για διαγνωστικο ελεγχο. Μια ανασκόπηση της κατανομής των σφαλμάτων μπορεί να βοηθήσει στην εξάλειψη της προκατάληψης στο μοντέλο. Τα σφάλματα από ένα ιδανικό μοντέλο θα μοιάζουν με λευκό θόρυβο, δηλαδή μια κατανομή Gauss με μέση τιμή μηδέν και συμμετρική διακύμανση. Για

αυτό, μπορείτε να χρησιμοποιήσετε γραφικές παραστάσεις πυκνότητας, ιστογράμματα και γραφήματα Q-Q που συγκρίνουν την κατανομή σφαλμάτων με την αναμενόμενη κατανομή.

Μια μη-Gaussian διανομή μπορεί να προτείνει μια ευκαιρία για την προεπεξεργασία δεδομένων. Μια λοξή κατανομή ή μη μηδενικός μέσος όρος μπορεί να υποδηλώνει μια προκατάληψη στις προβλέψεις που μπορεί να είναι σωστές. Επιπλέον, ένα ιδανικό μοντέλο δεν θα άφηνε χρονική δομή στη χρονολογική σειρά των υπολειμμάτων πρόβλεψης. Αυτά μπορούν να ελεγχθούν δημιουργώντας γραφήματα ACF και PACF των υπολειπόμενων χρονοσειρών σφάλματος. Η παρουσία σειριακής συσχέτισης στα υπολειπόμενα σφάλματα υποδηλώνει περαιτέρω ευκαιρίες για τη χρήση αυτών των πληροφοριών στο μοντέλο.

4.2.6 Αυτοσυσχέτιση και Μερική Αυτοσυσχέτιση

Ο συντελεστής αυτοσυσχέτισης είναι ένας από τους πιο σημαντικούς στατιστικούς δείκτες ο οποίος χρησιμοποιείται ευρέως στην ανάλυση χρονοσειρών για την διαπίστωση ύπαρξης τυχαιότητας ή μη στην χρονοσειρά. Η αυτοσυσχέτιση (autocorrelation) j-οστής τάξης ρjt της τυχαίας μεταβλητής Yt με μια προηγούμενη χρονική εκδοχή της Yt-j ορίζεται για κάθε t ως εξής:

$$\mathrm{R}_{YY}(t_i, t_j) = \mathbb{E}[\, Y_i \, Y_j] \qquad 4,14$$

Το σύνολο τιμών της αυτοσυσχέτισης είναι το [-1,1]. Η αυτοσυσχέτιση είναι ένας καταλυτικής σημασίας στατιστικός δείκτης στη μελέτη χρονοσειρών, διότι έτσι δίνεται ένα μέτρο για τον βαθμό της σχέσης μεταξύ των δύο μεταβλητών. Ειδικότερα, φτιάχνοντας το γράφημα της αυτοσυσχέτισης συναρτήσει της καθυστέρησης j, το οποίο ονομάζεταισυνάρτηση αυτοσυσχέτισης (autocorrelation function- ACF) δύναται να αποσαφηνιστούν στοιχεία για τα χαρακτηριστικά της χρονοσειράς. Αν η τιμή του ρjt ισούται περίπου με +1 ή -1, τότε οι παρατηρήσεις Yt και Yt-j είναι ισχυρά συσχετισμένες. Ενώ, αν ρjt ισούται με 0, τότε εύκολα προκύπτει ότι οι παρατηρήσεις Yt και Yt-j είναι ασυσχέτιστες. Συνεπώς, με γνώμονα την αυτοσυσχέτιση μπορούν να μελετηθούν τα ποιοτικά χαρακτηριστικά των χρονοσειρών, όπως η εποχικότητα και η στασιμότητα.

Όπως και η συνάρτηση αυτοσυσχέτισης, η συνάρτηση μερικής αυτοσυσχέτισης (partial autocorrelation function-PACF) αποτελεί πολύτιμη πηγή πληροφόρησης σχετικά με τα χαρακτηριστικά της αλληλεξάρτησης που δημιουργεί μια στοχαστική διαδικασία. Οι συντελεστές μερικής αυτοσυσχέτισης μετρούν το βαθμό της σχέσης μεταξύ των Yt και Yt-k όταν οι επιδράσεις όλων των άλλων χρονικών υστερήσεων 1,2,3, ...,k-1 έχουν αφαιρεθεί. Ο συντελεστής μερικής αυτοσυσχέτισης τάξης k συμβολίζεται με αk και μπορεί να υπολογισθεί εφαρμόζοντας τη μέθοδο της πολλαπλής γραμμικής παλινδρόμησης με εξαρτημένη μεταβλητή την Yt και ανεξάρτητες μεταβλητές τις Yt-1, ... , Yt-k :

$$Y_t = b_0 + b_1 Y_{t-1} + \ldots + b_k Y_{t-k} \qquad 4,15$$

4.2.7 Αυτοπαλινδρούμενη χρονοσειρά (AR)

Το μοντέλο αυτόματης παλινδρόμησης είναι το απλούστερο μοντέλο για χρονοσειρές, προέρχεται από την ιδέα ότι οποιαδήποτε τιμή μιας χρονοσειράς μπορεί να προβλεφθεί βάσει των προηγούμενων καταγεγραμμένων τιμών. Μαθηματικά, αυτή η διατύπωση σημαίνει ότι η μελλοντική τιμή της χρονοσειράς, yt, θα είναι ο γραμμικός συνδυασμός των προηγούμενων τιμών με μια τομή (3,5, 3,6), με φi∈ [1..p] το βάρος κάθε παρατήρησης. Η παράμετρος p του AR (p) καθορίζει τον αριθμό των καθυστερήσεων που θα χρησιμοποιηθούν για την πρόβλεψη της τιμής του yt.

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} \qquad 4,16$$

$$y_t = c + \sum_{k=1}^{p} \phi_k y_{t-k} \qquad 4,17$$

4.2.8 Χρονοσειρές Κινητού Μέσου (MA)

Οι χρονοσειρές κινητού μέσου είναι χρήσιμες για περιγραφή φαινομένων στα οποία τα γεγονότα παράγουν ένα άμεσο αποτέλεσμα, η επίδραση του οποίου όμως δεν σταματά εκεί αλλά διαρκεί, αν και το ίδιο το γεγονός σταματάει να υφίσταται. Οι διαδικασίες κινητού

μέσου έχουν χρησιμοποιηθεί σε πολλούς τομείς και ιδιαίτερα στην οικονομετρία. Για παράδειγμα, η οικονομία επηρεάζεται από μια απεργία όχι μόνο την στιγμή που πραγματοποιείται, αλλά και τους επόμενους μήνες αν και έχει λήξει.

4.2.9 ARMA

Σε κάποιες περιπτώσεις υποθέτουμε πως εξωγενείς παράγοντες σε προηγούμενους χρόνους μπορούν επίσης να επηρεάζουν τη μεταβλητή της χρονοσειράς τη χρονική στιγμή t. Συμπεριλαμβάνοντας και αυτό το μέρος που λέγεται μέρος κινούμενου μέσου (moving average), το γενικό γραμμικό μοντέλο για την πρόβλεψη στάσιμης χρονοσειράς είναι το αυτοπαλινδρομούμενο μοντέλο κινούμενου μέσου (AutoRegressive Moving Average, ARMA) που δίνεται ως

$$y_t = \delta + \varphi_1 y_{t-1} + \ldots + \varphi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \ldots - \theta_q \varepsilon_{t-q} \qquad 4,18$$

Το αυτοπαλινδρομούμενο μέρος (AR) είναι τάξης p και το μέρος του κινούμενου μέσου (MA) είναι τάξης q και το μοντέλο συμβολίζεται ARMA(p,q).

Το μοντέλο ARMA(p,q) είναι ο συνδυασμός των προαναφερθέντων AR(p) και MA(q), μαθηματικά το μοντέλο είναι απλώς το άθροισμα αυτών των μοντέλων όπως περιγράφεται από τον τύπο 4,18 . Αλλά η αδυναμία αυτών των μοντέλων είναι ότι υποθέτουν ότι οι χρονοσειρές που δουλεύουμε είναι στάσιμη, κάτι που δεν συμβαίνει στις περισσότερες πραγματικές περιπτώσεις. Επομένως, πρέπει να βεβαιωθούμε ότι οποιαδήποτε χρονοσειρά που μοντελοποιείται με αυτή τη συνάρτηση πρέπει να είναι σταθερή, για το σκοπό αυτό, χρησιμοποιούμε τη λειτουργία διαφοροποίησης για να σταθεροποιήσουμε τη σειρά. Αυτή η λειτουργία χειρίζεται από το μοντέλο ARIMA που γενικεύει το μοντέλο ARMA για μη στάσιμες χρονοσειρές.

4.2.10 ARIMA Model

Στην πλειονότητά τους οι χρονοσειρές δεν χαρακτηρίζονται ως στάσιμες

διαδικασίες. Οι χρονοσειρές βολεύει να είναι στάσιμες διότι είναι εύκολα επεξεργάσιμες. Όταν μια χρονοσειρά μετατρέπεται σε στάσιμη, χρησιμοποιώντας τις πρώτες διαφορές η σειρά ονομάζεται ολοκληρώσιμη πρώτης τάξης και συμβολίζεται με I(1). Εάν η χρονοσειρά μετατρέπεται σε στάσιμη χρησιμοποιώντας τις δεύτερες διαφορές, είναι ολοκληρώσιμη δεύτερης τάξης και συμβολίζεται με I(2). Γενικεύοντας, εάν d είναι ο αριθμός των διαφορών που μετατρέπει μια σειρά σε στάσιμη, η σειρά ονομάζεται ολοκληρώσιμη d τάξεως και συμβολίζεται με I(d). Με χρήση του τελεστή ολίσθησης οι πρώτες διαφορές ορίζονται ως:

$$\boldsymbol{y_t - y_{t-1} = (1 - B)y_t = \Delta y_t} \qquad 4,19$$

Τα στοχαστικά μοντέλα ARIMA είναι εύχρηστα καθώς ξεπερνούν το εμπόδιο της στασιμότητας που αντιμετώπιζαν τα μοντέλα ARMA, καθώς με την διαφόριση γίνεται προσπάθεια να εξαλειφθεί η στασιμότητα. Η λέξη ARIMA (Autoregressive Integrated Moving Average) ελληνιστί μεταφράζεται ως αυτοπαλίνδρομα μοντέλα κινητού μέσου και ανάλογα την τάξη, ARIMA(p, d, q) (Yao & Herbert, 2009). Επομένως, μια ARIMA(p, d, q) διαδικασία, είναι μια διαδικασία η οποία «διαφορίζεται» d φορές εξάγει μια ARMA(p, q) διαδικασία. Για ένα ολοκληρωμένο μοντελο ARIMA(p, d, q) ισχύει ότι p είναι η τάξη του αυτοπαλίνδρομου μοντέλο, d η τάξη της διαφόρισης για την επίτευξη της στασιμότητας και q η τάξη του κινητού μέσου όρου μοντέλου.

Ορίζεται μαθηματικά ως (Pfaff, 2008):

$$\boldsymbol{\phi(B)(1 - B)^d y_1 = \delta + \theta(B)\varepsilon_t} \qquad 4,20$$

4.3 FB Prophet

Δεν είναι δυνατή η επίλυση όλων των προβλημάτων πρόβλεψης με την ίδια μέθοδο. Υπάρχουν διάφορες μέθοδοι ανάλογα με την φύση του προβλήματος. Η Prophet είναι μία βελτιστοποιημένη μέθοδος για επιχειρηματικές προβλέψεις, οι οποίες συνήθως έχουν κάποιο από τα παρακάτω χαρακτηριστικά:

• Ωριαίες, καθημερινές ή εβδομαδιαίες παρατηρήσεις με τουλάχιστον μερικούς μήνες κατά προτίμηση ένα χρόνο) ιστορίας.

• Εποχικότητα της ανθρώπινης κλίμακας, όπως ποια ημέρα της εβδομάδας είναι και ποια ημερομηνία του έτους είναι.

• Αργίες, που συμβαίνουν σε ακανόνιστα διαστήματα, τα οποία όμως είναι γνωστά εκ των προτέρων (πχ. Χριστούγεννα, Πάσχα).

• Έναν λογικό αριθμό ελλιπών παρατηρήσεων ή εσφαλμένων καταχωρήσεων.

• Ιστορικές μεταβολές τάσεων, για παράδειγμα εξαιτίας λανσαρίσματος νέων προϊόντων.

• Τάσεις, οι οποίες δεν είναι γραμμικές καμπύλες ανάπτυξης, αλλά η τάση αγγίζει ένα όριο (φτάνει σε κορεσμό) (Taylor and Letham (2017)).

Ουσιαστικά, η Prophet είναι μία διαδικασία αθροιστικής παλινδρόμησης με τέσσερα βασικά στοιχεία:

$$\mathbf{y(t)} = \mathbf{g(t)} + \mathbf{s(t)} + \mathbf{h(t)} + \boldsymbol{\epsilon} * \mathbf{g(t)} \qquad 4{,}21$$

Μοντελοποιεί την τάση, το οποίο περιγράφει μία μακροπρόθεσμη αύξηση ή μείωση των δεδομένων. Η Prophet ενσωματώνει δύο μοντέλα τάσεων, ένα μοντέλο λογικής ανάπτυξης (κορεσμού) και ένα μερικώς γραμμικό μοντέλο, ανάλογα με το είδος του προβλήματος πρόβλεψης. Η Prophet εντοπίζει αυτόματα τις αλλαγές στις τάσεις, επιλέγοντας τα σημεία αλλαγής από τα δεδομένα.

- g(t): piecewise linear or logistic growth curve for modelling non-periodic changes in time series
- s(t): Μοντελοποιεί την εποχικότητα με τη χρήση σειρών Fourier, το οποίο περιγράφει πως τα δεδομένα επηρεάζονται από εποχιακούς παράγοντες, όπως η χρονική στιγμή στο έτος.
- h(t): Μοντελοποιεί τις επιπτώσεις των διακοπών ή των μεγάλων γεγονότων που επηρεάζουν τις χρονοσειρές των επιχειρήσεων (πχ. Χριστούγεννα, Black Friday,

τελικός Champions league, κλπ). Η λίστα των σημαντικών αργιών ορίζεται χειροκίνητα από τον χρήστη.

- ϵ: Αντιπροσωπεύει τον όρο του μη αναστρέψιμου σφάλματος.

Χρησιμοποιώντας το χρόνο ως παλινδρόμηση, η Prophet προσπαθεί να χωρέσει πολλές γραμμικές και μη γραμμικές συναρτήσεις του χρόνου ως συστατικά. Η μοντελοποίηση της εποχικότητας ως πρόσθετο συστατικό είναι η ίδια προσέγγιση με την εκθετική εξομάλυνση στην τεχνική Holt-Winters. Στην πραγματικότητα, διαμορφώνουμε το πρόβλημα πρόβλεψης ως μια άσκηση προσαρμογής καμπύλης παρά να εξετάζουμε ρητά την χρονική εξάρτηση κάθε παρατήρησης μέσα σε μια χρονολογική σειρά.

Η μέθοδος Prophet αξιολογεί αυτόματα τις επιδόσεις πρόβλεψης και προειδοποιεί όπου απαιτείται χειροκίνητη παρέμβαση. Ένας από τους ευκολότερους τρόπους αξιολόγησης είναι να ορίσουμε ένα επίπεδο με μερικές απλές μεθόδους πρόβλεψης (π.χ. εποχιακή αφελής τάση, μέσος όρος δείγματος, κλπ.). Είναι χρήσιμο να συγκρίνετε τις απλές και προηγμένες μεθόδους πρόβλεψης για να είστε σε θέση να προσδιορίσετε εάν μπορεί να επιτευχθεί πρόσθετη απόδοση χρησιμοποιώντας ένα πιο πολύπλοκο μοντέλο. Μερικές φορές, ίσως είναι καλύτερο να χρησιμοποιήσετε απλώς μία απλοϊκή μέθοδο.

Ενώ εγκαταλείπουμε ορισμένα σημαντικά συμπεράσματα πλεονεκτήματος από τη χρήση ενός γενετικού μοντέλου όπως το ARIMA, αυτή η διατύπωση παρέχει μια σειρά πρακτικών πλεονεκτημάτων:

• Ευελιξία: Μπορούμε να προσαρμόσουμε εύκολα την εποχικότητα με πολλές περιόδους και να αφήσουμε τον αναλυτή να κάνει διαφορετικές παραδοχές σχετικά με τις τάσεις.

• Σε αντίθεση με τα μοντέλα ARIMA, οι μετρήσεις δεν χρειάζεται να βρίσκονται σε απόσταση μεταξύ τους και δεν χρειάζεται να παρεμβάλλουμε τιμές που λείπουν π.χ. από την αφαίρεση των ακραίων τιμών.

• Η τοποθέτηση είναι πολύ γρήγορη, επιτρέποντας στον αναλυτή να διερευνήσει διαδραστικά πολλές προδιαγραφές μοντέλου, για παράδειγμα σε μια εφαρμογή Shiny (Chang et al. 2015).

• Το μοντέλο προβλεψεων έχει εύκολα ερμηνεύσιμες παραμέτρους που μπορούν να αλλάξουν από τον αναλυτή για να επιβάλουν παραδοχές στην πρόβλεψη. Επιπλέον, οι αναλυτές έχουν συνήθως εμπειρία με παλινδρόμηση και είναι εύκολα σε θέση να επεκτείνουν το μοντέλο ώστε να συμπεριλάβει νέα στοιχεία.

Κεφάλαιο 5. ΜΕΤΡΑ ΑΠΟΔΟΣΗΣ ΠΡΟΒΛΕΨΗΣ ΧΡΟΝΟΣΕΙΡΩΝ

Τα μέτρα απόδοσης προβλέψεων Χρονοσειρών παρέχουν μια σύνοψη της ικανότητας του μοντέλου πρόβλεψης που έκανε τις προβλέψεις. Υπάρχουν πολλά διαφορετικά μέτρα απόδοσης, οπότε θα τα αναλύσουμε και θα δούμε στην πορεία πιο θα διαλέξουμε για την επιλογή του μοντέλου μας.

5.1.1 Forecast Error (or Residual Forecast Error)

Το σφάλμα πρόβλεψης υπολογίζεται ως η αναμενόμενη τιμή μείον την προβλεπόμενη τιμή. Αυτό ονομάζεται το εναπομένον σφάλμα της πρόβλεψης.

$$\text{forecast error} = \text{expected value} - \text{predicted value} \qquad 5,1$$

Το σφάλμα πρόβλεψης μπορεί να υπολογιστεί για κάθε πρόβλεψη, παρέχοντας μια χρονική σειρά σφαλμάτων πρόβλεψης. Οι μονάδες του σφάλματος πρόβλεψης είναι ίδιες με τις μονάδες της πρόβλεψης. Ένα σφάλμα πρόβλεψης μηδέν δεν δείχνει κανένα σφάλμα ή τέλεια δεξιότητα για αυτήν την πρόβλεψη.

5.1.2 Mean Forecast Error

Το μέσο σφάλμα πρόβλεψης υπολογίζεται ως ο μέσος όρος των τιμών των σφαλμάτων πρόβλεψης.

$$\text{mean forecast error} = \text{mean}(\text{forecast error}) \qquad 5,2$$

Τα σφάλματα των προβλεψεων μπορεί να είναι θετικά και αρνητικά. Αυτό σημαίνει ότι όταν υπολογίζεται ο μέσος όρος αυτών των τιμών, ένα ιδανικό μέσο σφάλμα πρόβλεψης θα είναι μηδέν. Μια μέση τιμή σφάλματος πρόβλεψης εκτός από το μηδέν υποδηλώνει την τάση του μοντέλου να υπερβαίνει την πρόβλεψη (αρνητικό σφάλμα) ή κατω από την πρόβλεψη (θετικό σφάλμα. Το σφάλμα πρόβλεψης μπορεί να υπολογιστεί απευθείας ως ο

μέσος όρος των τιμών πρόβλεψης πρόβλεψη του μηδέν, ή ένας πολύ μικρός αριθμός κοντά στο μηδέν, δείχνει ένα αμερόληπτο μοντέλο..

5.1.3 Mean Absolute Error

Το Μέσο Απόλυτο Σφάλμα αποτελεί ένα μέτρο διαφοράς μεταξύ δύο συνεχών μεταβλητών. Η μαθηματική φόρμουλα για τον υπολογισμό του μέσου απόλυτου σφάλματος αναφέρεται ακολούθως, όπου το n είναι το σύνολο των σημείων και (xi,yi) οι συντεταγμένες του σημείου.

$$Mean\ absolute\ error: MAE = mean(|et|) \qquad 5,3$$

5.1.4 Mean Squared Error

Το Μέσο Τετραγωνικό Σφάλμα μετρά τον μέσο όρο των τετραγώνων σφαλμάτων, δηλαδή τη μέση

$$\mathrm{MSE} = \frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2 \qquad 5,4$$

$$MSE = 1$$

τετραγωνική διαφορά μεταξύ των εκτιμώμενων τιμών από τις πραγματικές τιμές ενός συνόλου δεδομένων που έχουν χρησιμοποιηθεί για την δημιουργία ενός μοντέλου μηχανικής μάθησης-μάθησης σε βάθος. Το Μέσο Τετραγωνικό Σφάλμα αποτελεί ένα μέτρο εκτίμησης απόδοσης, βγάζει μόνο θετικές τιμές και οι καλύτερες τιμές είναι όσο πιο κοντά στο μηδέν. Παρακάτω παρουσιάζεται η μαθηματική φόρμουλα για τον υπολογισμό του μέσου τετραγωνικού σφάλματος. Όπου το n είναι το σύνολο δεδομένων, Y είναι οι πραγματικές τιμές και Ŷi είναι οι εκτιμώμενες τιμές.

5.1.5 Root Mean Squared Error

Το μέσο τετραγωνικό σφάλμα που περιγράφεται παραπάνω βρίσκεται στις τετραγωνικές μονάδες των προβλέψεων. Μπορεί να μετατραπεί πίσω στις αρχικές μονάδες

των προβλέψεων λαμβάνοντας την τετραγωνική ρίζα του μέσου τετραγώνου βαθμού σφαλμάτων. Αυτό ονομάζεται τετραγωνικό σφάλμα root mean, ή RMSE.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2} \qquad 5,5$$

Οι τιμές σφάλματος RMES βρίσκονται στις ίδιες μονάδες με τις προβλέψεις. Όπως με το μέσο τετράγωνο σφάλμα, ένα RMSE μηδέν υποδεικνύει κανένα σφάλμα

5.1.6 Διαγνωστικός έλεγχος

Σε τελικό στάδιο εφόσον το επιλεγμένο μοντέλο θεωρείται το βέλτιστο, είναι επιτακτική ανάγκη να επιβεβαιωθεί η επάρκεια του μοντέλου. Αυτό επιτυγχάνεται με την εξέταση των υπολοίπων (σφαλμάτων) ώστε να διαπιστωθεί εάν ακολουθούν αυτά κάποιο πρότυπο.

Τα υπόλοιπα, λοιπόν, ενός καλού μοντέλου πρόβλεψης πρέπει να αποτελούν έναν «λευκό θόρυβο» ενώ συγχρόνως οι ACF και PACF των υπολοίπων δεν πρέπει να παρουσιάζουν στατιστικά σημαντικές αυτοσυσχετίσεις και αντίστοιχα μερικές αυτοσυσχετίσεις. Για να εξετασθούν συνολικά οι συντελεστές αυτοσυσχέτισης των υπολοίπων χρησιμοποιούνται διάφοροι στατιστικοί και εμπειρικοί έλεγχοι. Αναφορικά κάποια από αυτά είναι ο στατιστικός δείκτης Q * (Ljung-Box), το γράφημα Q-Q plot και ο έλεγχος Anderson-Darling. Όσον αφορά το δείκτη Q * , αν η τιμή του δεν είναι στατιστικά σημαντική τα υπόλοιπα θεωρούνται μια σειρά λευκού θορύβου. Σε περίπτωση που τα υπόλοιπα δεν αποτελούν σειρά λευκού θορύβου τότε το μοντέλο είναι ανεπαρκές και πρέπει να πραγματοποιηθεί περαιτέρω εξέταση.

Γενικεύοντας, το πρότυπο που ακολουθούν οι στατιστικά σημαντικοί συντελεστές αυτοσυσχέτισης και μερικής αυτοσυσχέτισης των υπολοίπων, υποδεικνύουν άμεσα τον τρόπο βελτίωσης του μοντέλου. Παραδείγματος χάρη, για στατιστικά σημαντικές τιμές, για εποχιακές καθυστερήσεις, προτείνεται η προσθήκη μιας εποχικής συνιστώσας. Επιπλέον, στατιστικά σημαντικές τιμές για μικρές καθυστερήσεις υποδεικνύουν κατά κανόνα την αύξηση των μη εποχιακών AR ή MA

συνιστωσών του μοντέλου. Είθισται τα μοντέλα με τις μικρότερες AIC τιμές να έχουν υπόλοιπα λευκού θορύβου. Παραδόξως αλλά περιστασιακά, συμβαίνει να υιοθετούνται μοντέλα όχι με την μικρότερη AIC τιμή αλλά αυτά με τα «καλύτερα» υπόλοιπα (Κουγιουμτζής, 2004)

Κεφάλαιο 6. ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

Στο κεφάλαιο αυτό παρουσιάζεται η μεθοδολογία η οποία αναπτύχθηκε με στόχο την δημιουργία των μοντέλων προβλέψεων στην γλώσσα προγραμματισμού Python. Συγκεκριμένα, γίνεται εκτεταμένη αναφορά στην γλώσσα προγραμματισμού, στα εργαλεία και στα γραφικά περιβάλλοντα χρήστη που χρησιμοποιήθηκαν. Ακόμη, αναφέρονται αναλυτικά όλες οι βιβλιοθήκες που εγκαταστάθηκαν και ο λόγος χρήσης τους. .

6.1 Γλώσσα προγραμματισμού

Το οικοσύστημα Python αναπτύσσεται και μπορεί να γίνει η κυρίαρχη πλατφόρμα για την εφαρμοσμένη μηχανικη μάθηση. Το πρωταρχικό σκεπτικό για την υιοθέτηση της Python για προβλέψεις Χρονοσειρών είναι επειδή είναι μια γλώσσα προγραμματισμού γενικής χρήσης που μπορείτε να χρησιμοποιήσετε τόσο στην ερευνα οσο και στην παραγωγή .

Οι λόγοι για τον οποίο χρησιμοποιήθηκε η Python είναι, ότι αποτελεί μια γλώσσα προγραμματισμού υψηλού επιπέδου ανοικτού κώδικα, η οποία παρέχει στον χρήστη αναγνωσιμότητα του κώδικα και κυρίως ευκολία στην χρήση της. Διακρίνεται ακόμη για τον μεγάλο αριθμό βιβλιοθηκών της οι οποίες διευκολύνουν ιδιαίτερα τον χρήστη, όπως μαθηματικά μοντέλα, αλγορίθμους μάθησης σε βάθος, και είναι πολύ εύκολες στην διασύνδεση τους με γραφικά περιβάλλοντα χρήστη και εργαλεία.

6.2 Εργαλεία για την ανάπτυξη του μοντέλου

Η δημιουργία των μοντέλων προβλέψεων χρονοσειρών, πραγματοποιήθηκε με την χρήση των εργαλείων Anaconda Navigator. Ο κύριος λόγος χρήσης αυτών των εργαλείων είναι ότι το μοντέλο μπορεί μέσω αυτών των εργαλείων να δημιουργηθεί σε ποικίλα εικονικά περιβάλλοντα και με την δυνατότητα χρήσης παράλληλου προγραμματισμού μέσω της υπολογιστικής ισχύς της κάρτας γραφικών. Η παραπάνω διαδικασία δίνει το πλεονέκτημα στον χρήστη να χρησιμοποιεί μεγάλες σε όγκο βάσεις δεδομένων και πολύπλοκους αλγορίθμους μάθησης σε βάθος, χωρίς να χρειάζεται να εγκαθιστά και να επανεγκαθιστά

διάφορες εκδόσεις βιβλιοθηκών, προσφέροντας εκπαίδευση, επαλήθευση και δοκιμή του μοντέλου σε πολύ μικρό χρονικό διάστημα, χωρίς να υπάρχει η ανάγκη αγοράς υπέρογκων υπολογιστικών συστημάτων με υψηλό κόστος και χαμηλότερη υπολογιστική ισχύ.

Το εργαλείο Anaconda.Navigator αποτελεί εργαλείο ανοιχτού κώδικα το οποίο επιτρέπει στον χρήστη την δημιουργία και την διαχείριση εικονικών περιβαλλόντων και την ευκολία εγκατάστασης πακέτων σε αυτά. Χρησιμοποιείται ευρέως από το πεδίο της επιστήμης της πληροφορικής, για την ανάπτυξη μοντέλων μηχανικής μάθησης και μάθησης σε βάθος, καθώς και για την ανάλυση και την επεξεργασία μεγάλων βάσεων δεδομένων.

5.3 Γραφικά περιβάλλοντα χρήστη

Για την ανάπτυξη των μοντέλων προβλέψεων χρονοσειρών χρησιμοποιήθηκαν τα περιβάλλοντα γραφικών χρήστη, Visual Studio Code και Spyder. Το Visual Studio Code είναι ένα ανοικτού κώδικα γραφικό περιβάλλον επεξεργασίας πηγαίου κώδικα, το οποίο δίνει την δυνατότητα στον χρήστη να αναπτύξει κώδικα σε ποικίλες γλώσσες προγραμματισμού με την δυνατότητα της εύκολής ενσωμάτωσης ποικίλων βιβλιοθηκών και πρόσθετων υπηρεσιών. Ακολούθως το Spyder είναι και αυτό ένα ανοικτού κώδικα περιβάλλον επεξεργασίας κώδικα αλλά αποκλειστικά για την γλώσσα προγραμματισμού Python. Το περιβάλλον Spyder δίνει την δυνατότητα στον χρήστη να εκτελεί γραμμή-γραμμή των πηγαίο κώδικα που έχει δημιουργήσει, εμφανίζοντας ταυτόχρονα τα αποτελέσματα για κάθε βήμα εκτέλεσης του. Έτσι ο χρήστης μπορεί να βλέπει αν το κάθε βήμα κώδικα που έχει αναπτύξει δουλεύει ή όχι και να εκτυπώνει μεμονωμένα διαγράμματα ή τμήματα κώδικα, χωρίς να χρειάζεται να επανεκτελέσει όλο τον κώδικα από την αρχή.

6.3 Βιβλιοθήκες

Όπως είδαμε στα προαναφερθέντα για την δημιουργία των μοντέλων προβλέψεων χρονοσειρών, χρησιμοποιήθηκαν η γλώσσα προγραμματισμού Python, το εργαλείο Anaconda Navigator καθώς και τα περιβάλλοντα επεξεργασίας πηγαίου κώδικα που χρησιμοποιήθηκαν για την ανάπτυξη του μοντέλου

Σε αυτό το υπό-κεφάλαιο αναφέρονται οι πιο σημαντικές βιβλιοθήκες που χρησιμοποιήθηκαν για την τελική δημιουργία του μοντέλου. Για την δημιουργία του μοντέλου χρησιμοποιήθηκαν ποικίλες βιβλιοθήκες για την γλώσσα προγραμματισμού Python, από αυτές κάποιες είναι μαθηματικές, άλλες χρησιμοποιούνται για την απεικόνιση διαγραμμάτων και οι πιο σημαντικές είναι βιβλιοθήκες για την δημιουργία μοντέλων μηχανικής μάθησης-μάθησης σε βάθος, καθώς και βιβλιοθήκες μηχανικής όρασης.

6.3.1 Python Libraries for Time Series

Το SciPy είναι ένα οικοσύστημα βιβλιοθηκών Python για μαθηματικά, επιστήμη και μηχανική. Είναι ένα πρόσθετο στο Python που θα χρειαστούμε για τα μοντέλα πρόβλεψης Χρονοσειρών. Δύο βιβλιοθήκες SciPy παρέχουν τη βάση για τους περισσότερους άλλους, η πρώτη είναι η NumPy3 για την παροχή αποτελεσματικών λειτουργιών συστοιχίας και η Matplotlib4 για τη σχεδίαση δεδομένων. Υπάρχουν και τρεις βιβλιοθήκες SciPy υψηλότερου επιπέδου που παρέχουν βασικά χαρακτηριστικά για την πρόβλεψη Χρονοσειρών σε Python. Πρόκειται για Pandas, Statsmodels και scikit-learning για διαχείριση δεδομένων, μοντελοποίηση Χρονοσειρών και μηχανική εκμάθηση αντίστοιχα. Ας ρίξουμε μια πιο προσεκτική ματιά σε κάθε σειρά.

6.3.2 Βιβλιοθήκη: Pandas

Η βιβλιοθήκη Pandas παρέχει εργαλεία υψηλής απόδοσης για να φορτώνουμε και να χειριζομαστε δεδομενα στην Python. Εχει χτιστει και απαιτεί το οικοσύστημα SciPy και χρησιμοποιεί κυρίως πίνακες NumPy κάτω από τα καλύμματα, αλλά παρέχει βολικές και εύχρηστες δομές δεδομένων, όπως DataFrame και Series για την αναπαράσταση δεδομένων. Η βιβλιοθήκη Pandas δίνει ιδιαίτερη έμφαση στην υποστήριξη δεδομένων Χρονοσειρών. Τα βασικά χαρακτηριστικά που σχετίζονται με την πρόβλεψη Χρονοσειρών περιλαμβάνουν:

- Το Series object για την αναπαράσταση μιας μονομεταβλητής χρονικής σειράς.
- Άμεσο χειρισμός ευρετηρίων ημερομηνίας-ώρας στα δεδομένα και εύρη ημερομηνιών.
- Μετασχηματισμοί όπως μετατόπιση, καθυστέρηση και πλήρωση.

•Μέθοδοι δειγματοληψίας, όπως δειγματοληψία, δειγματοληψία προς τα κάτω και συσσωμάτωση.

6.3.3 Βιβλιοθήκη: Statsmodels

Η βιβλιοθήκη Statsmodels παρέχει εργαλεία στατιστικής μοντελοποίησης. Είναι βασισμένο και απαιτεί το οικοσύστημα SciPy και υποστηρίζει δεδομένα με τη μορφή συστοιχιών NumPy και αντικειμένων της σειράς Pandas. Παρέχει μια σειρά στατιστικών μεθόδων δοκιμών και μοντελοποίησης, καθώς και εργαλεία αφιερωμένα στην ανάλυση Χρονοσειρών που μπορούν επίσης να χρησιμοποιηθούν για την πρόβλεψη. Τα βασικά χαρακτηριστικά του Statsmodels που σχετίζονται με τις προβλέψεις Χρονοσειρών περιλαμβάνουν:

- Στατιστικά τεστ για σταθερότητα, όπως το τεστ ριζικής μονάδας Augmented Dickey-Fuller.
- Οικόπεδα ανάλυσης Χρονοσειρών όπως η λειτουργία αυτοσυσχέτισης (ACF) και η λειτουργία μερικής αυτοσυσχέτισης (PACF).
- Μοντέλα γραμμικών χρονολογικών σειρών όπως AR, MA, ARMA και (ARIMA).

6.3.4 Βιβλιοθήκη: scikit-learn

Με τη βιβλιοθήκη scikit-learn μπορείτε να αναπτύξετε και να εξασκήσετε μηχανική μάθηση στο Python. Το επίκεντρο της βιβλιοθήκης είναι οι αλγόριθμοι μηχανικής μάθησης για ταξινόμηση, παλινδρόμηση, ομαδοποίηση και άλλα. Παρέχει επίσης εργαλεία για συναφείς εργασίες, όπως αξιολόγηση μοντέλων, συντονισμό παραμέτρων και προ επεξεργασία δεδομένων. Τα βασικά χαρακτηριστικά που σχετίζονται με τις προβλέψεις Χρονοσειρών στη scikit-learning περιλαμβάνουν:

• Τη σειρά εργαλείων προετοιμασίας δεδομένων, όπως κλιμάκωση και καταλογισμός δεδομένων.

• Η σειρά αλγορίθμων μηχανικής μάθησης που θα μπορούσαν να χρησιμοποιηθούν για τη μοντελοποίηση δεδομένων και τη δημιουργία προβλέψεων.

• Οι μέθοδοι δειγματοληψίας για την εκτίμηση της απόδοσης ενός μοντέλου σε αόρατα δεδομένα, συγκεκριμένα την κλάση Time SeriesSplit

6.4 IBM SPSS Forecasting

Τα αρχικά του σημαίνουν Superior Performance Software System και για πρώτη φορά αναπτύχθηκε το 1965 στο Stanford της Καλιφόρνια. Πρόκειται για ένα στατιστικό πακέτο ανάλυσης δεδομένων που δίνει στο χρήστη τη δυνατότητα δημιουργίας αναφορών, μοντελοποίησης και ανάλυσης δεδομένων αλλά και τη γραφική αναπαράσταση αυτών. Το πακέτο αυτό προσφέρει ένα μεγάλο αριθμό στατιστικών συναρτήσεων μέσα από ένα φιλικό για το χρήστη, γραφικό περιβάλλον.

Ένα κρίσιμο χαρακτηριστικό της ενότητας IBM SPSS Forecasting είναι το Expert Modeller. Αντί για ορίζουμε τις παραμέτρους και τις ρυθμίσεις των μοντέλων χρονοσειρών με μη αυτόματο τρόπο, το Expert Modeller εντοπίζει αυτόματα και εκτιμά την καλύτερη εφαρμογή ARIMA ή το μοντέλο εκθετικης εξομάλυνσης για μία ή περισσότερες σειρές εξαρτημένων μεταβλητών. Αν και οι χρήστες μπορούν να καθορίσουν ένα προσαρμοσμένο μοντέλο ARIMA ή το μοντέλο εκθετικής εξομάλυνσης χειροκίνητα, το Expert Modeller εξαλείφει πολλά τη δοκιμή και το σφάλμα που σχετίζονται με αυτό

- Το υπολογιστικό σύστημα στο οποίο θα γίνει η ανάπτυξη των μοντέλων προβλεψέων, εχει επεξεργαστή Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz, μνήμη RAM 8,00 GB, δίσκο Nvme 512GB με ταχύτητα εγγραφής 520 MB/s και ανάγνωσης 550 MB/s και ενσωματωμένη κάρτα Intel Graphics

Κεφάλαιο 7. ΑΡΙΘΜΗΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Σε αυτό το κεφάλαιο θα παρουσιαστεί ο τρόπος με τον οποίο κατασκευάστηκαν το μοντέλα πρόβλεψης. Τα μοντέλα αυτά είναι σε γλώσσα προγραμματισμού Python όπως είπαμε. Αρχικά θα πραγματοποιήσουμε προβλέψεις με τις μεθόδους AR, MA, ARMA, ARIMA ,SARIMA Και με τη μέθοδο Prophet, η οποία μέθοδος παρουσιάστηκε πρόσφατα από τη Facebook. Στο τέλος θα παρουσιαστούν τα πλεονεκτήματα και τα μειονεκτήματα της Seasonal ARIMA και της Prophet.

7.1 Εισαγωγή Δεδομένων

Το πρώτο μας βήμα είναι να τσεκάρουμε της εκδόσεις των απαραίτητων βιβλιοθηκών μας

Επόμενο βήμα είναι να εισάγουμε τα πακέτα μας από της βιβλιοθήκες

```
In [1]: import pandas as pd
   ...: import numpy as np
   ...: import matplotlib.pyplot as plt
   ...: from datetime import datetime
   ...: from pandas import Series
   ...: %matplotlib inline
   ...: import warnings
   ...: from statsmodels.tsa.stattools import adfuller
   ...: from sklearn.metrics import mean_squared_error
   ...: from math import sqrt
   ...: from statsmodels.tsa.api import ExponentialSmoothing, SimpleExpSmoothing, Holt
   ...: from statsmodels.tsa.seasonal import seasonal_decompose
   ...: from matplotlib.pylab import rcParams
   ...: from statsmodels.tsa.stattools import acf, pacf
   ...: from statsmodels.tsa.arima_model import ARIMA
   ...: import statsmodels.api as sm
```

Εικόνα 5 Εισαγωγή Βιβλιοθηκών

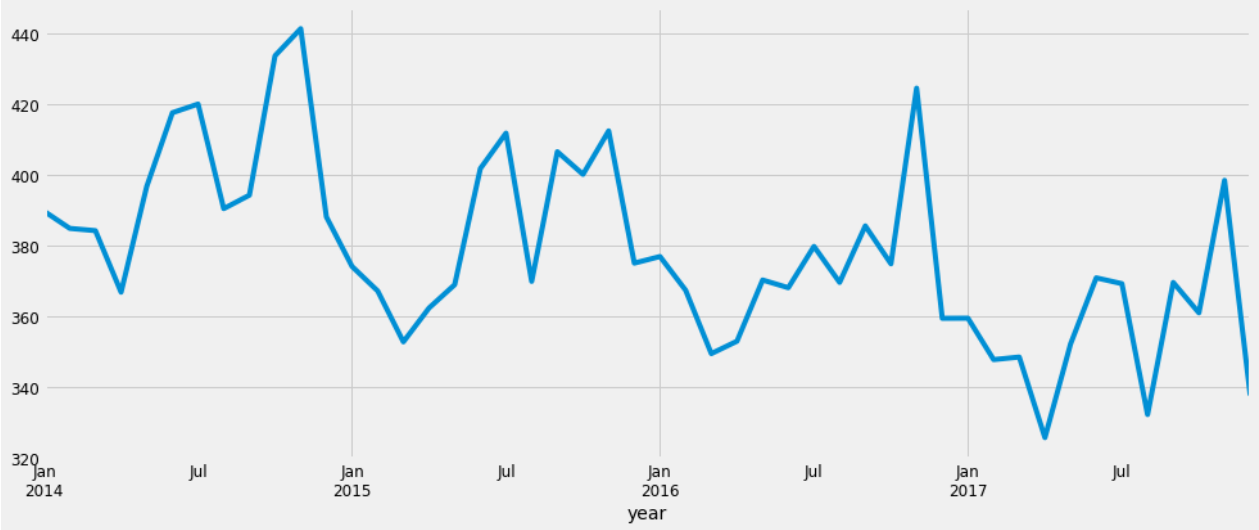
Πίνακας 1 Δεδομένα

Date	Cases

2014-01-01	389,870968
2014-02-01	385,000000
2014-03-01	384,354839
2014-04-01	366,933333
2014-05-01	396,870968
2014-06-01	417,733333
2014-07-01	420,193548
2014-08-01	390,580645
2014-09-01	394,400000
2014-10-01	433,870968
2014-11-01	441,533333
2014-12-01	388,258065
2015-01-01	374,225806
2015-02-01	367,285714
2015-03-01	352,870968
2015-04-01	362,466667

2015-05-01	369,064516
2015-06-01	401,933333
2015-07-01	411,967742
2015-08-01	370,000000
2015-09-01	406,700000
2015-10-01	400,290323
2015-11-01	412,600000
2015-12-01	375,161290
2016-01-01	377,032258
2016-02-01	367,482759
2016-03-01	349,548387
2016-04-01	353,066667
2016-05-01	370,387097
2016-06-01	368,200000
2016-07-01	379,903226
2016-08-01	369,709677

2016-09-01	385,700000
2016-10-01	374,967742
2016-11-01	424,700000
2016-12-01	359,548387
2017-01-01	359,580645
2017-02-01	347,857143
2017-03-01	348,580645
2017-04-01	325,766667
2017-05-01	352,322581
2017-06-01	371,000000
2017-07-01	369,354839
2017-08-01	332,290323
2017-09-01	369,700000
2017-10-01	361,096774
2017-11-01	398,600000
2017-12-01	337,741935



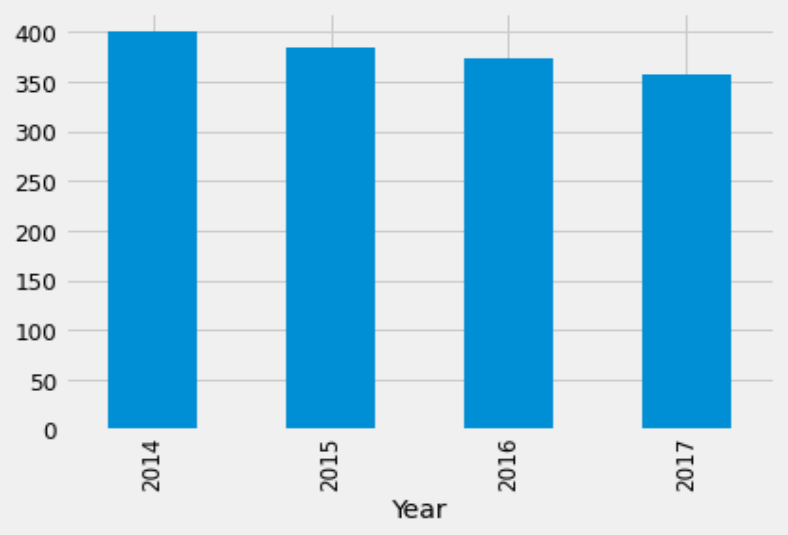
Εικόνα 6

Πίνακας – Χρονοσειρά των δεδομένων μας

Εδώ μπορούμε να συμπεράνουμε ότι υπάρχει μια πτωτική τάση στη σειρά, δηλαδή, ο αριθμός των μετρήσεων μειώνεται σε σχέση με το χρόνο.

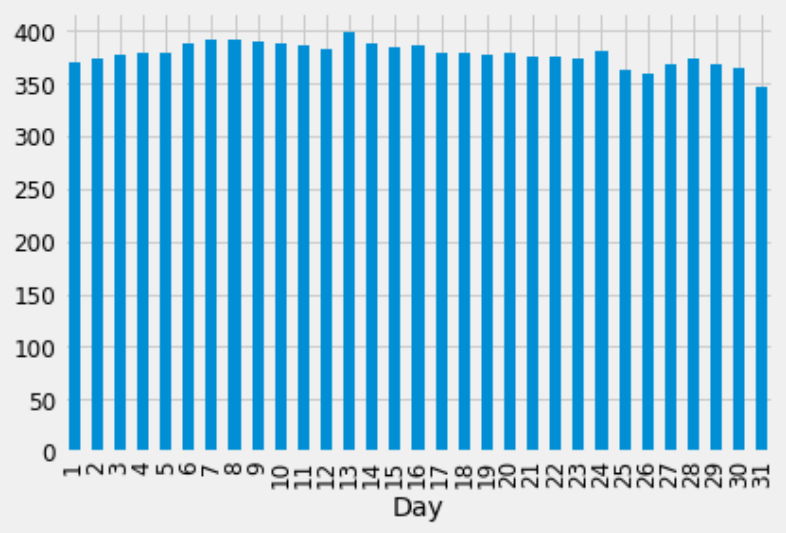
7.1.1 Διερευνητική ανάλυση

Στο βήμα αυτό θα ομαδοποιήσουμε τα δεδομένα μας να δούμε πως σχετίζονται είτε με τον χρόνο είτε μεταξύ τους. Στην αρχη θα ομαδοποιήσουμε ανα έτος.

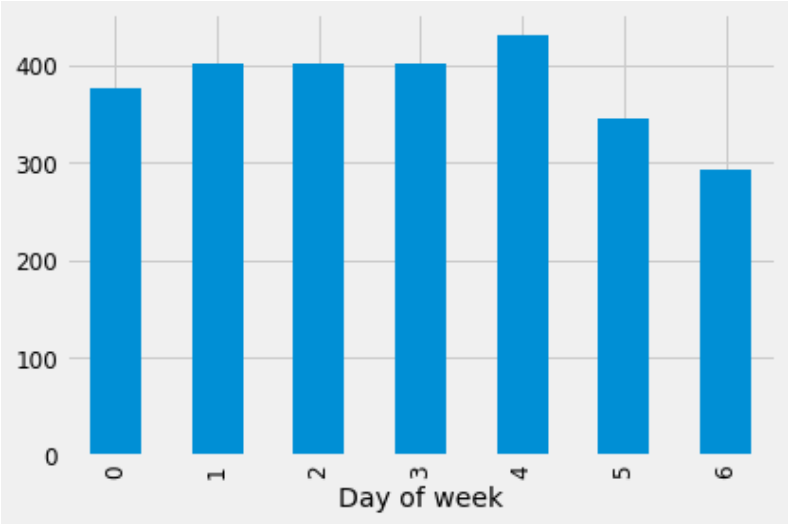


Εικόνα 7 Ετήσια Δεδομένα

Σε αυτό το διάγραμμα βλέπουμε μια εκθετική μείωση των δεδομένων μας .



Εικόνα 8 Ημερήσια Δεδομένα



Εικόνα 9 Δεδομένα ανα ημέρα εβδομάδας

Αφού εχουμε ολοκληρώσει πλεον όλες τις υπόθεσεις μας, ας προχωρήσουμε και φτιάξουμε μοντέλα για τις προβλέψεις. Αλλά πριν το κάνουμε αυτό, θα χρειαστούμε ένα σύνολο δεδομένων (επικύρωση) για να ελέγξουμε την απόδοση και τη δυνατότητα γενίκευσης του μοντέλου μας. Ακολουθούν μερικές από τις ιδιότητες του συνόλου δεδομένων που απαιτούνται για το σκοπό αυτό.

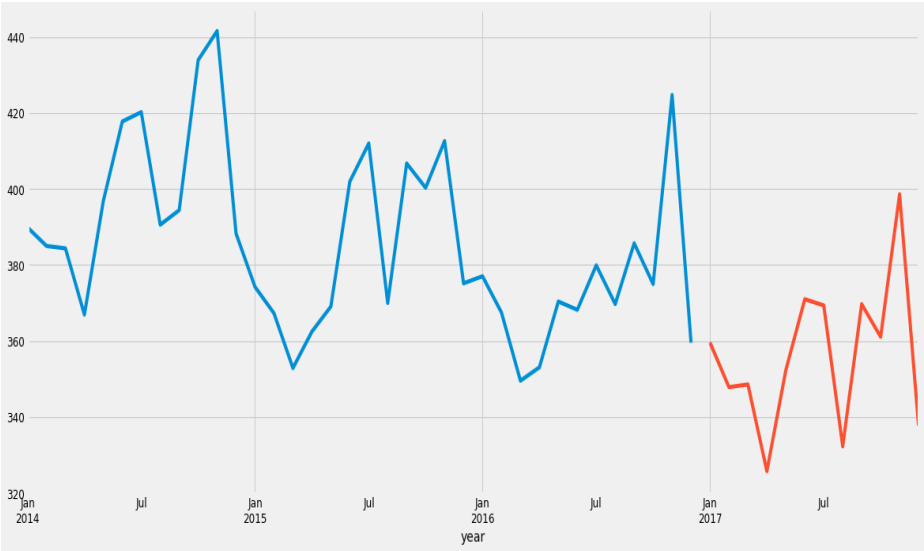
· Το σύνολο δεδομένων πρέπει να έχει τις πραγματικές τιμές της εξαρτημένης μεταβλητής έναντι της οποίας μπορούν να ελεγχθούν οι προβλέψεις. Επομένως, το σύνολο δεδομένων δοκιμής δεν μπορεί να χρησιμοποιηθεί για το σκοπό αυτό.

· Το μοντέλο δεν πρέπει να εκπαιδευτεί στο σύνολο δεδομένων επικύρωσης. Επομένως, δεν μπορούμε να εκπαιδεύσουμε το μοντέλο στο σύνολο δεδομένων αμαξοστοιχίας και να το επικυρώσουμε επίσης.

Έτσι, για τους παραπάνω δύο λόγους, γενικά διαιρούμε το σύνολο δεδομένων αμαξοστοιχίας σε δύο μέρη. Το ένα μέρος χρησιμοποιείται για την εκπαίδευση του μοντέλου και το άλλο χρησιμοποιείται ως σύνολο δεδομένων επικύρωσης. Τώρα υπάρχουν πολλοί τρόποι για να διαιρέσετε το σύνολο δεδομένων του τρένου, όπως το Random Division κ.λπ.

7.1.2 Splitting the data into training and validation part

Η ημερομηνία έναρξης του συνόλου δεδομένων είναι 01-01-2014 όπως έχουμε δει στο μέρος εξερεύνησης και η ημερομηνία λήξης είναι 31-12-2017.Τώρα θα δούμε πώς διαιρέθηκε το τμήμα εκπαίδευσης και επικύρωσης.

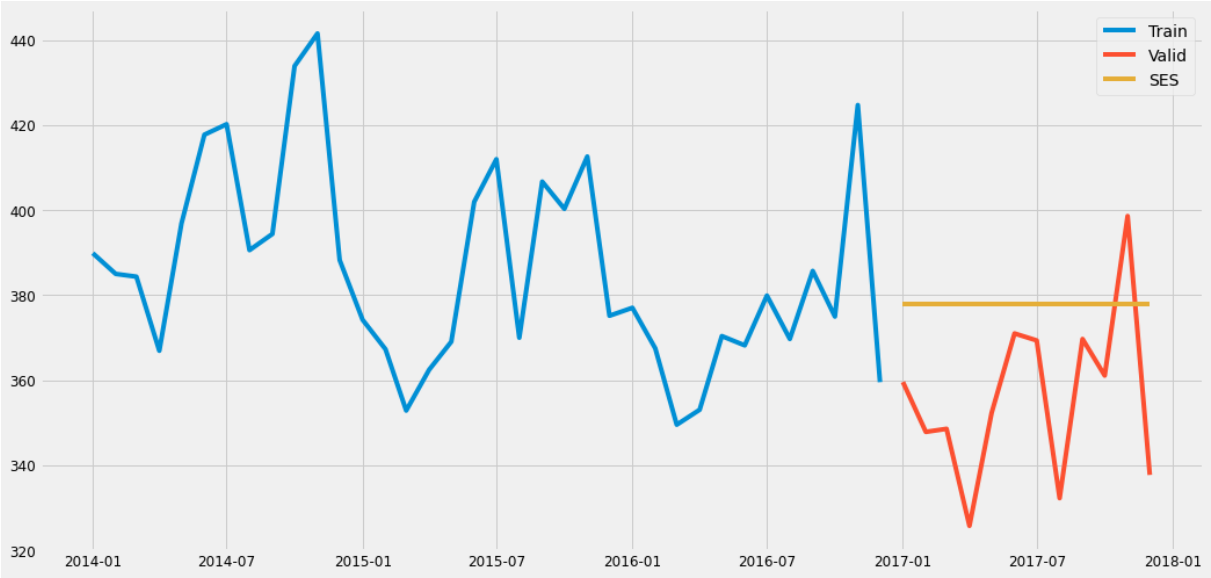


Εικόνα 10

Εδώ το μπλε μέρος αντιπροσωπεύει τα δεδομένα εκπαίδευσης τα οποία είναι από 01/2014 έως και 12/2016 και το πορτοκαλί μέρος αντιπροσωπεύει τα δεδομένα επικύρωσης το οποίο είναι από 01/2017 έως 12/2017. Θα εξετάσουμε διάφορα μοντέλα τώρα για να προβλέψουμε τις χρονοσειρές. Οι μέθοδοι που θα συζητήσουμε για τις προβλέψεις είναι:

7.2 Simple Exponential Smoothing

Σε αυτήν την τεχνική, αποδίδουμε μεγαλύτερα βάρη σε πιο πρόσφατες παρατηρήσεις παρά σε παρατηρήσεις από το μακρινό παρελθόν. Τα βάρη μειώνονται εκθετικά καθώς οι παρατηρήσεις προέρχονται από το παρελθόν, τα μικρότερα βάρη συνδέονται με τις παλαιότερες παρατηρήσεις.



Εικόνα 11 Διάγραμμα Προβλέψεων με την μέθοδο Simple Exponential Smoothing

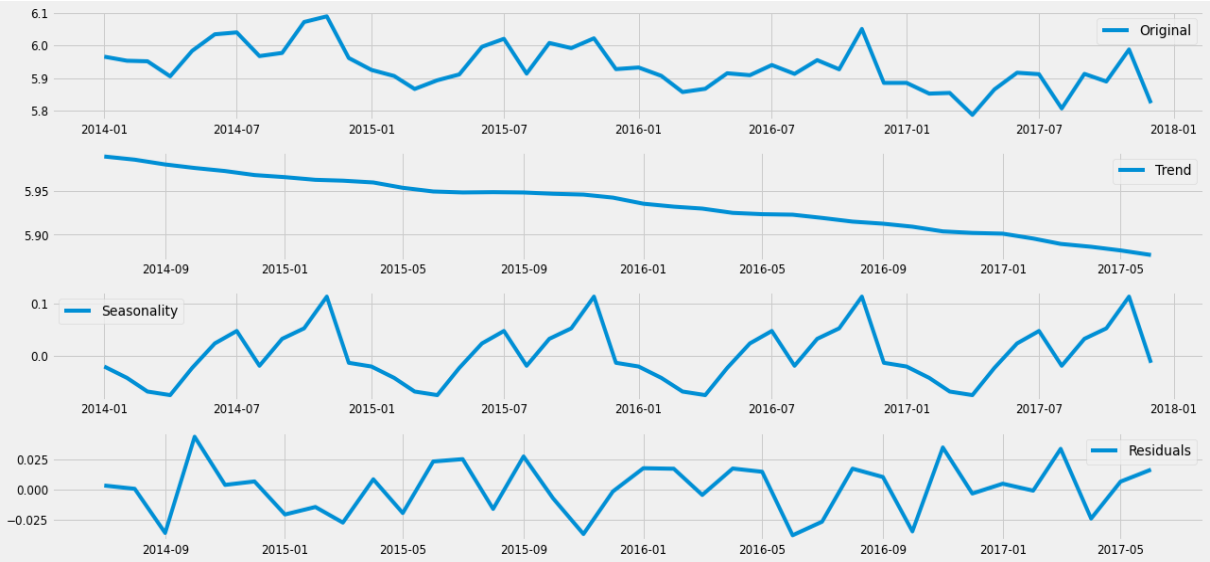
Και μας επιστρέφει σαν αποτέλεσμα:

Exponential Smoothing **MSE = 29.01**

7.3 Holt's Linear Trend Model

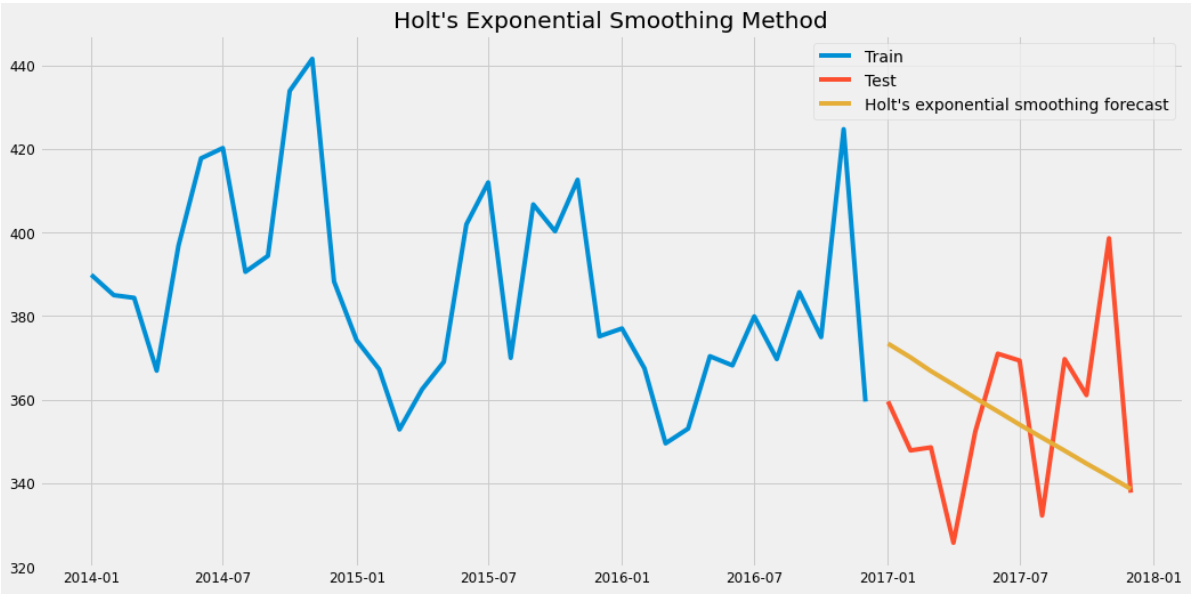
Πρώτα από 'όλα, ας απεικονίσουμε την τάση, την εποχικότητα και το σφάλμα στη σειρά. Μπορούμε να αποσυνθέσουμε τις χρονοσειρές σε τέσσερα μέρη:.

1. Τις παρατηρήσεις, που είναι η αρχική χρονοσειρά.
2. Την Τάση, δείχνει την τάση στις χρονοσειρές, δηλαδή αύξηση ή μείωση της συμπεριφοράς των Χρονοσειρών.
3. Εποχικότητα, η οποία μας λέει για την εποχικότητα στις χρονοσειρές.
4. Residuals, το οποίο επιτυγχάνεται αφαιρώντας οποιαδήποτε τάση ή εποχικότητα στις χρονοσειρές.



Εικόνα 12 Διάγραμμα Ανάλυσης της χρονοσειράς

Μια φθίνουσα τάση φαίνεται στο σύνολο δεδομένων, οπότε τώρα θα φτιάξουμε ένα μοντέλο με βάση την τάση.



Εικόνα 13 - Διάγραμμα Προβλέψεων με την Holt Exponential Method

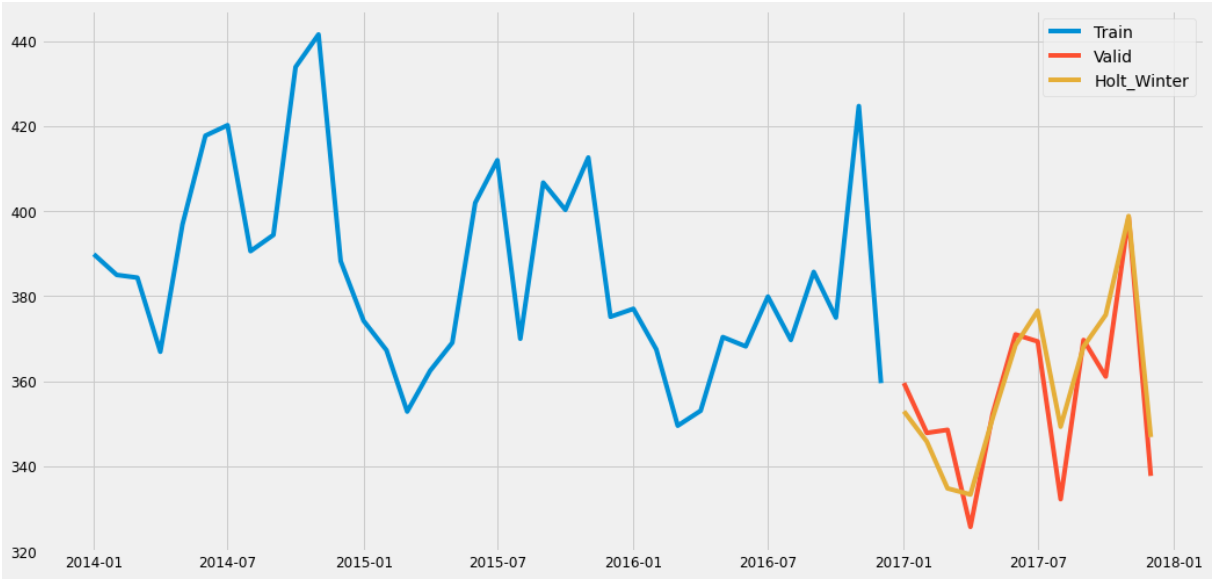
Holt's exponential smoothing **MSE = 24.6247006664489**

Μπορεί να δούμε λαμβάνοντας υπόψιν την τάση και μόνο ότι η τιμή rmse έχει μειωθεί.

Τώρα θα προβλέψουμε τον αριθμό επιβατών για το σύνολο δεδομένων δοκιμής χρησιμοποιώντας διάφορα μοντέλα

7.4 Holt winter's model

Τα σύνολα δεδομένων που εμφανίζουν παρόμοιο σύνολο προτύπων μετά από σταθερά διαστήματα μιας χρονικής περιόδου υποφέρουν από εποχικότητα. Τα προαναφερθέντα μοντέλα δεν λαμβάνουν υπόψη την εποχικότητα του συνόλου δεδομένων κατά την πρόβλεψη. Ως εκ τούτου, χρειαζόμαστε μια μέθοδο που να λαμβάνει υπόψη τόσο την τάση όσο και την εποχικότητα για να προβλέψουμε τις μελλοντικές τιμές. Ένας τέτοιος αλγόριθμος που μπορούμε να χρησιμοποιήσουμε σε ένα τέτοιο σενάριο είναι η μέθοδος Holt's Winter. Η ιδέα πίσω από το Holt's Winter είναι η εφαρμογή εκθετικής εξομάλυνσης στα εποχιακά στοιχεία εκτός από το επίπεδο και την τάση.



Εικόνα 14- Διάγραμμα Προβλέψεων με την Holt winter's model

Holt_Winter MSE =8.856644277599546

Μπορούμε να δούμε ότι η τιμή rmse έχει μειωθεί πολύ από αυτήν τη μέθοδο.

7.5 ARIMA model

Το ARIMA σημαίνει Auto Regression Integrated Moving Average. Καθορίζεται από τρεις παραμέτρους που ταξινομούνται (p, d, q).

- P είναι η σειρά του αυτεπαρκτικού μοντέλου (αριθμός χρονικών καθυστερήσεων)
- d είναι ο βαθμός διαφοράς (πόσες φορές τα δεδομένα είχαν αφαιρεθεί προηγούμενες τιμές)
- q είναι η σειρά του κινούμενου μέσου μοντέλου.

Η πρόβλεψη ARIMA για μια στάσιμη χρονική σειρά δεν είναι παρά μια γραμμική εξίσωση (όπως μια γραμμική παλινδρόμηση).

Πρώτα από 'όλα πρέπει να βεβαιωθούμε ότι οι χρονοσειρές είναι στάσιμη. Εάν η σειρά δεν είναι στατική, θα την κάνουμε στάσιμη. Χρησιμοποιούμε το τεστ Dickey Fuller για να ελέγξουμε τη σταθερότητα της σειράς.

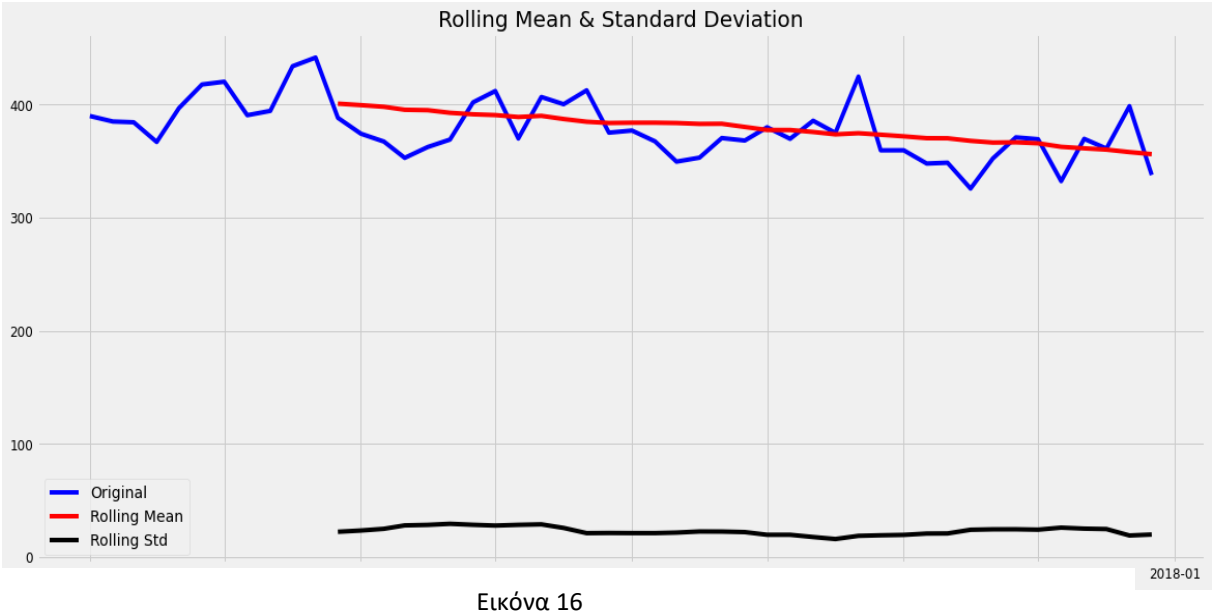
- Η διαίσθηση πίσω από αυτό το τεστ είναι ότι καθορίζει πόσο έντονα καθορίζεται μια χρονική σειρά από μια τάση.
- Η αρχική υπόθεση του τεστ είναι ότι οι χρονοσειρές δεν είναι στάσιμες (έχει κάποια χρονοεξαρτώμενη δομή).
- Η εναλλακτική υπόθεση (απόρριψη της αρχικής υπόθεσης) είναι ότι οι χρονοσειρές είναι σταθερές.

Τα αποτελέσματα των δοκιμών περιλαμβάνουν μια στατιστική δοκιμή και μερικές κρίσιμες τιμές για τα επίπεδα εμπιστοσύνης διαφοράς. Αν το "Test Statistic" είναι μικρότερο από το "Critical Value", μπορούμε να απορρίψουμε την μηδενική υπόθεση και να πούμε ότι η σειρά είναι στάσιμη.

```
In [46]: def Pred_stationarity(timeseries):
    ...:
    ...:     #Determing rolling statistics
    ...:     rolmean = timeseries.rolling(window=12).mean()
    ...:     rolstd = timeseries.rolling(window=12).std()
    ...:
    ...:     #Plot rolling statistics:
    ...:     orig = plt.plot(timeseries, color='blue',label='Original')
    ...:     mean = plt.plot(rolmean, color='red', label='Rolling Mean')
    ...:     std = plt.plot(rolstd, color='black', label = 'Rolling Std')
    ...:     plt.legend(loc='best')
    ...:     plt.title('Rolling Mean & Standard Deviation')
    ...:     plt.show(block=False)
    ...:
    ...:     #Perform Dickey-Fuller Pred:
    ...:     print ('Results of Dickey-Fuller Test:')
    ...:     dfPred = adfuller(timeseries, autolag='AIC')
    ...:     dfoutput = pd.Series(dfPred[0:4], index=['Test Statistic','p-value','#Lags Used','Number of Observations Used'])
    ...:     for key,value in dfPred[4].items():
    ...:         dfoutput['Critical Value (%s)'%key] = value
    ...:     print (dfoutput)
```

Εικόνα 15 Κώδικας για Dickey Fuller Test

Ας κάνουμε την συνάρτηση με την οποία μπορούμε να χρησιμοποιήσουμε για τον υπολογισμό των αποτελεσμάτων του τεστ Dickey-Fuller..



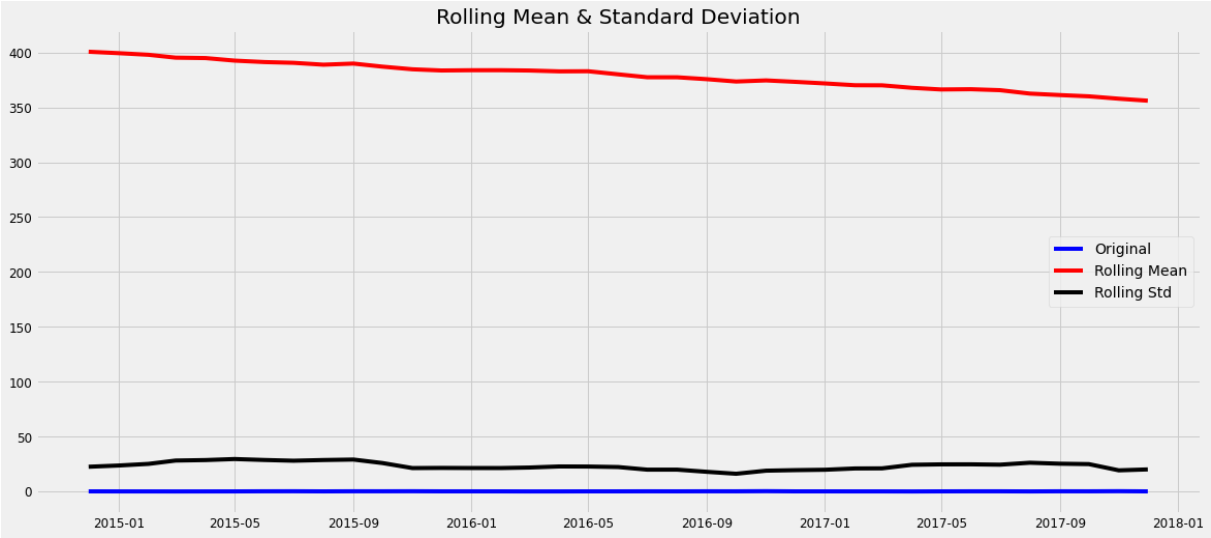
Εικόνα 16

Results of Dickey-Fuller Test:	
Test Statistic	1.244085
p-value	0.654350
#Lags Used	10.000000
Number of Observations	37.000000
Critical Value (1%)	-3.620918
Critical Value (5%)	-2.943539
Critical Value (10%)	-2.610400

Τα στατιστικά στοιχεία δείχνουν ότι οι χρονοσειρές είναι σταθερές ως προς τη Στατιστική δοκιμής < Κρίσιμη τιμή, αλλά μπορούμε να δούμε μια φθίνουσα τάση στα δεδομένα. Έτσι, πρώτα θα προσπαθήσουμε να κάνουμε τα δεδομένα πιο σταθερά. Για να γίνει αυτό, πρέπει να καταργήσουμε την τάση και την εποχικότητα από τα δεδομένα.

7.6 Αφαίρεση Τάσης

Βλέπουμε μια αυξανόμενη τάση στα δεδομένα, ώστε να μπορούμε να εφαρμόσουμε μετασχηματισμό που τιμωρεί υψηλότερες τιμές από τις μικρότερες, για παράδειγμα μετασχηματισμός καταγραφής. Θα πάρουμε κυλιόμενο μέσο όρο εδώ για να καταργήσουμε την τάση. Θα πάρουμε το μέγεθος του παραθύρου των 24 με βάση το γεγονός ότι κάθε μέρα έχει 24 ώρες

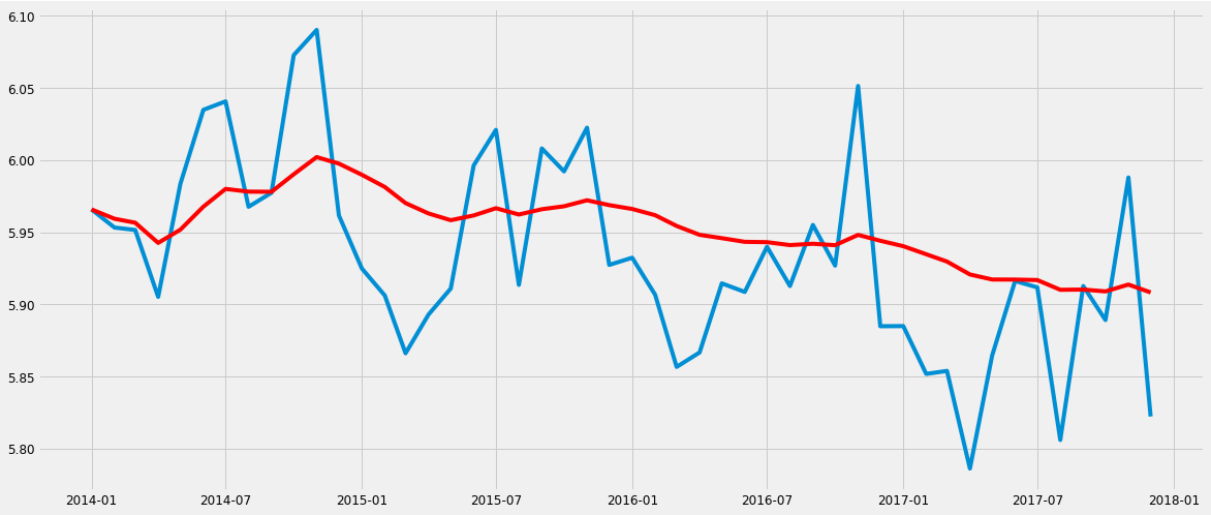


Εικόνα 17

Results of Dickey-Fuller Test:	
Test Statistic	-5.081427
p-value	0.000015
#Lags Used	10.000000
Number of Observations Used	26.000000
Critical Value (1%)	-3.711212
Critical Value (5%)	-2.981247
Critical Value (10%)	-2.630095

Έτσι μπορούμε να παρατηρήσουμε μια φθίνουσα τάση. Τώρα θα καταργήσουμε αυτήν την αυξανόμενη τάση για να κάνουμε την χρονοσειρά μας στάσιμη

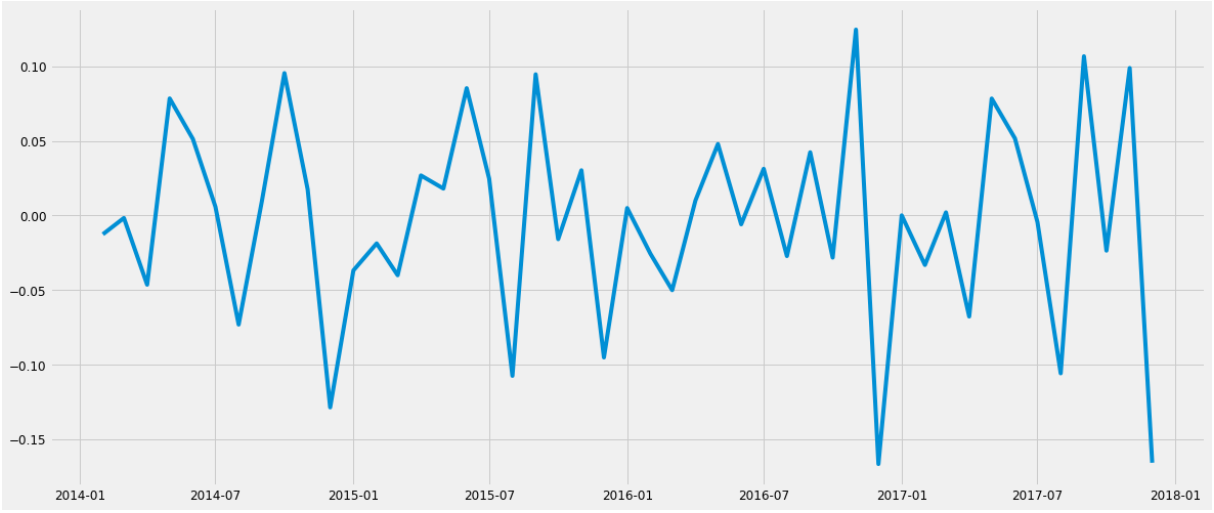
Δεδομένου ότι πήραμε τον μέσο όρο των 24 τιμών, ο κυλιόμενος μέσος όρος δεν ορίζεται για τις πρώτες 23 τιμές. Ας ρίξουμε λοιπόν αυτές τις μηδενικές τιμές.



Εικόνα 18

Results of Dickey-Fuller Test:	
Test Statistic	-3.065551
p-value	0.029210
#Lags Used	8.000000
Number of Observations Used	39.000000
Critical Value (1%)	-3.610400
Critical Value (5%)	-2.939109
Critical Value (10%)	-2.608063

Μπορούμε να δούμε ότι η στατιστική δοκιμή είναι πολύ μικρότερη σε σύγκριση με την κρίσιμη τιμή. Έτσι, μπορούμε να είμαστε σίγουροι ότι η τάση έχει σχεδόν αφαιρεθεί. Ας σταθεροποιήσουμε τώρα το μέσο όρο των Χρονοσειρών που είναι επίσης απαίτηση για σταθερές χρονοσειρές. Η διαφοροποίηση μπορεί να βοηθήσει να κάνει τη σειρά σταθερή και να εξαλείψει την τάση.



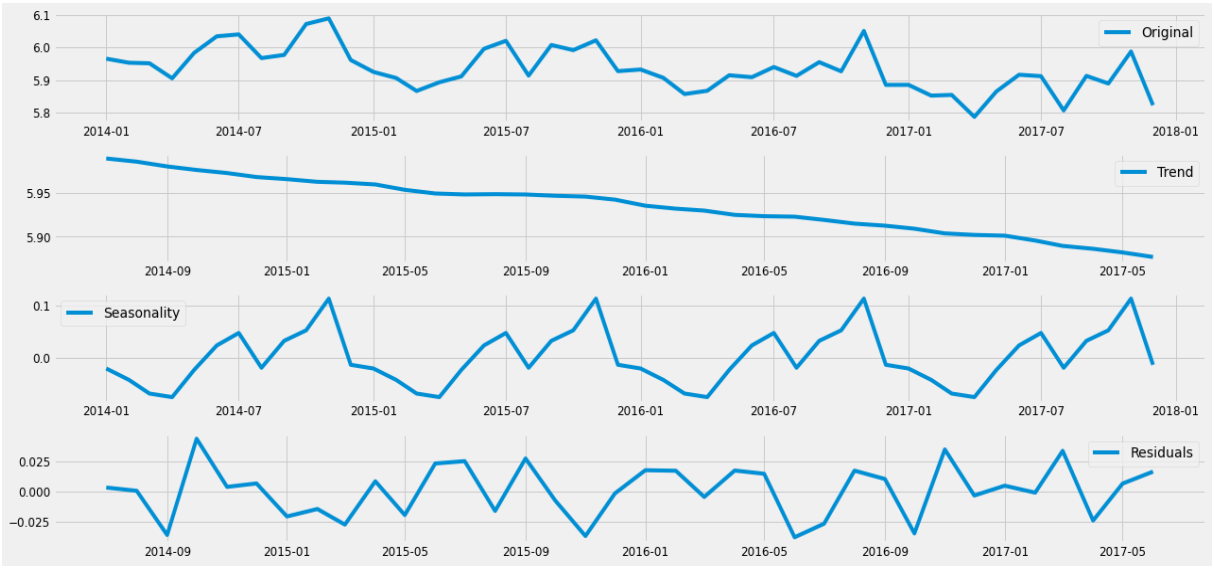
Εικόνα 19

Test Statistic	-8.690732e+00
p-value	4.047383e-14
#Lags Used	1.000000e+01
Number of Observations Used	3.600000e+01
Critical Value (1%)	-3.626652e+00
Critical Value (5%)	-2.945951e+00
Critical Value (10%)	-2.611671e+00

Τώρα θα αποσυνθέσουμε τις χρονοσειρές σε τάση και εποχικότητα και θα πάρουμε το υπόλοιπο που είναι η τυχαία μεταβολή στη σειρά.

7.7 Αφαιρώντας την εποχικότητα

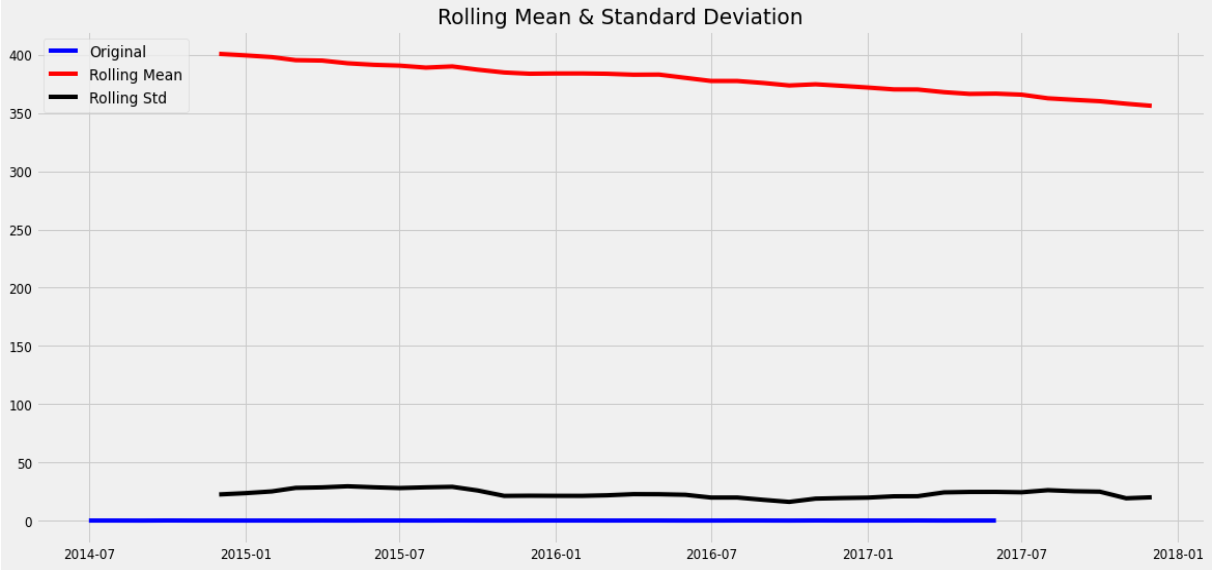
Θα χρησιμοποιήσουμε εποχιακή αποσύνθεση για να αποσυνθέσουμε τις χρονοσειρές σε τάση, εποχικότητα και υπολείμματα.



Εικόνα 20 Ανάλυση της χρονοσειράς

Μπορούμε να δούμε καθαρά την τάση, τα υπολείμματα και την εποχικότητα στο παραπάνω γράφημα. Η εποχικότητα δείχνει μια σταθερή τάση στο μετρητή.

Ας ελέγξουμε τη σταθερότητα των υπολειμμάτων.



Εικόνα 21

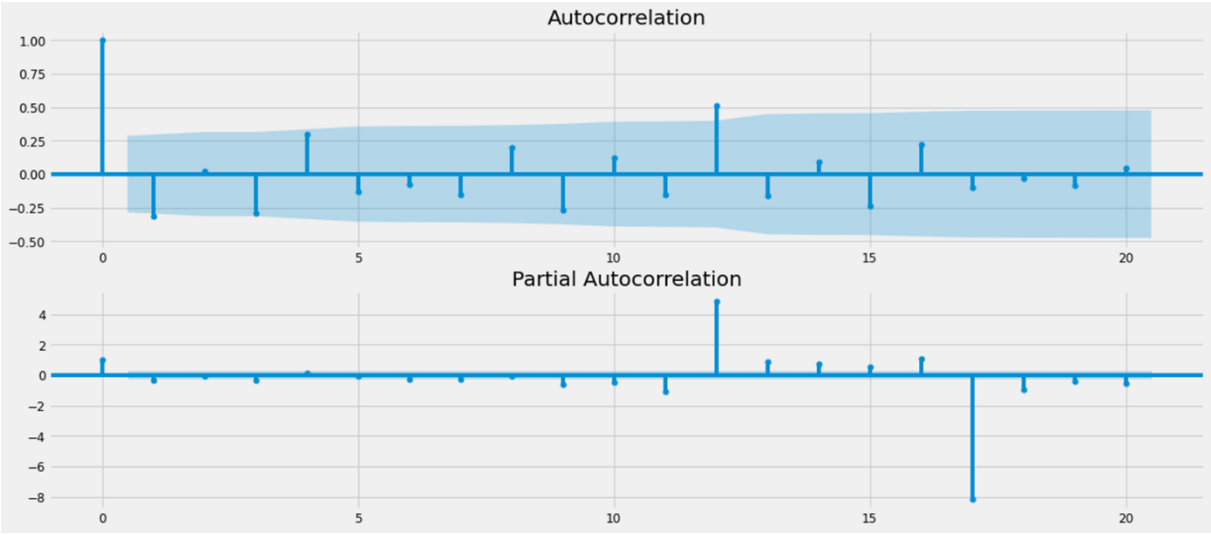
Μπορεί να ερμηνευθεί από τα αποτελέσματα ότι τα υπολείμματα είναι στάσιμα. Τώρα θα προβλέψουμε τις χρονοσειρές χρησιμοποιώντας διαφορετικά μοντέλα.

7.8 Προβλέψεις χρονοσειρών με ARIMA

Πρώτα από 'όλα θα ταιριάξουμε το μοντέλο ARIMA στις χρονολογικές μας σειρές για αυτό πρέπει να βρούμε τις βελτιστοποιημένες τιμές για τις παραμέτρους p, d, q.
Για να βρούμε τις βελτιστοποιημένες τιμές αυτών των παραμέτρων, θα χρησιμοποιήσουμε το γράφημα ACF (Funocorelation Function) και PACF (Partial Autocorrelation Function).

Το ACF είναι ένα μέτρο της συσχέτισης μεταξύ των χρονοσειρών με μια καθυστερημένη έκδοση του ίδιου.

Το PACF μετρά τη συσχέτιση μεταξύ των χρονοσειρών με μια καθυστερημένη έκδοση του, αλλά αφού εξαλείψει τις παραλλαγές που εξηγούνται ήδη από τις παρεμβατικές συγκρίσεις.



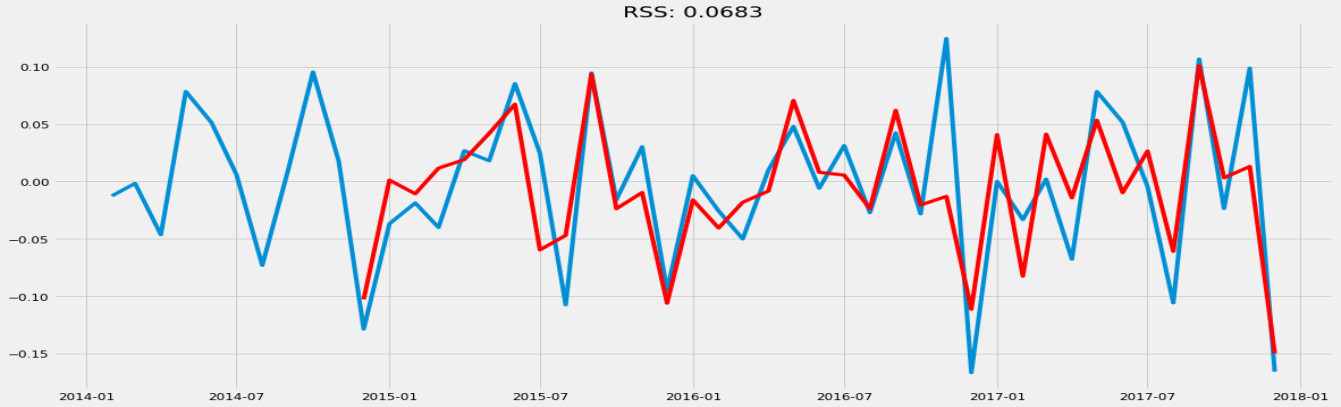
Εικόνα 22 Autocorrelation Plot & Partial

Οπώς βλέπουμε και από τα διαγράμματα η τιμή p είναι η τιμή υστέρησης όπου το διάγραμμα PACF διασχίζει το πρώτο διάστημα εμπιστοσύνης για πρώτη φορά. να παρατηρηθεί ότι σε αυτήν την περίπτωση p = 1. Η τιμή q είναι η τιμή υστέρησης όπου το

διάγραμμα ACF διασχίζει το ανώτερο διάστημα εμπιστοσύνης για πρώτη φορά. Μπορεί να παρατηρηθεί ότι σε αυτήν την περίπτωση q = 1. Τώρα θα φτιάξουμε το μοντέλο ARIMA καθώς έχουμε τις τιμές p, q. Θα φτιάξουμε τα μοντέλα AR και MA ξεχωριστά και στη συνέχεια θα τα συνδυάσουμε κάνοντας ένα μοντέλο ARMA

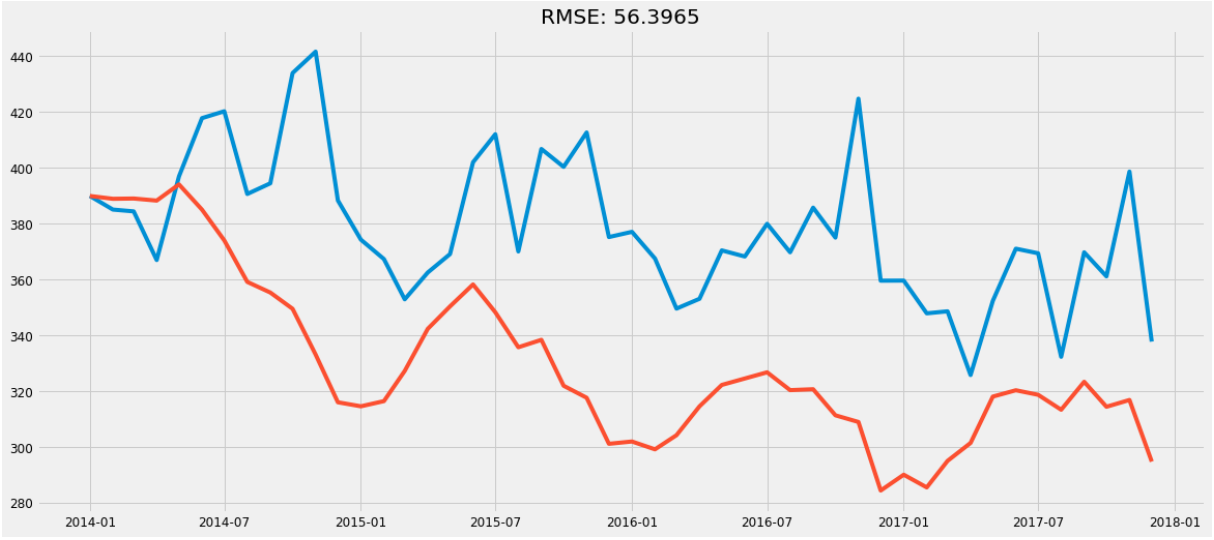
7.8.1 AR model

Το μοντέλο αυτόματης αύξησης καθορίζει ότι η μεταβλητή εξόδου εξαρτάται γραμμικά από τις προηγούμενες τιμές της.



Εικόνα 23

Ας σχεδιάσουμε την καμπύλη επικύρωσης για το μοντέλο AR. Πρέπει να αλλάξουμε την κλίμακα του μοντέλου στην αρχική κλίμακα. Το πρώτο βήμα θα ήταν να αποθηκεύσετε τα προβλεπόμενα αποτελέσματα ως ξεχωριστή σειρά και να τα παρατηρήσετε.



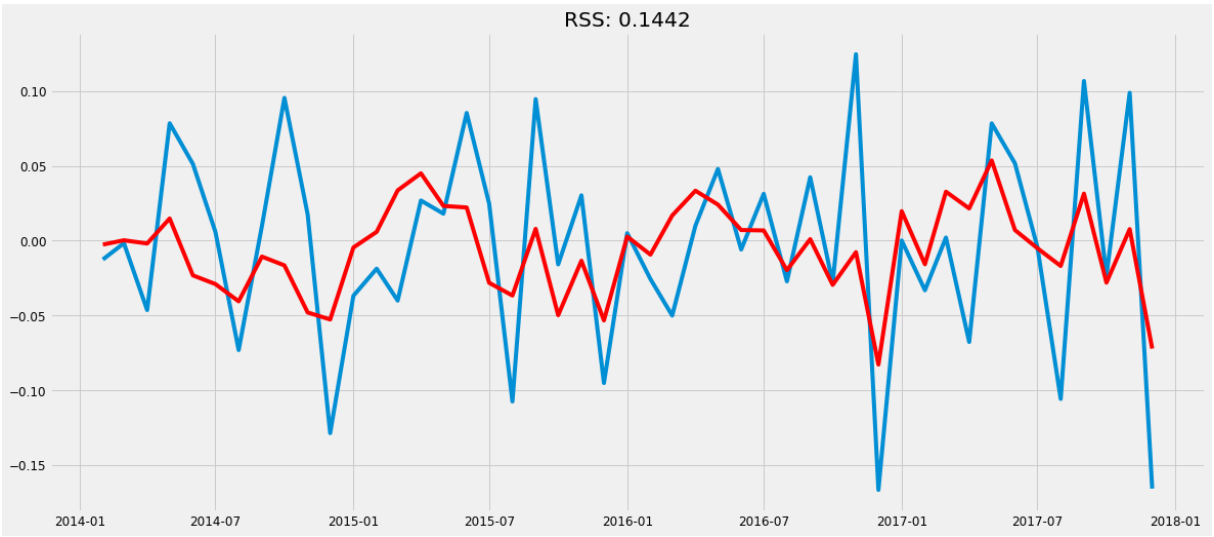
Εικόνα 24 Προβλέψεις τιμως για την μέθοδο AR

	r2_score	mean_absolute_error	median_absolute_error	mse	msle	mape	rmse
0	-3.828.671	50.054.458	48.753.264	3.180.570.142	0.025588	13.097.184	56.396.544

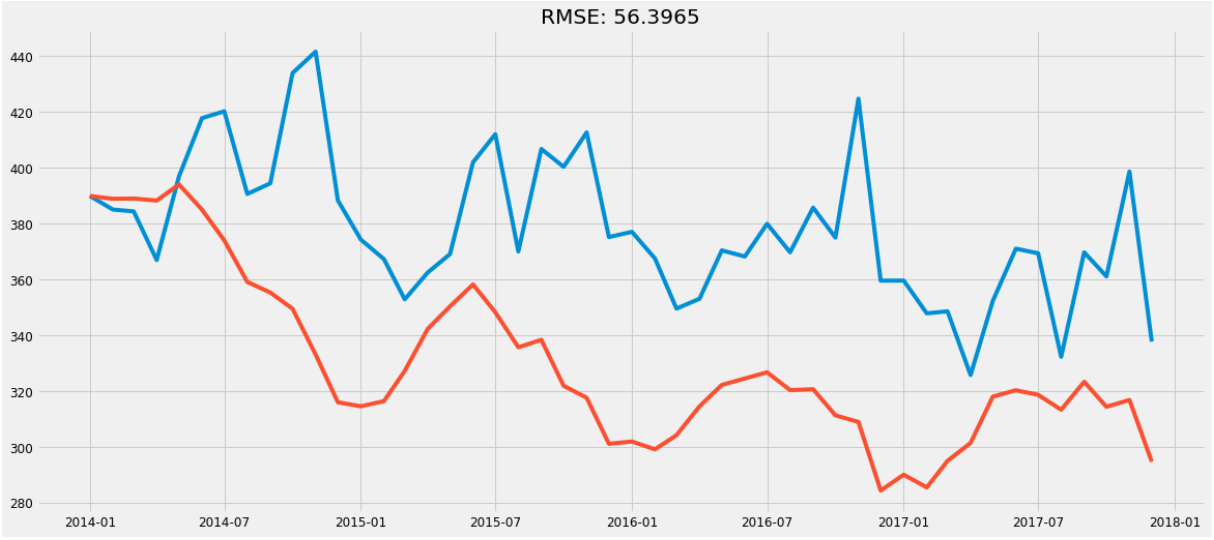
Εδώ η κόκκινη γραμμή δείχνει την πρόβλεψη για το σύνολο επικύρωσης.

7.9 MA model

Το μοντέλο κινούμενου μέσου όρου καθορίζει ότι η μεταβλητή εξόδου εξαρτάται γραμμικά από τις τρέχουσες και διάφορες προηγούμενες τιμές ενός στοχαστικού (ατελή προβλέψιμου) όρου.



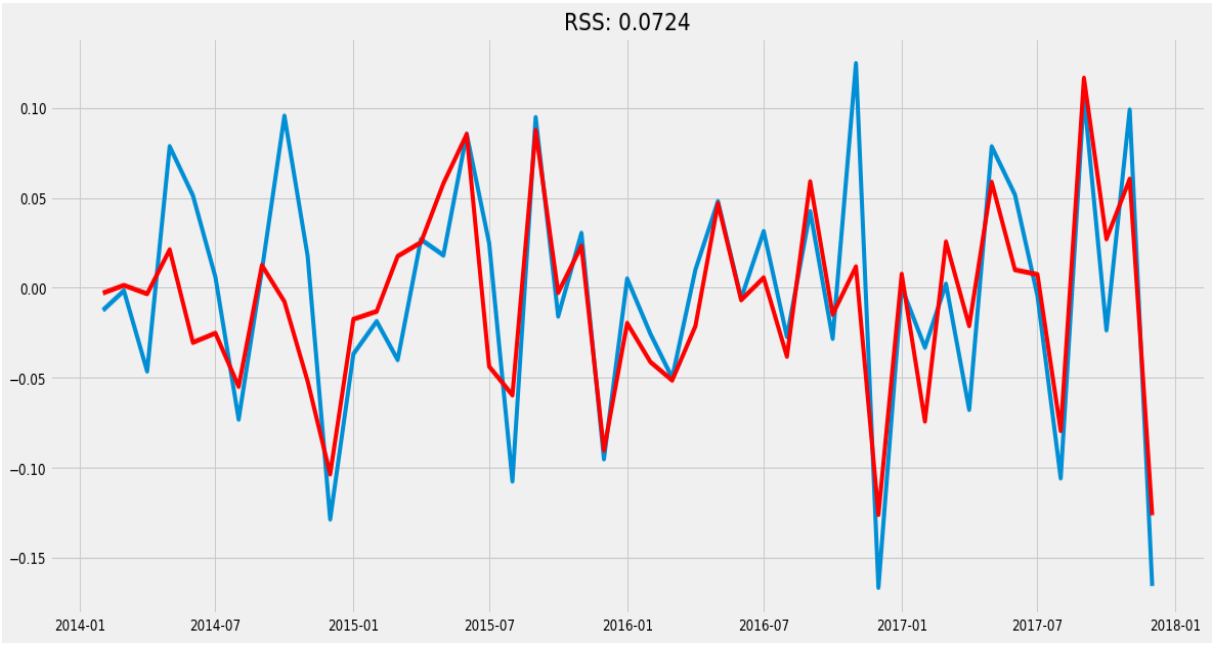
Εικόνα 25



Εικόνα 26 Προβλέψεις τιμων για την μέθοδα MA

	r2_score	mean_absolute_error	median_absolute_error	mse	msle	mape	rmse
0	-3.828.671	50.054.458	48.753.264	3.180.570.142	0.025588	13.097.184	56.396.544

7.10 ARMA Model



Εικόνα 27

"""
ARMA Model Results

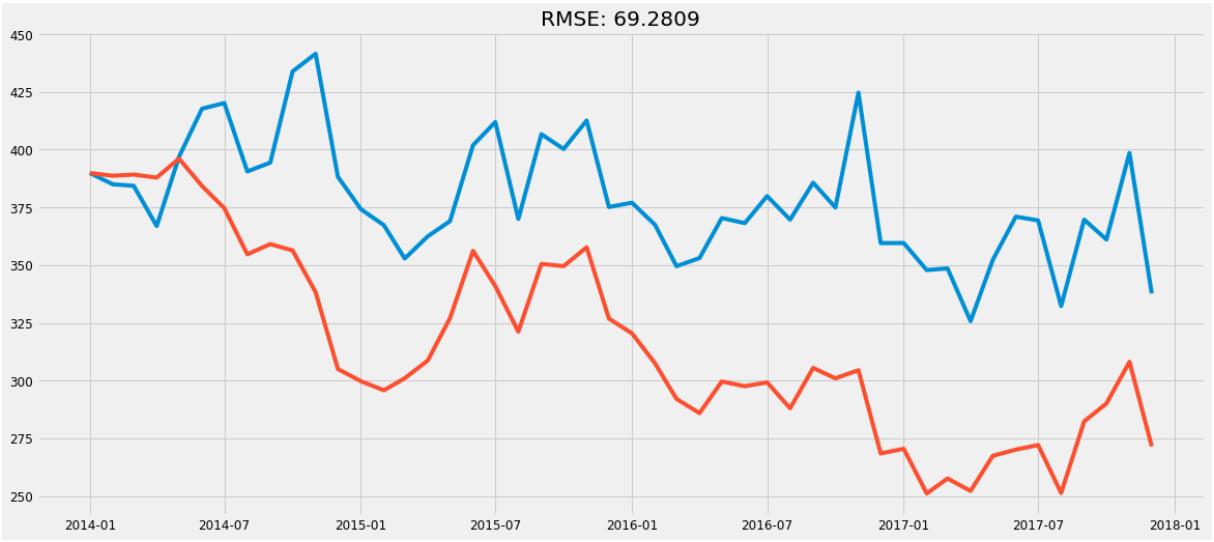
Dep. Variable:	passengers	No. Observations:	47
Model:	ARMA(9, 1)	Log Likelihood	86.351
Method:	css-mle	S.D. of innovations	0.034
Date:	Wed, 24 Feb 2021	AIC	-148.701
Time:	22:20:27	BIC	-126.499
Sample:	02-01-2014 - 12-01-2017	HQIC	-140.347

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.0030	0.000	-21.912	0.000	-0.003	-0.003
ar.L1.passengers	-0.0339	0.116	-0.293	0.770	-0.261	0.193
ar.L2.passengers	-0.1617	0.112	-1.438	0.150	-0.382	0.059
ar.L3.passengers	-0.5219	0.112	-4.651	0.000	-0.742	-0.302
ar.L4.passengers	-0.0368	0.120	-0.306	0.760	-0.272	0.199
ar.L5.passengers	-0.2807	0.107	-2.619	0.009	-0.491	-0.071
ar.L6.passengers	-0.4575	0.118	-3.881	0.000	-0.689	-0.226
ar.L7.passengers	-0.1913	0.118	-1.627	0.104	-0.422	0.039
ar.L8.passengers	-0.1691	0.120	-1.414	0.157	-0.404	0.065
ar.L9.passengers	-0.6148	0.118	-5.221	0.000	-0.846	-0.384
ma.L1.passengers	-0.9988	0.077	-12.907	0.000	-1.150	-0.847

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	0.8837	-0.4968j	1.0137	-0.0815
AR.2	0.8837	+0.4968j	1.0137	0.0815
AR.3	0.4936	-0.9476j	1.0684	-0.1736
AR.4	0.4936	+0.9476j	1.0684	0.1736
AR.5	-0.0997	-1.0423j	1.0470	-0.2652
AR.6	-0.0997	+1.0423j	1.0470	0.2652
AR.7	-1.0308	-0.0000j	1.0308	-0.5000
AR.8	-0.8997	-0.6462j	1.1077	-0.4009
AR.9	-0.8997	+0.6462j	1.1077	0.4009
MA.1	1.0012	+0.0000j	1.0012	0.0000

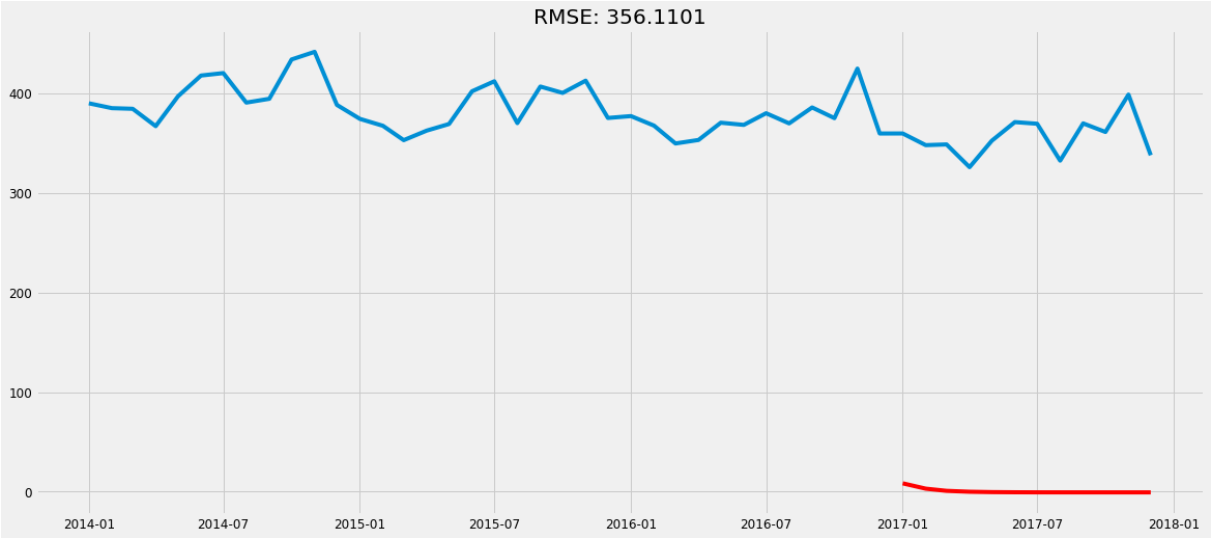
Εικόνα 28 Πίνακας αποτελεσμάτων ARMA



Εικόνα 29 Πρόβλεψη τιμων για τη μέθοδο ARMA

	r2_score	mean_absolute_error	median_absolute_error	mse	msle	mape	rmse
0	-6.287.022	63.604.474	70.696.485	4.799.847.049	0.043293	16.954.699	69.280.928

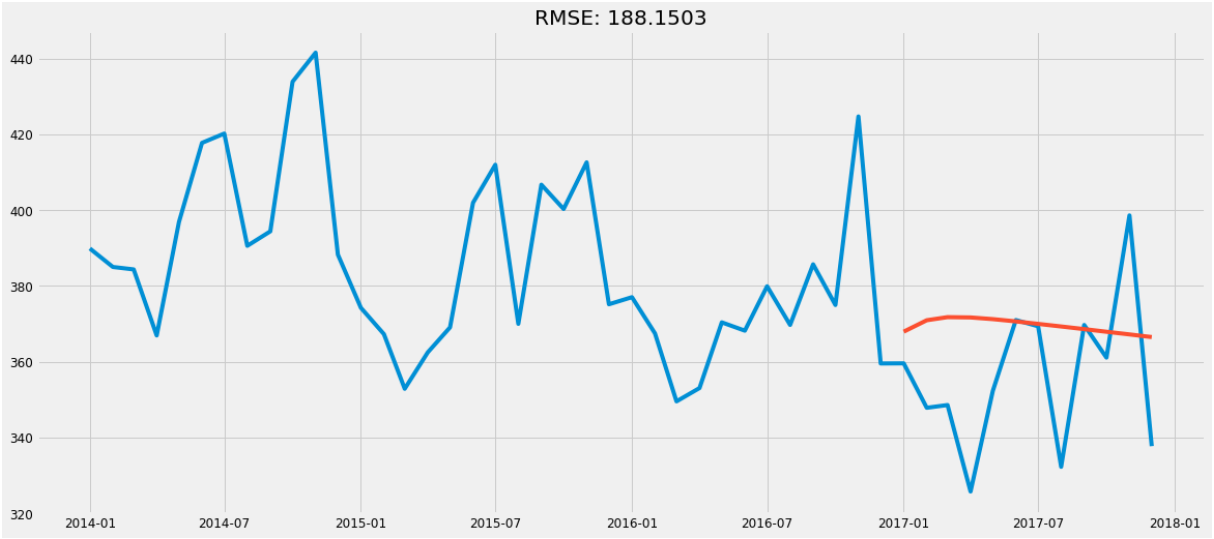
7.11 ARIMA Model



Εικόνα 30

```
"""
                             ARIMA Model Results
==============================================================================
Dep. Variable:           D.passengers   No. Observations:                   35
Model:                ARIMA(1, 1, 1)   Log Likelihood                -157.046
Method:                       css-mle   S.D. of innovations             20.672
Date:                Wed, 24 Feb 2021   AIC                            322.092
Time:                        22:31:10   BIC                            328.313
Sample:                    02-01-2014   HQIC                           324.239
                         - 12-01-2016
==========================================================================================
                             coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------------------
const                     -0.7005      0.531     -1.320      0.187      -1.741       0.340
ar.L1.D.passengers         0.4094      0.160      2.561      0.010       0.096       0.723
ma.L1.D.passengers        -1.0000      0.076    -13.110      0.000      -1.150      -0.850
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            2.4426           +0.0000j            2.4426            0.0000
MA.1            1.0000           +0.0000j            1.0000            0.0000
-----------------------------------------------------------------------------
"""
 This problem is unconstrained.
```

Εικόνα 31 Πίνακας αποξεσμάτων για μοντέλο ARIMAX(1,1,1)



Εικόνα 32. Πρόβλεψη για την μέθοδο ARIMA(1,1,1)

7.12 SARIMAX

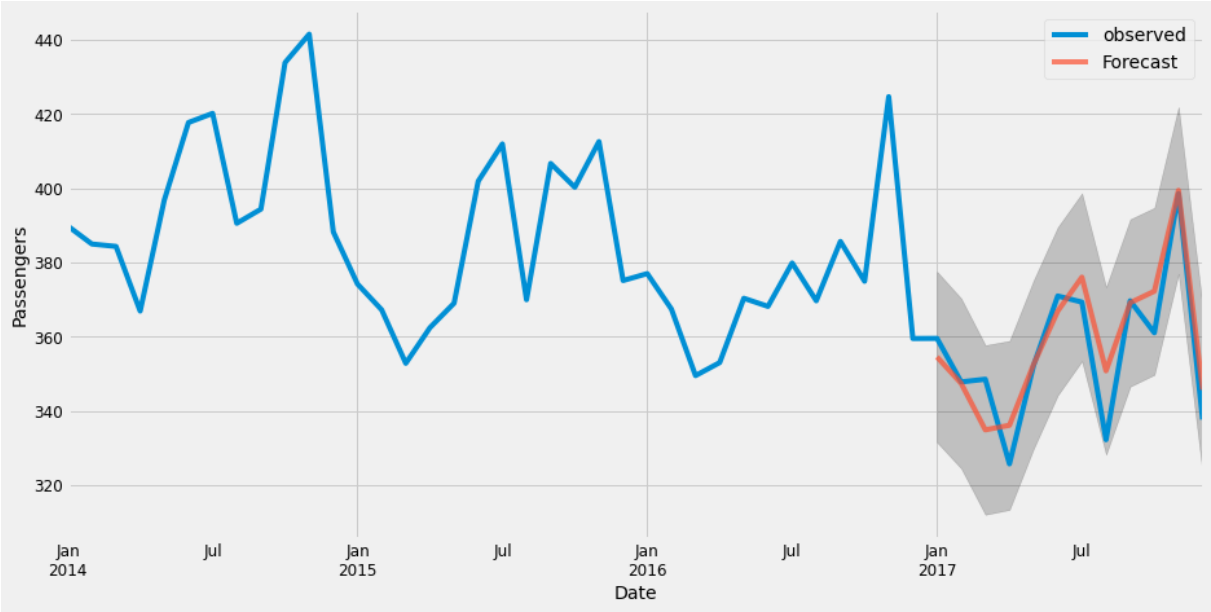
```
Out[5]:
<class 'statsmodels.iolib.summary.Summary'>
"""
                                SARIMAX Results
==========================================================================================
Dep. Variable:                          passengers   No. Observations:                   48
Model:             SARIMAX(0, 1, 1)x(0, 1, 1, 12)   Log Likelihood                -138.413
Date:                            Thu, 25 Feb 2021   AIC                             282.826
Time:                                    22:48:02   BIC                             287.492
Sample:                                01-01-2014   HQIC                            284.436
                                     - 12-01-2017
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1         -1.0000   1960.655     -0.001      1.000   -3843.813    3841.813
ma.S.L12      -0.6153      0.430     -1.433      0.152      -1.457       0.227
sigma2       120.8416   2.37e+05      0.001      1.000   -4.64e+05    4.65e+05
===================================================================================
Ljung-Box (L1) (Q):                   0.87   Jarque-Bera (JB):                 0.47
Prob(Q):                              0.35   Prob(JB):                         0.79
Heteroskedasticity (H):               0.50   Skew:                             0.05
Prob(H) (two-sided):                  0.24   Kurtosis:                         2.44
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
"""
```

Εικόνα 33 Πίνακας αποτεσμάτων για μοντέλο SARIMAX(0,1,1)(0,1,1,12)

Η σειρά στο παραπάνω μοντέλο αντιπροσωπεύει τη σειρά του αυτεπαρκτικού μοντέλου (αριθμός χρονικών καθυστερήσεων), τον βαθμό διαφοροποίησης (πόσες φορές τα δεδομένα είχαν αφαιρεθεί οι προηγούμενες τιμές) και τη σειρά του κινούμενου μέσου μοντέλου.

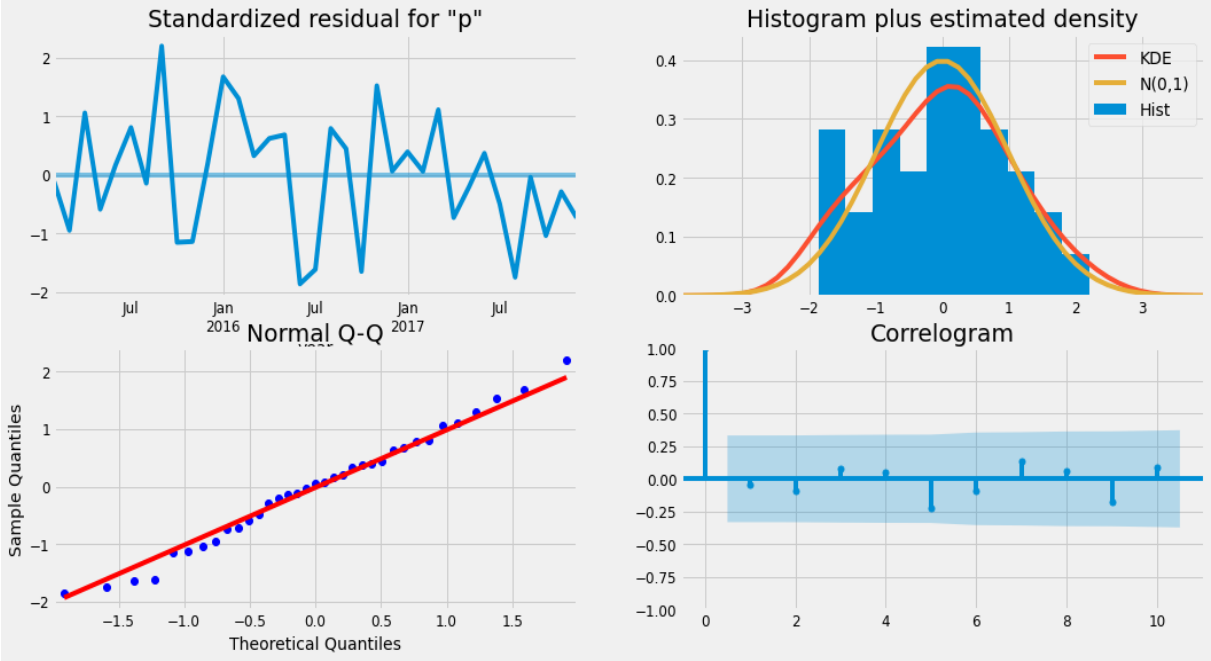
Η εποχιακή σειρά αντιπροσωπεύει τη σειρά του εποχιακού στοιχείου του μοντέλου για τις παραμέτρους AR, τις διαφορές, τις παραμέτρους MA και την περιοδικότητα.



Εικόνα 34 Πρόβλεψη τιμών για την μέθοδο SARIMA

r2_score	mean_absolute_error	median_abso	mse	msle	mape	rmse
0,79124	6,63647	5,85931	76,34798	0,00064	191,80300	8,73773

7.12.1 Διαγνωστικό Μοντέλο:



Παρατηρούμε ότι δεν υπάρχουν στατιστικά σημαντικές αυτοσυσχετίσεις στη χρονοσειρά των καταλοίπων. Επιπλέον ελέγχουμε την κανονικότητα των υπολοίπων. Αυτό

Εικόνα 35

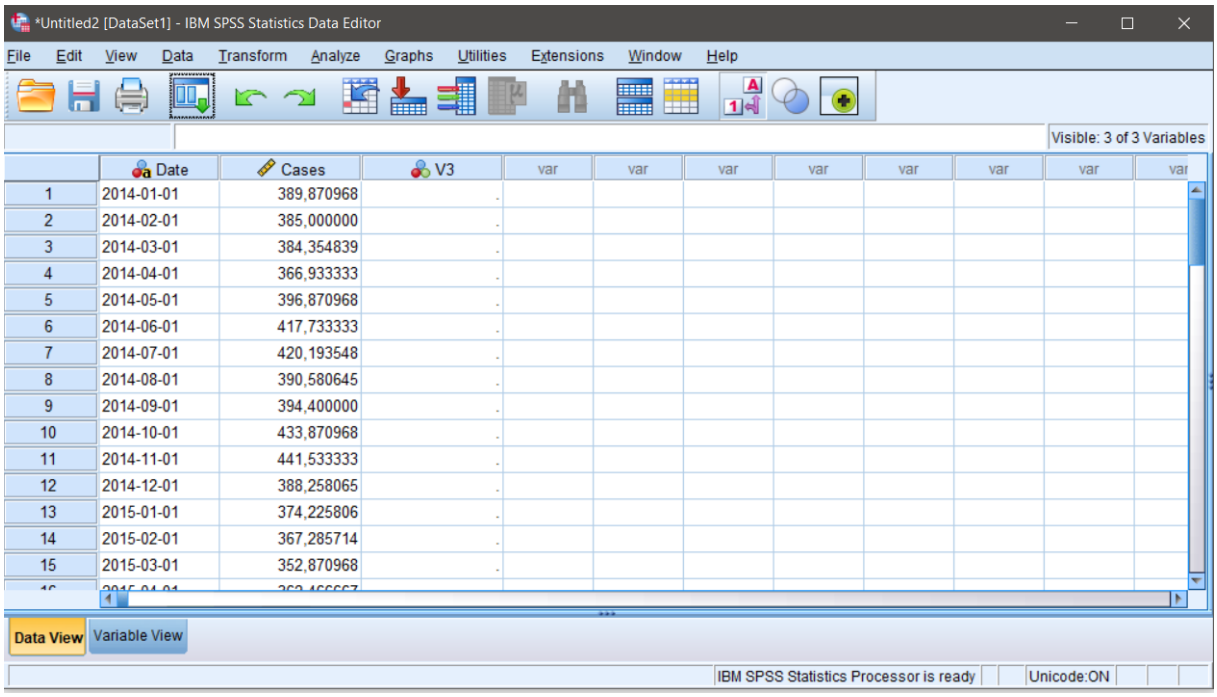
γίνεται γραφικά μέσω του QQ plot (quantile to quantile plot) των υπολοίπων. Παρατηρούμε ότι τα ποσοστιαία σημεία της κατανομής των υπολοίπων είναι πολύ κοντά στα αντίστοιχα ποσοστιαία σημεία της κανονικής κατανομής, καθώς τα περισσότερα σημεία βρίσκονται πάνω στην ευθεία γραμμή, η οποία αντιστοιχεί στην κανονική κατανομή. Συμπεραίνουμε λοιπόν ότι η χρονοσειρά των υπολοίπων είναι λευκός θόρυβος, συνεπώς δε χάνεται πληροφορία που θα μπορούσε να υπάρχει στα υπόλοιπα. Άρα μπορούμε να χρησιμοποιήσουμε το μοντέλο $SARIMA$(0,1,1)(0,1,1)12 για πρόβλεψη.

Για να ελέγξουμε αν τα υπόλοιπα είναι ανεξάρτητα μεταξύ τους εφαρμόζουμε τον έλεγχο BoxLjung. Η τιμή της στατιστικής συνάρτησης Q για τα υπόλοιπα και το αντίστοιχο $p - value$ για κάθε περιφέρεια δίνονται στον Πίνακα 5. Παρατηρούμε ότι το $p - value$ είναι μεγαλύτερο του 0.05 συνεπώς δε μπορούμε να απορρίψουμε τη μηδενική υπόθεση. Άρα μπορούμε να ισχυριστούμε σε επίπεδο σημαντικότητας 5% ότι δεν υπάρχει συσχέτιση μεταξύ των υπολοίπων για τα δεδομένα κάθε περιφέρειας. Επιπλέον, για να ελέγξουμε ότι τα δεδομένα προέρχονται από κανονικά κατανεμημένο πληθυσμό εφαρμόζουμε τον έλεγχο Jarque-Bera. Η τιμή της στατιστικής συνάρτησης JB και το αντίστοιχο $p - value$ για κάθε περιφέρεια δίνονται στον Πίνακα 5. Παρατηρούμε ξανά ότι το $p - value$ είναι μεγαλύτερο του 0.05, συνεπώς δε μπορούμε να απορρίψουμε τη μηδενική υπόθεση.

7.13 IBM SPSS Forecasting

Για να δοκιμάσουμε τα μοντέλα των προβλέψεων που φτιάξαμε από το προγραμματιστικό περιβάλλον της Python και για να δούμε πόσο αξιόπιστα είναι, θα συγκρίνουμε με μοντέλα με το λογισμικό της IBM με το όνομα SPSS Forecasting.

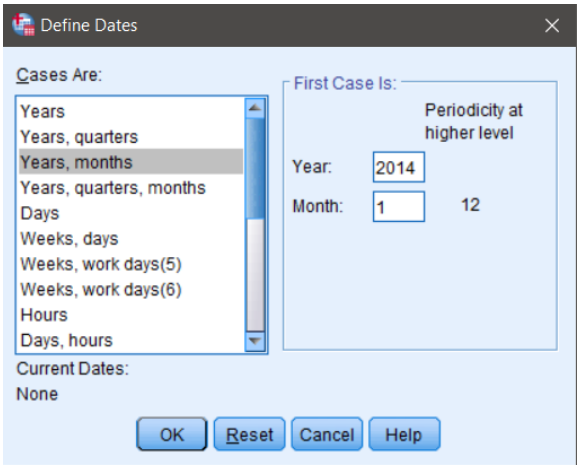
Σαν πρώτο βήμα θα κάνουμε την εισαγωγή των δεδομένων μας στην εφαρμογή.



	Date	Cases	V3
1	2014-01-01	389,870968	
2	2014-02-01	385,000000	
3	2014-03-01	384,354839	
4	2014-04-01	366,933333	
5	2014-05-01	396,870968	
6	2014-06-01	417,733333	
7	2014-07-01	420,193548	
8	2014-08-01	390,580645	
9	2014-09-01	394,400000	
10	2014-10-01	433,870968	
11	2014-11-01	441,533333	
12	2014-12-01	388,258065	
13	2015-01-01	374,225806	
14	2015-02-01	367,285714	
15	2015-03-01	352,870968	

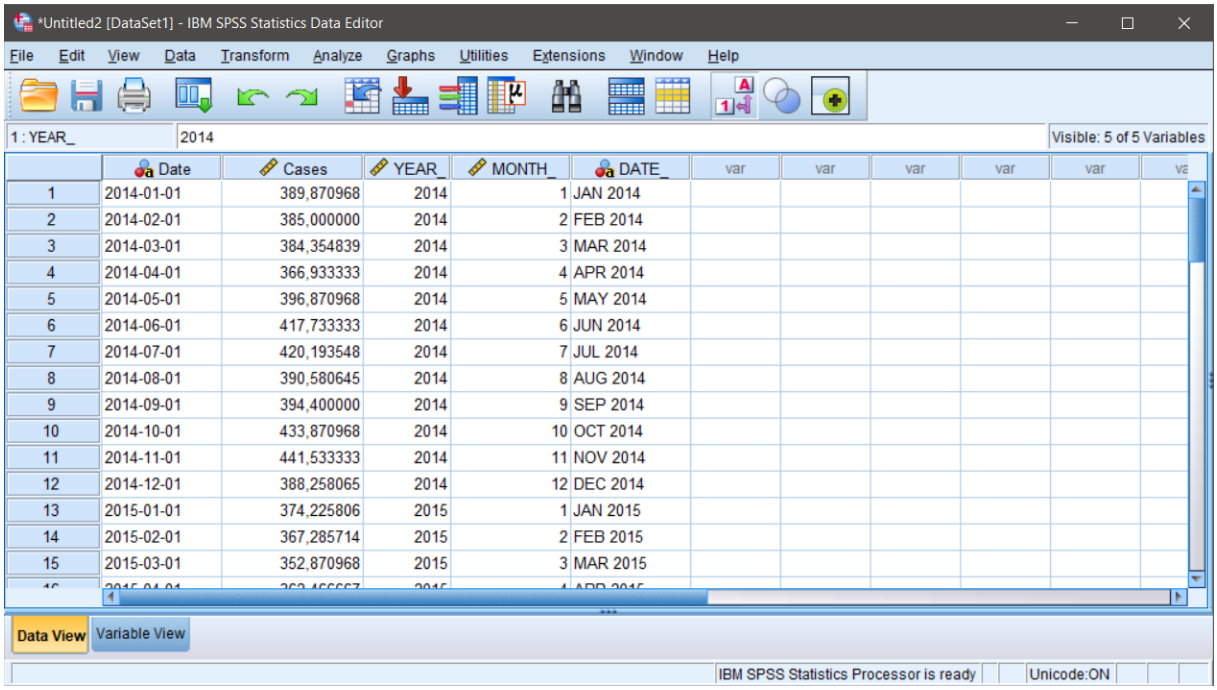
Εικόνα 36 – Εισαγωγη δεδομένων στη SPSS

Προκειμένου να εισαχθούν οι χρονολογίες στο πρόγραμμα γίνεται η επιλογή του πεδίου Data και έπειτα του Define Dates. Αφού επιλεγεί το πεδίο Data και έπειτα του Define Dates, εμφανίζεται το παράθυρο Define Dates όπου γίνεται επιλογή του πεδίου Years, quarters και έπειτα δηλώνεται αρχικό έτος το 2014 και Month το 1. Έπειτα επιλέγεται το πεδίο OK.



Εικόνα 37

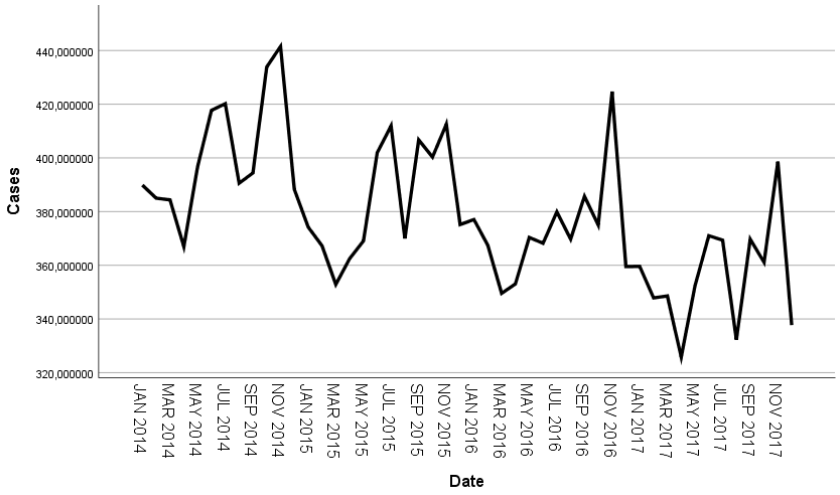
Αφού εισαχθούν οι χρονολογίες στο πρόγραμμα πλέον το παράθυρο Data View έχει τη μορφή της εικόνας 39, αφού έχουν εισαχθεί τρεις ακόμη μεταβλητές που ορίζουν, η πρώτη τα έτη, η δεύτερη τους μήνες και η τρίτη και τα δύο μαζί σε αλφαριθμητική μορφή.



	Date	Cases	YEAR_	MONTH_	DATE_	var	var	var	var	var	va
1	2014-01-01	389,870968	2014	1	JAN 2014						
2	2014-02-01	385,000000	2014	2	FEB 2014						
3	2014-03-01	384,354839	2014	3	MAR 2014						
4	2014-04-01	366,933333	2014	4	APR 2014						
5	2014-05-01	396,870968	2014	5	MAY 2014						
6	2014-06-01	417,733333	2014	6	JUN 2014						
7	2014-07-01	420,193548	2014	7	JUL 2014						
8	2014-08-01	390,580645	2014	8	AUG 2014						
9	2014-09-01	394,400000	2014	9	SEP 2014						
10	2014-10-01	433,870968	2014	10	OCT 2014						
11	2014-11-01	441,533333	2014	11	NOV 2014						
12	2014-12-01	388,258065	2014	12	DEC 2014						
13	2015-01-01	374,225806	2015	1	JAN 2015						
14	2015-02-01	367,285714	2015	2	FEB 2015						
15	2015-03-01	352,870968	2015	3	MAR 2015						

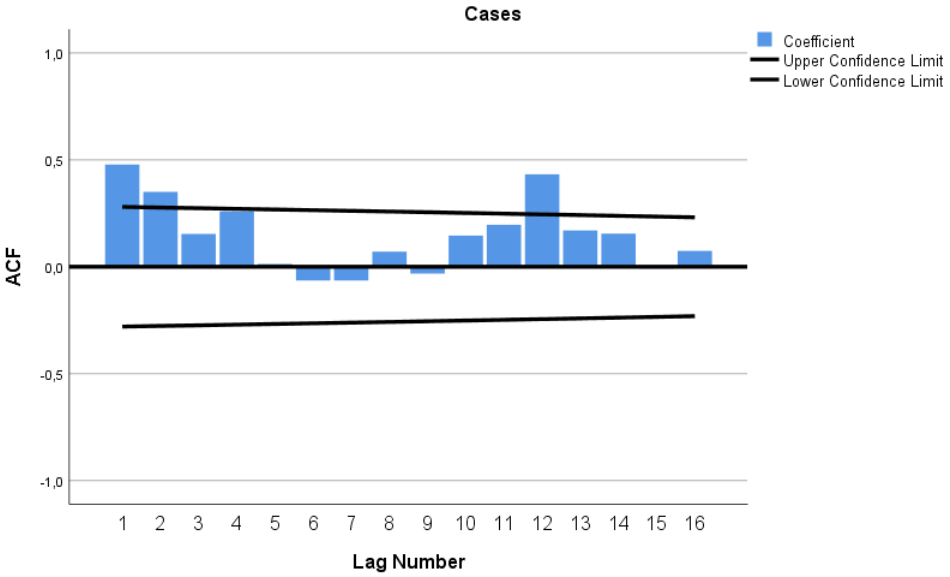
Εικόνα 38 Εισαγωγή Χρονολογιών

Αφότου πραγματοποιηθεί η εισαγωγή των δεδομένων και οριστούν οι χρόνοι γίνεται ο ορισμός των χρονολογικών σειρών μέσω της διαδικασίας Analyze Forecasting και έπειτα Sequence Charts. Αυτό πραγματοποιείται προκειμένου να πραγματοποιηθεί έλεγχος στασιμότητας.

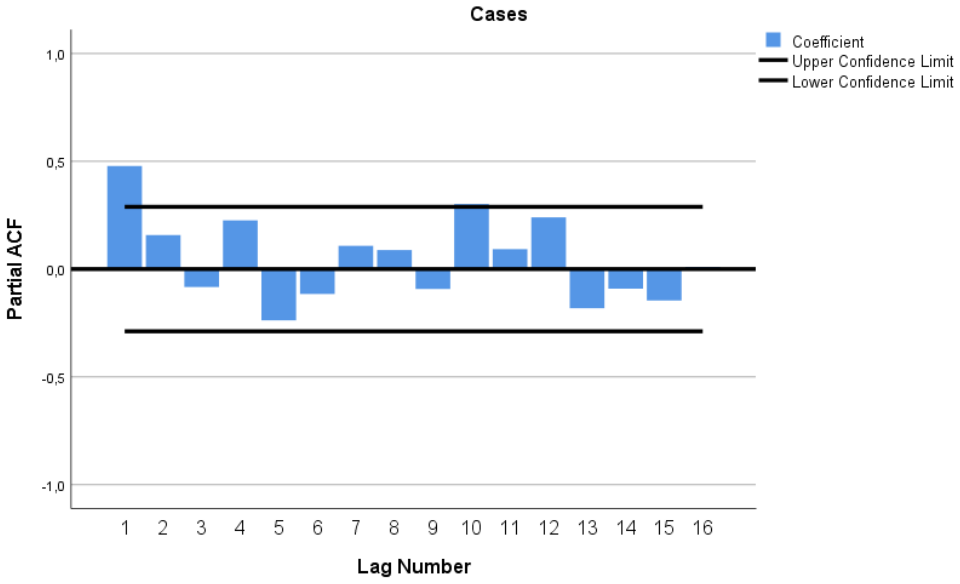


Εικόνα 39 Παρουσίαση της στασιμότητας

Στη συνέχεια θα δουμε τα γραφήματα ACF ΚΑΙ PACF για την ανάλυση της αυτοσυσχέτισης.

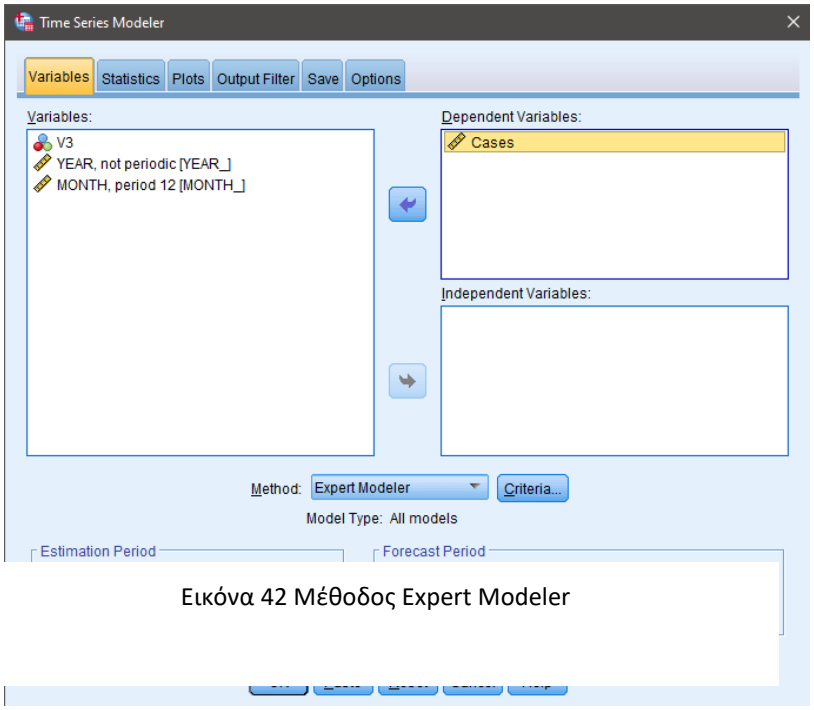


Εικόνα 40 Γράφημα αυτοσυσχέτισης ACF



Εικόνα 41 Γράφημα μερικής αυτοσυσχέτισης PACF

Για να δημιουργήσουμε τα μοντέλα πρόβλεψης μας θα χρησιμοποιήσουμε την Μέθοδο Expert Modeler. Με την μέθοδο αυτή θα φτιάξουμε 2 μοντέλα. Ένα μοντέλο SARIMA και ένα μοντέλο Exponential Smoothing, ώστε να συγκρίνουμε με τα μοντέλα που φτιάξαμε στο προγραμματιστικό περιβάλλον Python.



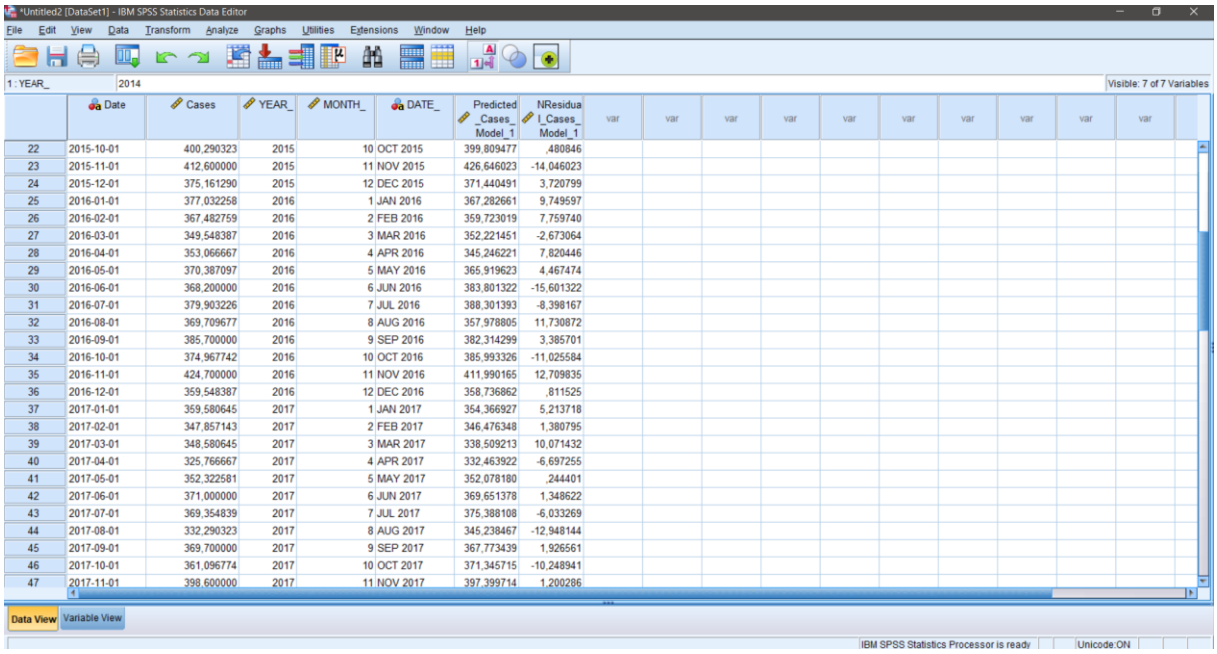
Εικόνα 42 Μέθοδος Expert Modeler

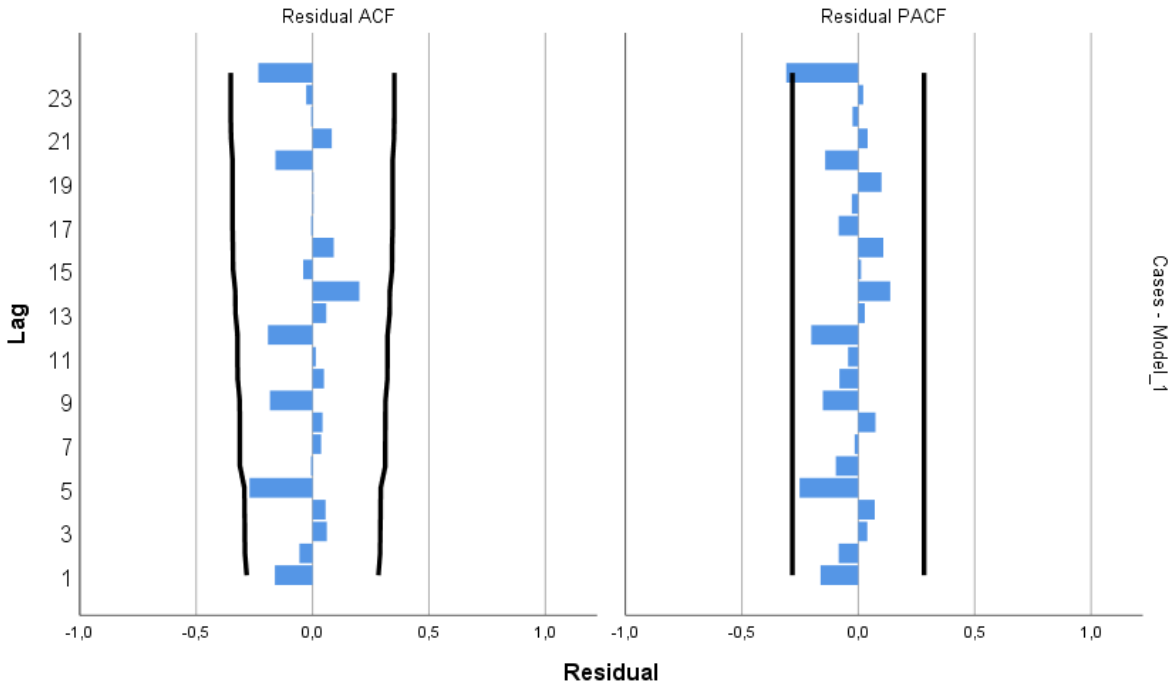
7.13.1 Winters' Additive

Το πρώτο μοντέλο που δημιουργήσαμε είναι ένα Winter's Addictive μοντέλο. Παρα

Model Description

		Model Type
Model ID	Cases	Winters' Additive
	Model_1	





Εικόνα 43 Αυτοσυσχέτιση και Μερική αυτοσυσχέτιση για την Winter's Addictive

Model Statistics				
Model	Model Fit statistics			
	R-squared	RMSE	MAPE	MAE
Cases-Model_1	,895	8,574	1,799	6,817

7.13.2 SARIMA

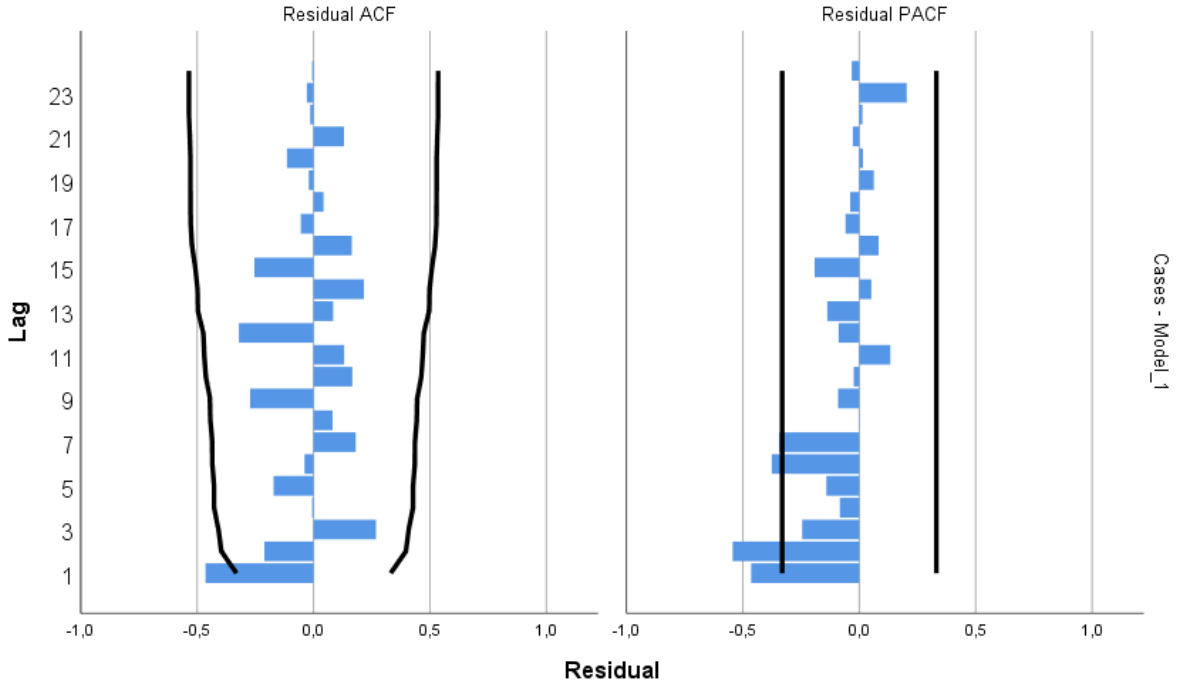
Το δευτερο μοντέλο που φτίαξαμε στο λιογισμικό είναι SARIMA(0,1,0)(0,1,0)12

Ολες τις παραμέτρους για την επιλογή του μοντελου την έχει κάνει μόνο του το λογισμικο. Με την μέθοδο βρίσκει το βέλτιστο μοντέλο. Αυτό είναι κατι χρήσιμο γιατι δίνει πρόσβαση σε εργαλεία σημαντικά χωρις την προυπόθεση της απαραίτητης γνώσης.

Ας δούμε παρακάτω κάποια χρήσιμα χαρακτηριστικά του μοντέλου μας

Model Description

		Model Type
Model ID	Cases	Model_1	ARIMA(0,1,0)(0,1,0)



Εικόνα 44 Αυτοσυσχέτιση και Μερική αυτοσυσχέτιση για την SARIMA

Model Statistics				
Model	Model Fit statistics			
	R-squared	RMSE	MAPE	MAE
Cases-Model_1	,895	8,574	1,799	6,817

Κεφάλαιο 8. ΑΠΟΤΕΛΕΣΜΑΤΑ

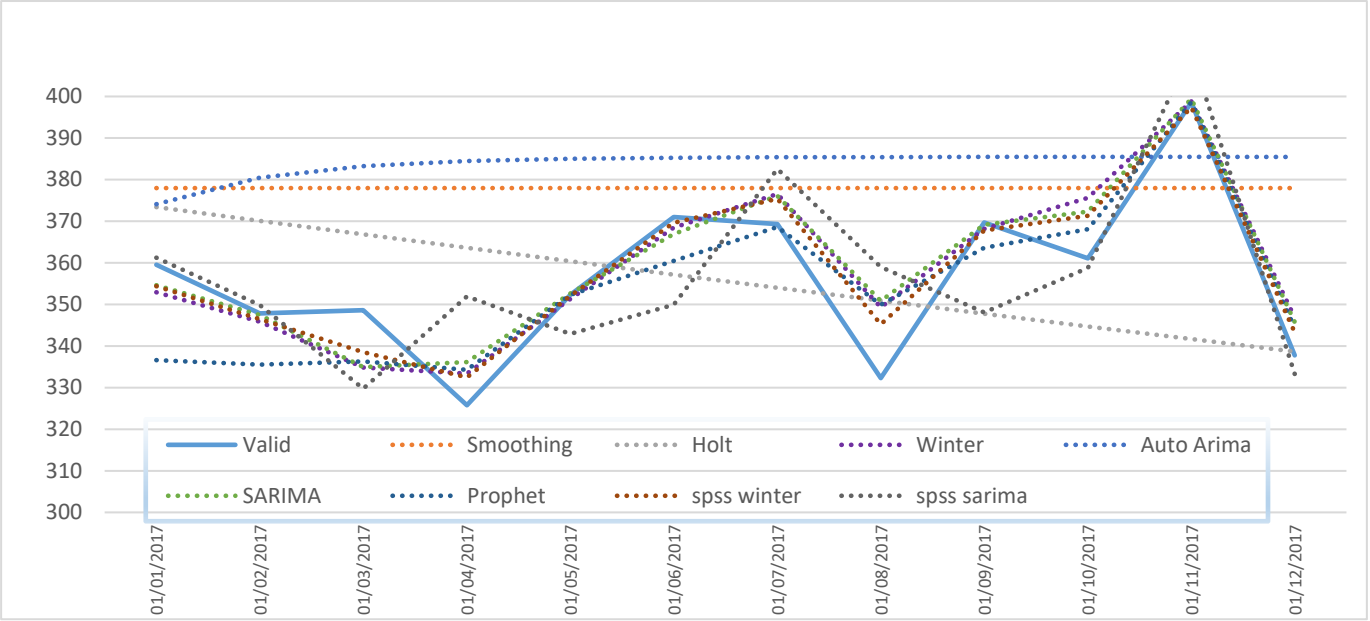
Στο κεφάλαιο αυτό θα κάνουμε αποτίμηση όλων το μοντέλων προβλέψεων που δημιουργήσαμε είτε στο προγραμματιστικό περιβάλλον της Python όπως SARIMA, PROPHET κλπ., είτε από την εφαρμογή προβλέψεων IBM SPSS Forecasting. Θα ζητήσουμε από τα μοντέλα μας να τρέξουν προβλέψεις για 24 περιόδους, δηλαδή μέχρι Δεκέμβριο του 2019.

Στο πρώτο μέρος θα δούμε τα αποτελέσματα από τις προβλέψεις που πραγματοποιήθηκαν στο σετ επικύρωσης το οποίο ήταν από 01/2017 εως 12/2017. Μπορούμε να διακρίνουμε ότι κάποια μοντέλα είναι κατά στις πραγματικές τιμές του σετ δεδομένων επικύρωσης ενώ κάποια δεν είναι και τόσο ρεαλιστικά.

Πίνακας 2 Προβλέψεις μοντέλων χρονοσειρών για το σετ επικύρωσης

Date	Valid	Smoothing	Holt	Winter	Auto Arima	SARIMA	Prophet	spss winter	spss sarima
01/01/2017	359,581	377,980	373,410	352,930	374,103	354,580	336,629	354,367	361,243
01/02/2017	347,857	377,980	370,108	345,810	380,483	347,454	335,564	346,476	349,855
01/03/2017	348,581	377,980	366,835	334,812	383,279	334,941	336,312	338,509	329,747
01/04/2017	325,767	377,980	363,591	333,376	384,505	336,177	334,276	332,464	351,923
01/05/2017	352,323	377,980	360,376	351,328	385,042	352,582	352,305	352,078	342,911
01/06/2017	371,000	377,980	357,189	368,509	385,278	366,929	360,492	369,651	349,959
01/07/2017	369,355	377,980	354,030	376,575	385,381	376,072	368,641	375,388	382,527
01/08/2017	332,290	377,980	350,900	349,317	385,426	350,849	349,954	345,238	358,985
01/09/2017	369,700	377,980	347,797	368,154	385,446	369,169	363,659	367,773	348,105
01/10/2017	361,097	377,980	344,721	375,597	385,455	372,288	368,118	371,346	358,792
01/11/2017	398,600	377,980	341,673	398,832	385,458	399,445	398,590	397,400	410,653
01/12/2017	337,742	377,980	338,651	346,876	385,460	345,754	344,372	343,307	333,272

Με το γράφημα παρακάτω μπορούμε να δούμε τις προβλέψεις και να δούμε πόσο κοντά είναι στις πραγματικές τιμές για την περίοδο αυτή.



Εικόνα 45 Συγκριτικός πίνακας τιμών του σετ επικύρωσης

Για να μπορέσουμε να συγκρίνουμε τις προβλέψεις μας μεταξύ τους στο πρώτο βήμα θα πάρουμε τις τιμές που έχουμε προβλέψει για τα δεδομένα επικύρωσης. Για κάθε μέθοδο ξεχωριστά έχουμε υπολογίσει το MAE, MSE , MAPE και το RMSE. Όπως βλέπουμε και στον Πίνακα 3 παρακάτω:

Πίνακας 3 Μέτρα απόδοσης Προβλέψεων

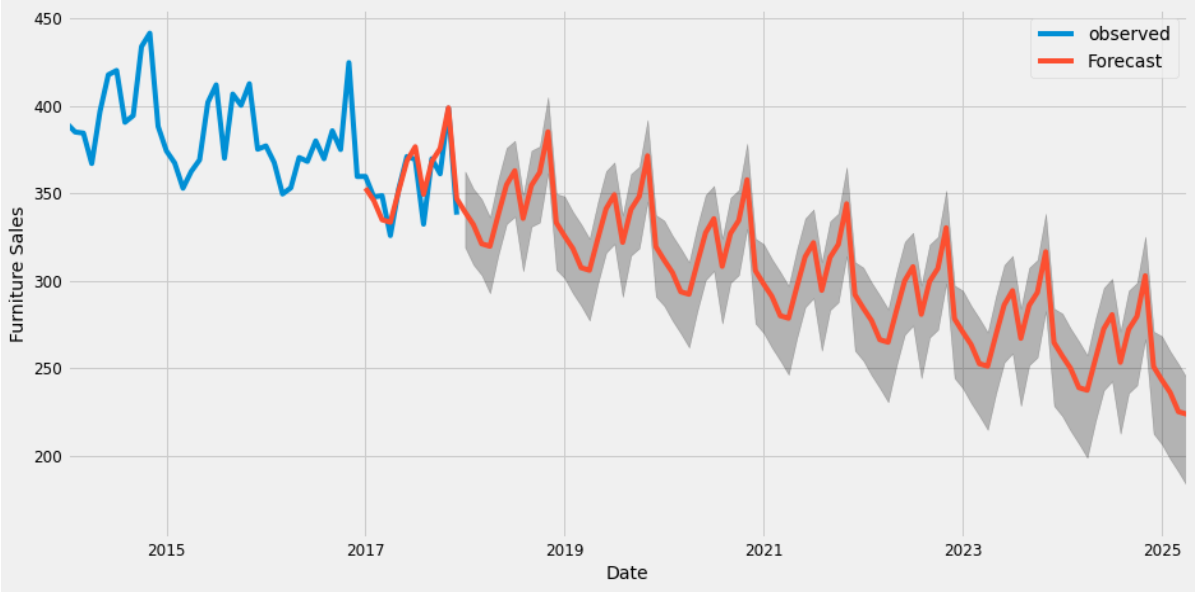
METHOD	MAE	MAPE	MSE	RMSE
SMOOTHING	25,2587	7,2907	841,9267	29,0100
HOLT	20,3394	5,6549	606,3759	24,6247
WINTERS	6,9351	1,9961	78,4401	8,8566
AR	27,1667	7,1895	29,4206	32,0635
MA	50,0545	13,0972	31,8057	56,3965
ARMA	63,6045	8,6081	4,79985	69,2809
ARIMA	29,8089	4,0388	1131,2595	33,6342
SARIMA	6,6365	1,9180	76,3480	8,7377
PROPHET	8,7190	2,4949	121,1291	11,0059
SPSS WINTER'S	6,8171	1,7992		8,5742
SPSS SARIMA	15,8312	4,2460		20,0465

Όπως διακρίνουμε και στο παραπάνω πίνακα κάποια μοντέλα όσο μικρότερο είναι το RMSE τόσο καλύτερα αποδίδει το μοντέλο πρόβλεψης και τόσο πιο αξιόπιστο είναι και μπορούμε να βασιστούμε στις προβλέψεις του. Όπως βλέπουμε στα μοντέλα που θα βασιστούμε να πάρουμε προβλέψεις και να συγκρίνουμε είναι:

1. WINTER'S
2. SARIMA
3. FB PROPHET
4. SPSS SARIMA
5. SPSS WINTER'S

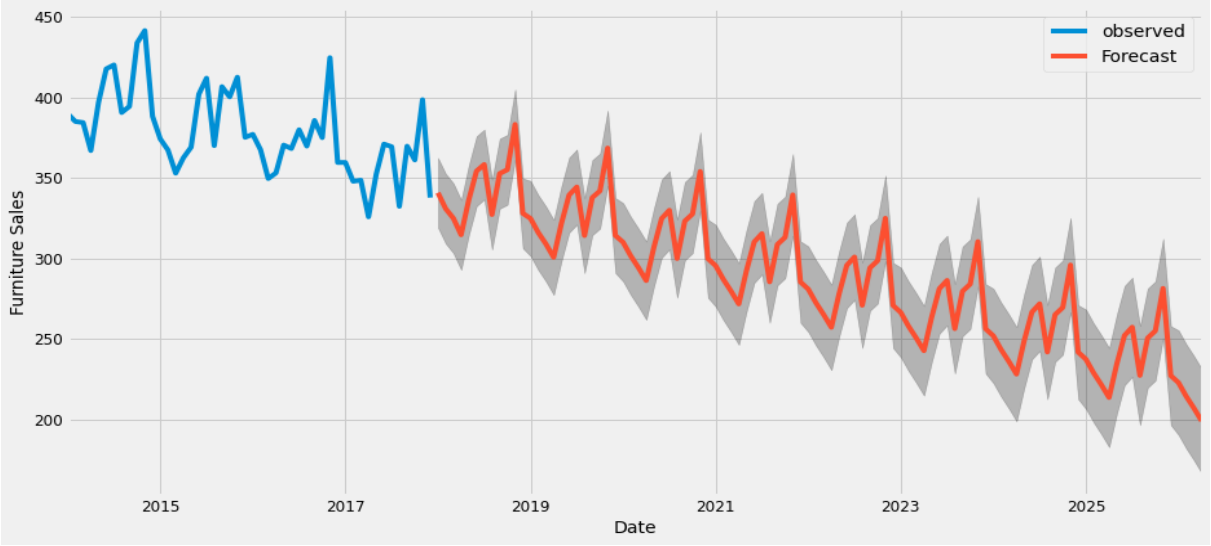
Στην συνέχεια θα δούμε τις τιμές που πήραμε κάθε μοντέλο πρόβλεψης και τη γραφική αναπαράσταση του μοντέλου πρόβλεψης

1. WINTER



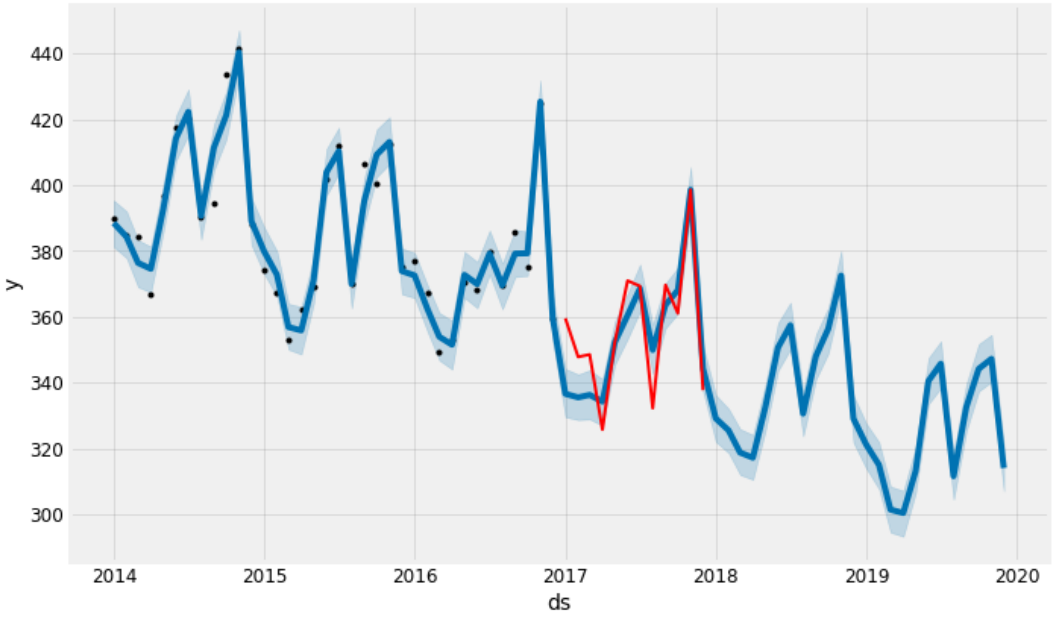
Εικόνα 46 Προβέψεις με την Μέθοδο Winter's

2. SARIMA



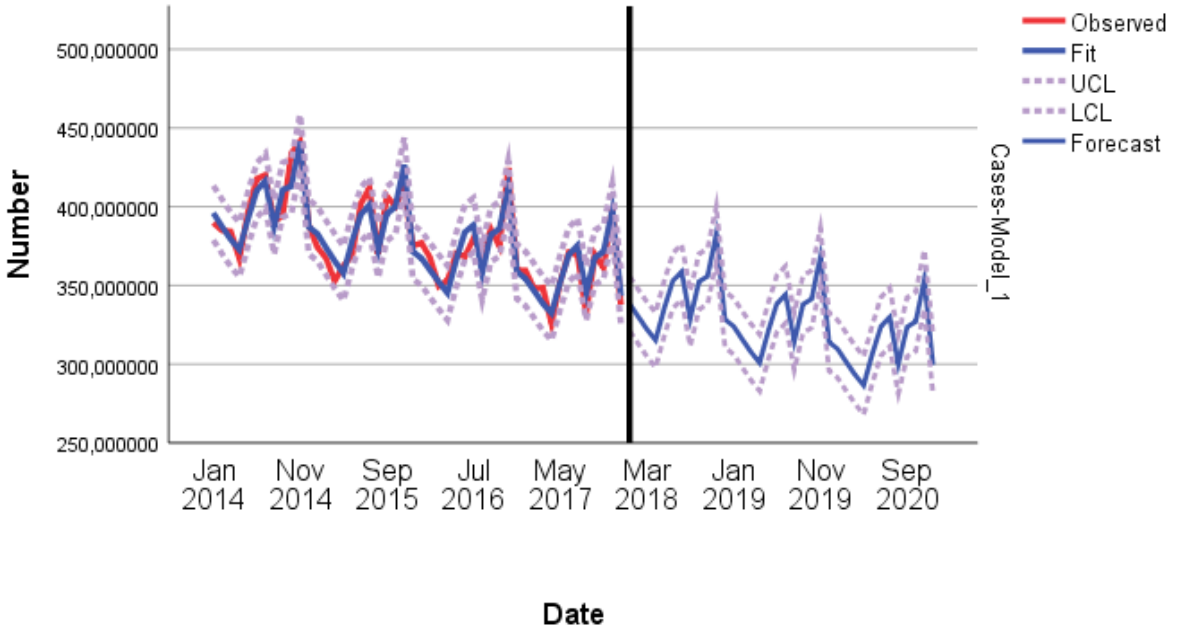
Εικόνα 47 Πρόβλεψη με την μέθοδο SARIMA

3. PROPHET



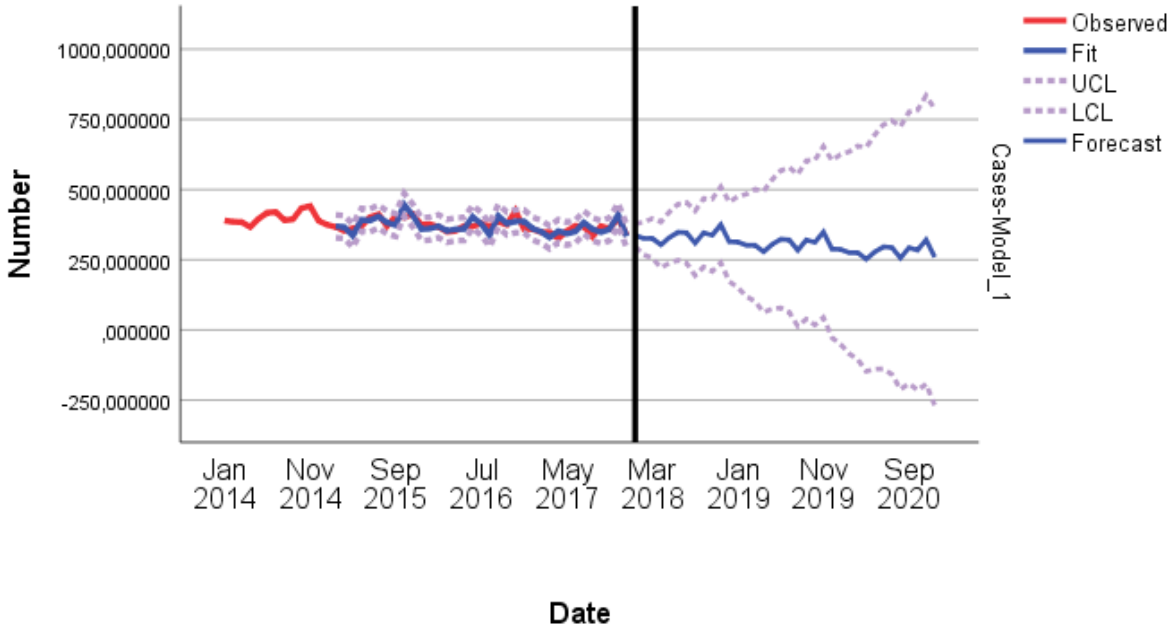
Εικόνα 48 Πρόβλεψη με την μέθοδο FB Prophet

4. SPSS WINTER'S



Εικόνα 49 Πρόβλεψη με την μέθοδο SPSS Winter's

5. SPPS SARIMA



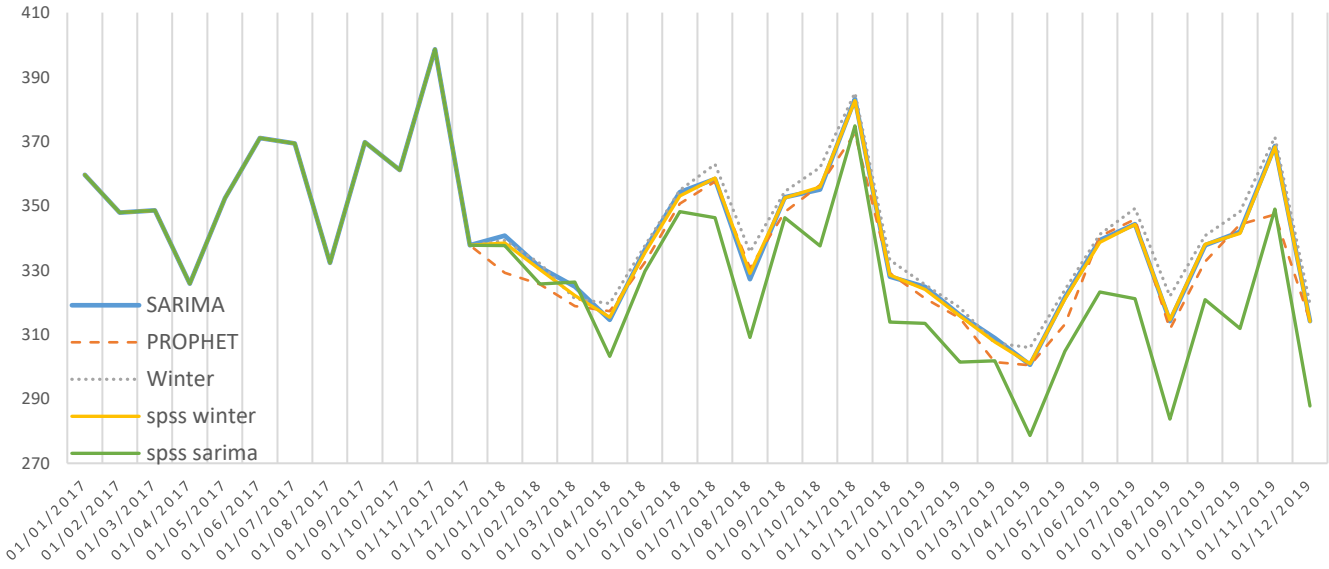
Εικόνα 50 Πρόβλεψη με την μέθοδο SPSS SARIMA

Στη συνέχεια παρουσιάζονται συγκεντρωτικά αποτελέσματα συγκρίσεων των μοντέλων για δύο έτη, από τον Ιανουάριο 2018 μέχρι τον Δεκέμβριο 2019. Ο παρακάτω πίνακας παρουσιάζει τις προβλέψεις κάθε μοντέλου για κάθε μήνα.

Πίνακας 4 Μελλοντικες Προβλέψεις ανα μοντέλο

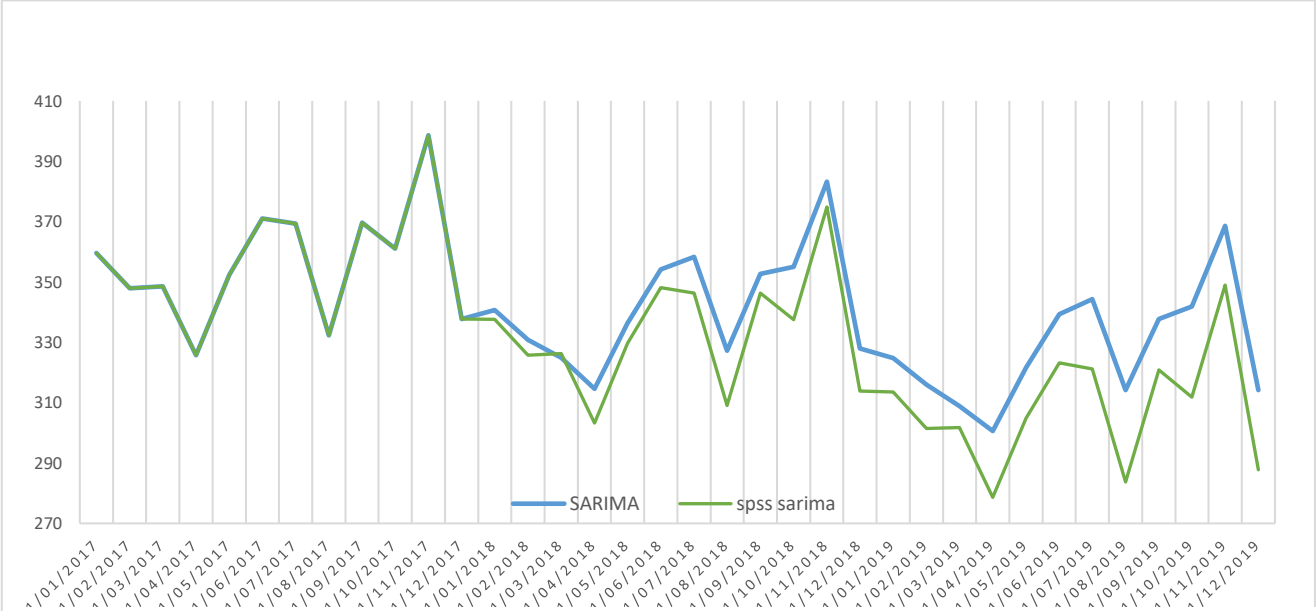
Date	SARIMA	PROPHET	WINTER'S	SPSS WINTER'S	SPSS SARIMA
01/01/2018	340,73408	329,175045	339,207	338,471549	337,598156
01/02/2018	330,85174	325,551164	332,087	330,200546	325,698617
01/03/2018	324,89063	318,804872	321,089	322,132865	326,246082
01/04/2018	314,60757	317,242362	319,653	315,352503	303,256068
01/05/2018	336,31473	332,444676	337,605	335,455475	329,635945
01/06/2018	354,20598	350,691792	354,786	353,010868	348,137327
01/07/2018	358,31977	357,479122	362,852	358,649059	346,316129
01/08/2018	327,19258	330,631956	335,594	328,939401	309,075576
01/09/2018	352,65779	348,117483	354,431	352,419261	346,309216
01/10/2018	355,06248	356,412927	361,874	355,850735	337,529953
01/11/2018	383,21874	372,55306	385,109	382,652642	374,857143
01/12/2018	327,9717	329,121918	333,153	328,471755	313,823041
01/01/2019	324,75464	321,236396	325,484	324,04283	313,503225
01/02/2019	316,02974	315,066741	318,364	315,771827	301,427649
01/03/2019	308,90316	301,47833	307,366	307,704146	301,799078
01/04/2019	300,60821	300,454516	305,930	300,923784	278,633026
01/05/2019	321,54984	313,17338	323,882	321,026756	304,836866
01/06/2019	339,31754	340,536919	341,063	338,582149	323,162211
01/07/2019	344,33825	345,792594	349,129	344,220339	321,164977
01/08/2019	314,14613	311,627032	321,871	314,510682	283,748387
01/09/2019	337,73022	332,610513	340,708	337,990542	320,80599
01/10/2019	341,86853	344,222951	348,151	341,422016	311,850691
01/11/2019	368,55276	347,27076	371,385	368,223923	349,001843
01/12/2019	314,18939	314,114	319,430	314,043035	287,791704

Η Εικόνα 52 μας δείχνει συγκριτικά τις προβλέψεις κάθε μοντέλου ανα μήνα ώστε να μπορούμε να συγκρίνουμε και την κάθε πρόβλεψη μεταξύ τους, να δούμε πως διαφέρουν και πως μπορεί ανάλογα με το λογισμικό και την παραμετροποίηση ίδιες μέθοδοι να απέχουν σημαντικά.



Εικόνα 51 Πίνακας σύγκρισης προβλέψεων

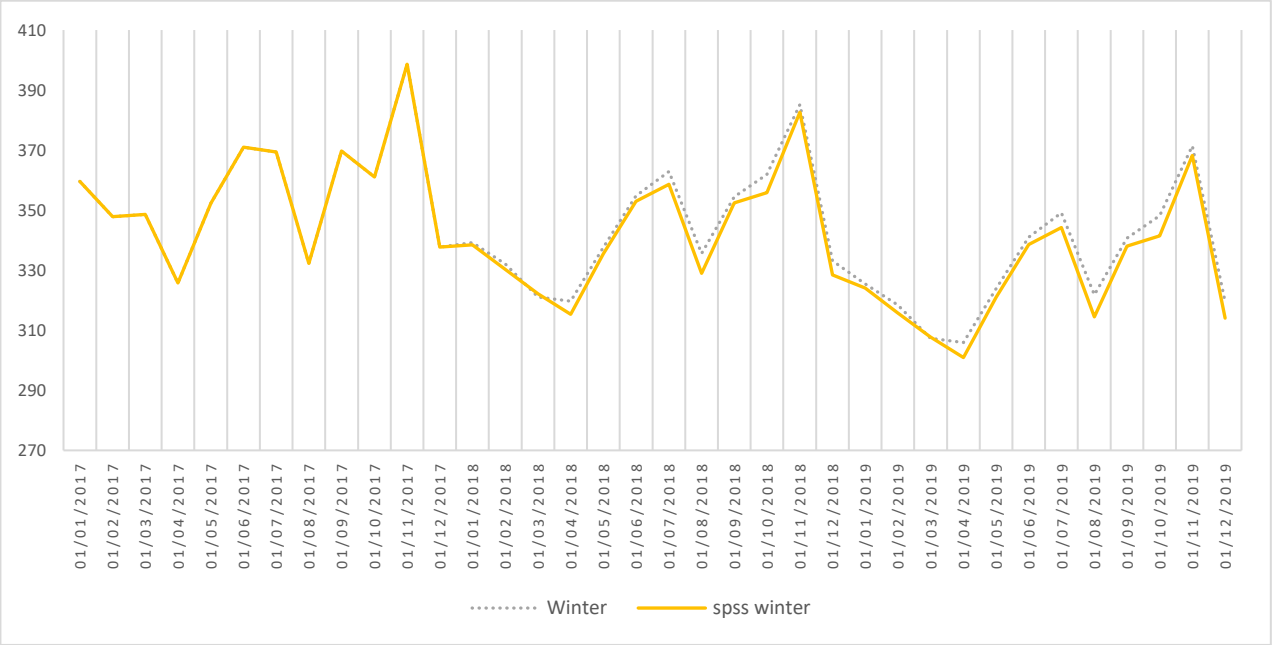
Στην εικόνα 54 παρουσιάζεται η σύγκριση μεταξύ των αποτελεσμάτων της μεθόδου SARIMA από το στο προγραμματιστικό περιβάλλον της Python (μπλέ γραμμή) και του SPSS. Όπως είναι εμφανές, τα αποτελέσματα της ίδιας μεθόδου έχουν απόκλιση ανάλογα με ποιο λογισμικό χρησιμοποιείς.



Εικόνα 52 Γράφημα συγκρισης προβλέψεων

SPSS ARIMA – ARIMA

Αντίθετα για την μέθοδο Winter’s μπορούμε να δούμε στο σχήμα ότι και οι 2 προβλέψεις είναι σχεδόν ίδιες. Αυτό είναι ένα καλό παράδειγμα ότι μπορούμε να πέτυχουμε ιδιά αποτελέσματα με λογισμικά και εφαρμογές.



Εικόνα 53 Πίνακας σύγκρισης προβλέψεων Holt Winters – SPSS Holt Winters

Κεφάλαιο 9. ΣΥΜΠΕΡΑΣΜΑΤΑ - ΠΡΟΤΑΣΕΙΣ

Έχοντας ολοκληρώσει την ανάλυση των διαφόρων μεθόδων πρόβλεψης, είμαστε σε θέση να εξάγουμε ασφαλή συμπεράσματα σχετικά με τη δυνατότητα πρόβλεψης Χρονοσειρών. Τα συμπεράσματα κινούνται σε δύο βασικούς άξονες. Ο πρώτος αφορά στη φύση των δεδομένων που χρησιμοποιούμε και πιο συγκεκριμένα το χρονικό βήμα των Χρονοσειρών. Ο δεύτερος αφορά στα συμπεράσματα για τις διάφορες κατηγορίες μεθόδων και τη δυνατότητά τους να μας εξασφαλίσουν αξιόπιστες προβλέψεις.

Η γλώσσα προγραμματισμού που επιλέξαμε να δημιουργήσουμε τα μοντέλα πρόβλεψης ήταν η Python. Η Python είναι μια εύκολη και εύχρηστη γλώσσα προγραμματισμού που με τις κατάλληλες βιβλιοθήκες μας βοήθησε να δημιουργήσουμε εύκολα όλα τα μοντέλα που περιγράψαμε στην μελέτη μας. Η δημιουργία πολλαπλών μοντέλων πρόβλεψης χρονοσειρών μας δίνει την δυνατότητα δούμε τυχόν διάφορες, αλλά και να επιλέγουμε την κατάλληλη σε κάθε περίσταση. Για την αξιολόγηση των προβλέψεων των μοντέλων μας υπάρχουν πολλές μετρικές που μπορούν να χρησιμοποιηθούν. Ωστόσο, στην συγκεκριμένη εργασία επικεντρωθήκαμε σε τέσσερις μετρικές τις MAE, MAPE, MSE, RMSE. Επιπλέον, ισχύει ότι όσο μικρότερη είναι η τιμή των παραπάνω τόσο καλύτερο είναι το συνολικό μοντέλο και πιο κοντά στις πραγματικές προβλέψεις.

Στην εργασία παρουσιάζονται εμπειρικά αποτελέσματα από την απόδοση πολλών μοντέλων. Τα αποτελέσματα που πήραμε από τα μοντέλα τα συγκρίναμε με μοντέλα που πήραμε από το λογισμικό της IBM SPSS Forecasting.

Τα αποτελέσματα μας έδειξαν ότι τόσο το μοντέλο SARIMA , το μοντέλο FB Prophet όσο και το μοντέλο Winter’s Holt μπορούν να επιτύχουν καλές προβλέψεις όταν εφαρμοστούν σωστά. Το μοντέλο Winter’s που αναπτύξαμε στο προγραμματιστικό περιβάλλον Python μαζί με το μοντέλο SARIMA απέδειξαν καλύτερή απόδοση από ότι συγκριτικά τα υπόλοιπα. Σε σύγκριση με τα παρόμοια μοντέλα του λογισμικού της IBM

μπορούμε να δούμε τα μοντέλα Winter’s έχουν σχεδόν την ίδια απόδοση, ενώ το μοντέλο SARIMA που δημιουργήσαμε εμείς έχει καλύτερη απόδοση από το αντίστοιχο του SPSS.

Δεδομένων των αποτελεσμάτων που προέκυψαν οδηγούμαστε στο συμπέρασμα ότι μπορούμε να κατασκευάσουμε παρόμοια μοντέλα προβλέψεων σε προγραμματιστικό περιβάλλον με αυτά που υπάρχουν σε λογισμικά έτοιμα προς χρήση με αντίστοιχα ή ακόμα και καλύτερα αποτελέσματα. Στο πλαίσιο βελτίωσης της εργασίας θα μπορούσαμε να βάλουμε και ένα τρίτο αλγόριθμο όπως η XGBoost, η ανάπτυξη σε μοντέλων σε ένα νέο προγραμματιστικό περιβάλλον όπως η R και να προσθέταμε μοντέλα από έτοιμα λογισμικά για σύγκριση όπως αυτό της SAP κλπ.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Σταύρος Δ. Κίγκιλος (2014). *Hellenic seaways*. 2014.
2. Nahmias, S., & Olsen, T. L. (2015). *Inventory Levels for the EOQ Model All-units discount average annual cost function*.
3. Logistics, L. (2016). *Logistics*.
    a. Çelik, A., Yaman, H., Turan, S., Kara, A., Kara, F., Zhu, B., Qu, X., Tao, Y., Zhu, Z., Dhokia, V., Nassehi, A., Newman, S. T., Zheng, L., Neville, A., Gledhill, A., `
4. Κουγιουμτζής, Δ. (2004). *Ανάλυση Χρονοσειρών*. 1–105. http://users.auth.gr/dkugiu/Teach/TimeSeriesTHMMY/TimeSeriesNotes.pdf
5. ΜΑΡΙΑ ΒΕΡΙΛΛΗ (2021). *Ανάπτυξη υπολογιστικού μοντέλου μάθησης σε βάθος για τον υπολογισμό ρύπων για βενζινοκίνητα ελαφρά και πετρελαιοκίνητα βαρεία επαγγελματικά οχήματα*.
6. Καραγάνης, Γ. (2009). *Διπλωματική Εργασία : Σύγχρονες τεχνολογίες Logistics*.
7. Σαρρής, Δ. Ν. (n.d.). *Έλεγχος ευστάθειας απόδοσης μοντέλων πρόβλεψης με χρήση τεχνικών αναδειγματοληψιας διπλωματικής εργασίας Δημήτριος ν. Σαρρής*.
8. Gakhov, A. (2018). *An Introduction to Time Series Forecasting with Python Andrii Gakhov , ferret go GmbH*. *April*. https://doi.org/10.13140/RG.2.2.18053.86249
9. *Forecasting power output of photovoltaic systems using machine learning techniques*. (2017). *November*.
10. Umke, C. H. L. D., Rock, D. A. W. B., Elms, B. R. H. H., Aff, G. G. R. H., Dumke, A. B., Brock, D. W., & Helms, B. H. (2006). *H Eart R Ate and C Ycling*. *20*(3), 601–607.
11. *Δια τμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών « Πληροφορική και Διοίκηση » Διπλωματική Εργασία : « Πρόβλεψη Πωλήσεων για Νέα Προϊόντα χωρίς Ιστορικό Πωλήσεων » Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης Ιανουάριος 2008*. (2008).
12. Fattah, J., Ezzine, L., Aman, Z., El Moussami, H., & Lachhab, A. (2018). Forecasting of demand using ARIMA model. *International Journal of Engineering Business Management*, *10*, 1–9. https://doi.org/10.1177/1847979018808673

13. Mgaya, J. F., & Yildiz, F. (2019). Application of ARIMA models in forecasting livestock products consumption in Tanzania. *Cogent Food & Agriculture*, *5*(1), 1607430. https://doi.org/10.1080/23311932.2019.1607430
14. Imran. (2015). *Analytics Vidhya forum*. https://discuss.analyticsvidhya.com/t/how-to-choose-the-value-of-k-in-knn-algorithm/2606
15. SINGH, A. (2019). *Time Series Classification with Python Code*. https://www.analyticsvidhya.com/blog/2019/01/introduction-time-series-classification/
16. Lee, T., Singh, V. P., & Cho, K. H. (2021). *Deep Learning for Time Series*. 107–131. https://doi.org/10.1007/978-3-030-64777-3_9
17. Sanayei, A., Zelinka, I., & Rössler, O. E. (2014). *ISCS 2013: Interdisciplinary Symposium on Complex Systems* (Vol. 8). http://link.springer.com/10.1007/978-3-642-45438-7
18. Pavlyshenko, B. M. (2019). Machine-learning models for sales time series forecasting. *Data*, *4*(1), 1–11. https://doi.org/10.3390/data4010015
19. ΘΩΜΑΗ ΠΑΛΙΑΡΗ(2020), Διπλωματική Εργασια: Πληροφοριακα Συστηματα Για Την Πρόβλεψη Χρονοσειρών με Χρήση Νευρωνικών Δικτύων
20. Ευσταθία Παναγοπούλου(2020), Διπλωματική Εργασια: Πρόβλεψη Χρονοσειρών με Έντονη Εποχικότητα
21. Yacine Ben Baccar(2019), Master Thesis: Comparative Study on Time Series Forecasting
22. Ευθύμιος Ι. Νικολάου(2007), Διπλωματική Εργασια: ΣΥΓΚΡΙΤΙΚΗ ΑΝΑΛΥΣΗ ΚΑΙ ΕΦΑΡΜΟΓΗ ΓΡΑΜΜΙΚΩΝ, ΜΗ-ΓΡΑΜΜΙΚΩΝ ΚΑΙ ΝΕΥΡΟ-ΑΣΑΦΩΝ ΜΕΘΟΔΩΝ, ΓΙΑ ΤΗ ΒΡΑΧΥΠΡΟΘΕΣΜΗ ΠΡΟΒΛΕΨΗ ΠΑΡΑΓΩΓΗΣ ΕΝΕΡΓΕΙΑΣ ΑΠΟ ΑΙΟΛΙΚΑ ΠΑΡΚΑ
23. Χασιρτζόγλου Μάρκος (2020), Διπλωματική Εργασια: ΑΝΑΛΥΣΗ ΚΑΙ ΠΡΟΒΛΕΨΗ ΧΡΟΝΟΣΕΙΡΩΝ ΜΕ ΜΟΝΤΕΛΑ ARIMA ΚΑΙ ΕΦΑΡΜΟΓΕΣ
24. Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and Practice. *Principles of Optimal Design*
25. Hamilton J.D., “Times Series Analysis”, Princeton University Press, 1994.
26. C. Chatfield, “The Analysis of time series: An Introduction”, Third Edition, 1984

27. Douglas C. Montgomery, “Introduction to Time Series Analysis and Forecasting”, 2008
28. https://www.kaggle.com/felixzhao/productdemandforecasting Forecasts for Product Demand - Make Accurate Forecasts for Thousands of Different Products
29. https://www.kaggle.com/veeralakrishna/predict-demand
30. https://machinelearningmastery.com/time-series-forecasting-with-prophet-in-python/
31. https://machinelearningmastery.com/time-series-forecasting-methods-in-python-cheat-sheet/
32. https://github.com/ashishpatel26/Introduction-to-Time-Series-forecasting
33. https://ucilnica.fri.uni-lj.si/mod/resource/view.php?id=28089
34. https://cran.r-project.org/web/packages/prophet/prophet.pdf
35. https://research.fb.com/blog/2017/02/prophet-forecasting-at-scale/
36. https://facebook.github.io/prophet/
37. https://github.com/ashishpatel26/Introduction-to-Time-Series-forecasting/blob/master/Time%20Series%20in%20Python.ipynb
38. https://www.kaggle.com/davidkt2/time-series-modeling-with-prophet/notebook
39. https://medium.com/rumpydas/everything-you-need-to-know-about-time-series-4a32373c4af5
40. https://www.statsmodels.org/stable/examples/notebooks/generated/tsa_arma_0.html
41. https://people.duke.edu/~rnau/411arim3.htm
42. http://www.samos.aegean.gr/actuar/amilionis/notes/TIME-%CE%9A%CE%95%CE%A6%CE%91%CE%9B%CE%91%CE%99A%204%20KAI%205-13-1-2016-SENT.pdf
43. http://users.auth.gr/dkugiu/Teach/DataAnalysis/Chp6.pdf
44. http://users.auth.gr/dkugiu/Teach/Econophysics/Chp4.pdf
45. http://www.mas.ucy.ac.cy/~fokianos/GreekRbook/timeseries.pdf

ΠΑΡΑΡΤΗΜΑ – ΚΩΔΙΚΑΣ PYTHON

```
import warnings
import itertools
import numpy as np
import matplotlib.pyplot as plt
warnings.filterwarnings("ignore")
plt.style.use('fivethirtyeight')

import pandas as pd
pd.set_option('display.expand_frame_repr', False)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
import pandas as pd
pd.set_option('display.expand_frame_repr', False)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)

import statsmodels.api as sm
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.graphics.tsaplots import plot_pacf
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.ar_model import AR
from statsmodels.tsa.arima_model import ARMA, ARIMA
from statsmodels.tsa.statespace.sarimax import SARIMAX
from fbprophet import Prophet

from math import sqrt

import matplotlib
matplotlib.rcParams['axes.labelsize'] = 14
matplotlib.rcParams['xtick.labelsize'] = 12
matplotlib.rcParams['ytick.labelsize'] = 12
matplotlib.rcParams['text.color'] = 'k'
import seaborn as sns
from pmdarima.arima import auto_arima

from random import random

from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error,
median_absolute_error, mean_squared_log_error
```

```
df = pd.read_csv('C:/Users/Ganze/Desktop/accident_UK.csv',header=None)


df.columns = ['year','passengers']


df['year'] = pd.to_datetime(df['year'], format='%d/%m/%Y')

y = df.set_index('year')

y = y.passengers.resample('MS').mean()


y.plot(figsize=(15, 6))
plt.show()


from pandas import Series
from matplotlib import pyplot
pyplot.figure(1)
pyplot.subplot(211)
y.hist()
pyplot.subplot(212)
y.plot(kind='kde')
pyplot.show()


fig, ax = plt.subplots(figsize=(15,6))
sns.boxplot(y.index.year, y, ax=ax)

fig, ax = plt.subplots(figsize=(15,6))
sns.boxplot(y.index.month, y, ax=ax)


from pylab import rcParams
rcParams['figure.figsize'] = 18, 8
decomposition = sm.tsa.seasonal_decompose(y, model='multiplicative')
fig = decomposition.plot()
plt.show()


plt.plot(y)
```

```
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.graphics.tsaplots import plot_pacf

pyplot.figure()
pyplot.subplot(211)
plot_acf(y, ax=pyplot.gca(), lags = 20)
pyplot.subplot(212)
plot_pacf(y, ax=pyplot.gca(), lags = 20)
pyplot.show()

#Determing rolling statistics
rolmean = y.rolling( window=12).mean()
rolstd = y.rolling( window=12).std()




#Plot rolling statistics:
orig = plt.plot(y, color='blue',label='Original')
mean = plt.plot(rolmean, color='red', label='Rolling Mean')
std = plt.plot(rolstd, color='black', label = 'Rolling Std')
plt.legend(loc='best')
plt.title('Rolling Mean & Standard Deviation')
plt.show(block=False)


#Perform Dickey-Fuller test:
print ('Results of Dickey-Fuller Test:')
dftest = adfuller(y, autolag='AIC')
dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags Used','Number
of Observations Used'])
for key,value in dftest[4].items():
   dfoutput['Critical Value (%s)'%key] = value
print (dfoutput)



def test_stationarity(timeseries):

   #Determing rolling statistics
   rolmean = y.rolling( window=12).mean()
   rolstd = y.rolling( window=12).std()

   #Plot rolling statistics:
   orig = plt.plot(timeseries, color='blue',label='Original')
```

```
        mean = plt.plot(rolmean, color='red', label='Rolling Mean')
        std = plt.plot(rolstd, color='black', label = 'Rolling Std')
        plt.legend(loc='best')
        plt.title('Rolling Mean & Standard Deviation')
        plt.show(block=False)

        #Perform Dickey-Fuller test:
        print ('Results of Dickey-Fuller Test:')
        dftest = adfuller(timeseries, autolag='AIC')
        dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags
Used','Number of Observations Used'])
        for key,value in dftest[4].items():
           dfoutput['Critical Value (%s)'%key] = value
        print (dfoutput)

    ts_log = np.log(y)
    plt.plot(ts_log)

    moving_avg = ts_log.rolling(12).mean()
    plt.plot(ts_log)
    plt.plot(moving_avg, color='red')


    ts_log_moving_avg_diff = ts_log - moving_avg
    ts_log_moving_avg_diff.head(12)


    ts_log_moving_avg_diff.dropna(inplace=True)
    test_stationarity(ts_log_moving_avg_diff)



    expwighted_avg = ts_log.ewm(halflife=12).mean()
    plt.plot(ts_log)
    plt.plot(expwighted_avg, color='red')


    ts_log_ewma_diff = ts_log - expwighted_avg
    test_stationarity(ts_log_ewma_diff)


    ts_log_diff = ts_log - ts_log.shift()
    plt.plot(ts_log_diff)


    ts_log_diff.dropna(inplace=True)
```

```
test_stationarity(ts_log_diff)



from statsmodels.tsa.seasonal import seasonal_decompose
decomposition = seasonal_decompose(ts_log)

trend = decomposition.trend
seasonal = decomposition.seasonal
residual = decomposition.resid

plt.subplot(411)
plt.plot(ts_log, label='Original')
plt.legend(loc='best')
plt.subplot(412)
plt.plot(trend, label='Trend')
plt.legend(loc='best')
plt.subplot(413)
plt.plot(seasonal,label='Seasonality')
plt.legend(loc='best')
plt.subplot(414)
plt.plot(residual, label='Residuals')
plt.legend(loc='best')
plt.tight_layout()


ts_log_decompose = residual
ts_log_decompose.dropna(inplace=True)
test_stationarity(ts_log_decompose)


from statsmodels.tsa.ar_model import AR
from random import random

# fit model
model = AR(ts_log_diff)
AR_Results = model.fit(disp=-1)


plt.plot(ts_log_diff)
plt.plot(AR_Results.fittedvalues, color='red')
plt.title('RSS: %.4f'% np.nansum((AR_Results.fittedvalues-ts_log_diff)**2))
plt.show()
```

```
predictions_ARIMA_diff = pd.Series(AR_Results.fittedvalues, copy=True)
print (predictions_ARIMA_diff.head())


predictions_ARIMA_diff_cumsum = predictions_ARIMA_diff.cumsum()
print (predictions_ARIMA_diff_cumsum.head())


predictions_ARIMA_log = pd.Series(ts_log.iloc[0], index=ts_log.index)
predictions_ARIMA_log =
predictions_ARIMA_log.add(predictions_ARIMA_diff_cumsum,fill_value=0)
predictions_ARIMA_log.head()


predictions_ARIMA = np.exp(predictions_ARIMA_log)



plt.plot(y)
plt.plot(predictions_ARIMA)
plt.title('RMSE: %.4f'% np.sqrt(np.nansum((predictions_ARIMA-y)**2)/len(y)))



from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error,
median_absolute_error, mean_squared_log_error


r2_score(y, predictions_ARIMA)

mean_absolute_error(y, predictions_ARIMA)
median_absolute_error(y, predictions_ARIMA)
mean_squared_error(y, predictions_ARIMA)
mean_squared_log_error(y, predictions_ARIMA)


def mean_absolute_percentage_error(y_true, y_pred):
   return np.mean(np.abs((y_true - y_pred) / y_true)) * 100


mean_absolute_percentage_error(y, predictions_ARIMA)


def evaluate_forecast(y,pred):
   results = pd.DataFrame({'r2_score':r2_score(y, pred),
                     }, index=[0])
```

```
    results['mean_absolute_error'] = mean_absolute_error(y, pred)
    results['median_absolute_error'] = median_absolute_error(y, pred)
    results['mse'] = mean_squared_error(y, pred)
    results['msle'] = mean_squared_log_error(y, pred)
    results['mape'] = mean_absolute_percentage_error(y, pred)
    results['rmse'] = np.sqrt(results['mse'])
    return results


from statsmodels.tsa.arima_model import ARMA
from random import random

# fit model
model = ARMA(ts_log_diff, order=(0, 3))
MA_results = model.fit(disp=False)


MA_results.summary()


plt.plot(ts_log_diff)
plt.plot(MA_results.fittedvalues, color='red')
plt.title('RSS: %.4f'% np.nansum((MA_results.fittedvalues-ts_log_diff)**2))



# ARMA example
from statsmodels.tsa.arima_model import ARMA
from random import random


# fit model
model = ARMA(ts_log_diff, order=(7, 1))
ARMA_Results = model.fit(disp=False)


ARMA_Results.summary()
plt.plot(ts_log_diff)
plt.plot(ARMA_Results.fittedvalues, color='red')
plt.title('RSS: %.4f'% np.nansum((ARMA_Results.fittedvalues-ts_log_diff)**2))



ts = y - y.shift()
ts.dropna(inplace=True)
```

```
pyplot.figure()
pyplot.subplot(211)
plot_acf(ts, ax=pyplot.gca(),lags=20)
pyplot.subplot(212)
plot_pacf(ts, ax=pyplot.gca(),lags=20)
pyplot.show()



#divide into train and validation set
train = y[:int(0.75*(len(y)))]
valid = y[int(0.75*(len(y))):]

#plotting the data
train.plot()
valid.plot()


from statsmodels.tsa.arima_model import ARIMA
from sklearn.metrics import mean_squared_error
from math import sqrt

# fit model
model = ARIMA(train, order=(1, 1, 1))
ARIMA_Results = model.fit(disp=1)


ARIMA_Results.summary()


start_index = valid.index.min()
end_index = valid.index.max()

#Predictions
predictions = ARIMA_Results.predict(start=start_index, end=end_index)


mse = mean_squared_error(y[start_index:end_index], predictions)
rmse = sqrt(mse)
print('RMSE: {}, MSE:{}'.format(rmse,mse))
```

```
        plt.plot(y)
        plt.plot(predictions, color='red')
        plt.title('RMSE: %.4f'% rmse)
        plt.show()



        predictions_ARIMA_diff = pd.Series(predictions, copy=True)
        print (predictions_ARIMA_diff.head())



        predictions_ARIMA_diff_cumsum = predictions_ARIMA_diff.cumsum()
        print (predictions_ARIMA_diff_cumsum.head())



        predictions_ARIMA_log = pd.Series(valid.iloc[0], index=valid.index)
        predictions_ARIMA_log =
predictions_ARIMA_log.add(predictions_ARIMA_diff_cumsum,fill_value=0)
        predictions_ARIMA_log.head()

        plt.plot(y)
        plt.plot(predictions_ARIMA_log)
        plt.title('RMSE: %.4f'% np.sqrt(np.nansum((predictions_ARIMA_log-ts)**2)/len(ts)))


        evaluate_forecast(y[start_index:end_index], predictions_ARIMA_log)



        model = auto_arima(train, trace=True, error_action='ignore',
suppress_warnings=True)
        model.fit(train)


        forecast_Arima = model.predict(n_periods=len(valid))
        forecast_Arima = pd.DataFrame(forecast_Arima,index =
valid.index,columns=['Prediction'])
        #plot the predictions for validation set
        plt.plot(y, label='Train')
        #plt.plot(valid, label='Valid')
        plt.plot(forecast_Arima, label='Prediction')
        plt.show()
```

```
evaluate_forecast(valid, forecast_Arima)


# SARIMA example
from statsmodels.tsa.statespace.sarimax import SARIMAX

# fit model
model = SARIMAX(train, order=(1, 1, 1), seasonal_order=(1, 1, 1, 12))
SARIMA_Results = model.fit(disp=False)


start_index = valid.index.min()
end_index = valid.index.max()
#Predictions
predictions = SARIMA_Results.predict(start=start_index, end=end_index)


mse = mean_squared_error(y[start_index:end_index], predictions)
rmse = sqrt(mse)
print('RMSE: {}, MSE:{}'.format(rmse,mse))



plt.plot(y)
plt.plot(predictions)
plt.title('RMSE: %.4f'% rmse)

evaluate_forecast(y[start_index:end_index], predictions)


model = auto_arima(train, trace=True, error_action='ignore',
suppress_warnings=True, seasonal=True, m=12, stepwise=True)
model.fit(train)


start_index = valid.index.min()
end_index = valid.index.max()
#Predictions
pred = model.predict()


pred = model.predict(n_periods=len(valid))
pred = pd.DataFrame(pred,index = valid.index,columns=['Prediction'])
```

```
        forecast_Sarima = model.predict(n_periods=len(valid))
        forecast_Sarima = pd.DataFrame(forecast_Sarima,index = valid.index,columns=['Prediction'])

        #plot the predictions for validation set
        plt.plot(y, label='Train')
        #plt.plot(valid, label='Valid')
        plt.plot(forecast_Sarima, label='Prediction')
        plt.show()


        evaluate_forecast(y[start_index:end_index], forecast_Sarima)



        p = d = q = range(0, 2)
        pdq = list(itertools.product(p, d, q))
        seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))]
        print('Examples of parameter combinations for Seasonal ARIMA...')
        print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
        print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
        print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
        print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))


        min_aic = 999999999
        for param in pdq:
           for param_seasonal in seasonal_pdq:
              try:
                 mod = sm.tsa.statespace.SARIMAX(y,
                                       order=param,
                                       seasonal_order=param_seasonal,
                                       enforce_invertibility=False)

                 results = mod.fit()

                 print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal, results.aic))
                 if results.aic < min_aic:
                    min_aic = results.aic
                    min_aic_model = results
              except:
                 continue
```

```
min_aic_model.summary()


start_index = valid.index.min()
end_index = valid.index.max()
#Predictions
pred = min_aic_model.get_prediction(start=start_index,end=end_index,
dynamic=False)


pred_ci = pred.conf_int()
ax = y['2014':].plot(label='observed')
pred.predicted_mean.plot(ax=ax, label='Forecast', alpha=.7, figsize=(14, 7))
ax.fill_between(pred_ci.index,
        pred_ci.iloc[:, 0],
        pred_ci.iloc[:, 1], color='k', alpha=.2)
ax.set_xlabel('Date')
ax.set_ylabel('Passengers')
plt.legend()
plt.show()

results.plot_diagnostics(figsize=(16, 8))
plt.show()



y_forecasted = pred.predicted_mean.values
y_truth = y[start_index:end_index].values
mse = ((y_forecasted - y_truth) ** 2).mean()
print('The Mean Squared Error of our forecasts is {}'.format(round(mse, 2)))
print('The Root Mean Squared Error of our forecasts is {}'.format(round(np.sqrt(mse),
2)))
evaluate_forecast(y_truth, y_forecasted)

pred_uc = results.get_forecast(steps=100)
pred_ci = pred_uc.conf_int()
ax = y.plot(label='observed', figsize=(14, 7))
pred_uc.predicted_mean.plot(ax=ax, label='Forecast')
ax.fill_between(pred_ci.index,
        pred_ci.iloc[:, 0],
        pred_ci.iloc[:, 1], color='k', alpha=.25)
ax.set_xlabel('Date')
ax.set_ylabel('Furniture Sales')
plt.legend()
plt.show()
```

```
x=results.forecast(steps=120)



=================SMOOTHING================

from statsmodels.tsa.api import ExponentialSmoothing, SimpleExpSmoothing, Holt
y_hat_avg = valid.copy()
fit1 = SimpleExpSmoothing(train).fit(smoothing_level=0.6,optimized=False)
y_hat_avg['SES'] = fit1.forecast(len(valid))
plt.figure(figsize=(16,8))
plt.plot(train, label='Train')
plt.plot(valid, label='Valid')
plt.plot(y_hat_avg['SES'], label='SES')
plt.legend(loc='best')
plt.show()

rms = sqrt(mean_squared_error(valid, y_hat_avg.SES))
print(rms)


from statsmodels.tsa.holtwinters import ExponentialSmoothing
fit2 = ExponentialSmoothing(train,trend='multiplicative', seasonal=None)
SES_Results = fit2.fit(smoothing_level=0.5, smoothing_slope=0.01, optimized=False)
print(SES_Results.params)
y_hat_holt = valid.copy()
y_hat_holt['holt_forecast'] = SES_Results.forecast(len(valid))

plt.figure(figsize=(16,8))
plt.plot( train, label='Train')
plt.plot(valid, label='Test')
plt.plot(y_hat_holt['holt_forecast'], label='Holt\'s exponential smoothing forecast')
plt.legend(loc='best')
plt.title('Holt\'s Exponential Smoothing Method')
plt.show()

rms = sqrt(mean_squared_error(valid, y_hat_holt['holt_forecast']))
print(rms)



y_hat_win = valid.copy()
fit3 = ExponentialSmoothing(train ,seasonal_periods=12 ,trend='add',
seasonal='add',).fit()
y_hat_win['Holt_Winter'] = fit3.forecast(len(valid))
plt.figure(figsize=(16,8))
```

```
plt.plot( train, label='Train')
plt.plot(valid, label='Valid')
plt.plot(y_hat_win['Holt_Winter'], label='Holt_Winter')
plt.legend(loc='best')
plt.show()


rms = sqrt(mean_squared_error(valid, y_hat_win['Holt_Winter']))
print(rms)


================PROPHET================

train.head()

train_prophet = pd.DataFrame()
train_prophet['ds'] = train.index
train_prophet['y'] = train.values


train_prophet.head()



from fbprophet import Prophet

#instantiate Prophet with only yearly seasonality as our data is monthly
model = Prophet( yearly_seasonality=True, seasonality_mode = 'multiplicative',holidays=holidays)
model.add_country_holidays(country_name='US')

model.fit(train_prophet) #fit the model with your dataframe



# predict for five months in the furure and MS - month start is the frequency
future = model.make_future_dataframe(periods = 36, freq = 'MS')
future.tail()

forecast = model.predict(n_periods=len(valid))
forecast = pd.DataFrame(forecast,index = valid.index,columns=['Prediction'])
#plot the predictions for validation set

forecast = model.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

```
fig = model.plot(forecast)
#plot the predictions for validation set
plt.plot(valid, label='Valid', color = 'red', linewidth = 2)
plt.show()


model.plot_components(forecast);


y_prophet = pd.DataFrame()
y_prophet['ds'] = y.index
y_prophet['y'] = y.values


y_prophet = y_prophet.set_index('ds')
forecast_prophet = forecast.set_index('ds')

evaluate_forecast(y_prophet.y[start_index:end_index],
forecast_prophet.yhat[start_index:end_index])

from fbprophet.diagnostics import cross_validation
df_cv = cross_validation(fig, initial='730 days', period='180 days', horizon = '365 days')
df_cv.head()

from fbprophet.diagnostics import performance_metrics
df_p = performance_metrics(df_cv)
df_p.head()
```