



ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΕΛΛΑΔΟΣ, ΤΜΗΜΑ ΟΡΓΑΝΩΣΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ
Πρόγραμμα Μεταπτυχιακών Σπουδών στη Διοίκηση Επιχειρήσεων (ΜΒΑ)

ΚΑΤΕΥΘΥΝΣΗ : [ΚΟΣΤΟΛΟΓΗΣΗ ΚΑΙ ΕΛΕΓΚΤΙΚΗ]

**Σχεδιασμός και ανάπτυξη εφαρμογής χρήσης
μεθόδων πρόβλεψης με τη γλώσσα
προγραμματισμού Python**

Η εργασία (Thesis) υποβάλλεται για τη μερική κάλυψη των απαιτήσεων για την
απόκτηση Μεταπτυχιακού Διπλώματος Ειδίκευσης

[Μπουταχίδης Παναγιώτης]

A.M. : [352]

Επιβλέπων Καθηγητές : Καθηγητής Δημητριάδης Σωτήριος

Καθηγητής Κεακής Ευάγγελος

[Μήνας και Έτος Υποβολής]

Δήλωση μη λογοκλοπής και ανάληψη προσωπικής ευθύνης

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Μεταπτυχιακής Διπλωματικής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Διπλωματική μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Μεταπτυχιακού Τίτλου των Μεταπτυχιακών Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Μεταπτυχιακή Διπλωματική Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

Όνομα & Επώνυμο Συγγραφέα (Με Κεφαλαία) :

ΠΑΝΑΓΙΩΤΗΣ ΜΠΟΥΤΑΧΙΔΗΣ

Υπογραφή (Ολογράφως, χωρίς μονογραφή) :

Ημερομηνία (Ημέρα - Μήνας - Έτος) :

Πίνακας περιεχομένων

1. Εισαγωγή
2. Βασικές έννοιες πρόβλεψης
 - 2.1. Τι είναι πρόβλεψη
 - 2.2. Αρχές προβλέψεων
 - 2.3. Μέθοδοι πρόβλεψης
 - 2.3.1. Ποιοτικές μέθοδοι πρόβλεψης
 - 2.3.2. Ποσοτικές μέθοδοι πρόβλεψης
 - 2.3.3. Μοντέλα χρονοσειρών (time series models)
 - 2.3.3.1 Ανάλυση χρονοσειρών και χρησιμότητα δεδομένων
 - 2.3.3.2 Συνιστώσες χρονοσειρών
 - 2.3.3.3 Αριθμητικά μέτρα και στατιστικοί δείκτες χρονοσειρών
 - 2.3.3.4 Διαδικασία πρόβλεψης χρονοσειρών
 - 2.3.3.5 Αξιολόγηση υπολογισμού προβλέψεων
 - 2.3.3.6 Μέθοδοι πρόβλεψης χρονοσειρών
 - 2.3.3.7 Άλλες μέθοδοι πρόβλεψης
 - 2.4. Η Γλώσσα Προγραμματισμού Python
3. Η εφαρμογή
 - 3.1. Ανάπτυξη της εφαρμογής
 - 3.2. Εισαγωγή δεδομένων και έλεγχος εγκυρότητας
4. Σχεδιασμός περιβάλλοντος και οδηγίες χρήσης
5. Συμπεράσματα
 - 5.1. Πιθανοί τρόποι εξέλιξης
6. Βιβλιογραφία

Περίληψη

Η διπλωματική εργασία έχει σκοπό να παρουσιάσει τον σχεδιασμό και την ανάπτυξη εφαρμογής λογισμικού με τη γλώσσα προγραμματισμού Python. Η εφαρμογή θα χρησιμοποιεί τις βασικές μεθόδους πρόβλεψης (μεθόδους ανάλυσης χρονοσειρών) για την πρόβλεψη των μελλοντικών τιμών σε χρονοσειρές. Η εργασία χωρίζεται σε 2 μέρη, το θεωρητικό και το πρακτικό. Στο θεωρητικό μέρος, θα αναλυθεί η έννοια των προβλέψεων. Πιο αναλυτικά, θα δοθούν οι ορισμοί των προβλέψεων, οι τομείς στους οποίους χρησιμοποιούνται, διάφοροι μέθοδοι προβλέψεων, οι μέθοδοι ανάλυσης χρονοσειρών βασισμένα σε βιβλιογραφική επισκόπηση. Επίσης θα πραγματοποιηθεί μια ανασκόπηση αναφορικά με τη γλώσσα προγραμματισμού Python και τους λόγους για τον οποίο αυτή επιλέχθηκε. Στο πρακτικό μέρος της εργασίας, θα πραγματοποιηθεί ο σχεδιασμός και η ανάπτυξη της εφαρμογής. Θα γίνει έλεγχος εγκυρότητας και αξιοπιστίας και θα παρουσιαστούν χαρακτηριστικά παραδείγματα. Τέλος θα αναγράφονται τα συμπεράσματα που προέκυψαν, ενδεχόμενοι τρόποι βελτίωσης της εργασίας, καθώς και προτάσεις για μελλοντική έρευνα και χρήση.

Λέξεις - Κλειδιά : [Προβλέψεις,Χρονοσειρές,Python]

Abstract

The thesis aims to present the design and development of a software application using the Python programming language. The application will use the basic forecasting methods (time series analysis methods) to predict future values in time series. The paper is divided into 2 parts, theoretical and practical. In the theoretical part, the concept of forecasting will be analyzed. In more detail, the definitions of forecasts, the areas in which they are used, various forecasting methods, time series analysis methods based on literature review will be given. A review will also be provided regarding the Python programming language and the reasons why it was chosen. In the practical part of the paper, the design and development of the application will be carried out. Validity and reliability checks will be performed and typical examples will be presented. Finally, the conclusions drawn, possible ways to improve the work, and suggestions for future research and use will be listed.

Keywords: *[Predictions, Time series, Python]*

1. Εισαγωγή

Η εδραίωση και η βιωσιμότητα μίας επιχείρησης είναι άρρηκτα συνδεδεμένη με την ικανότητα λήψης ορθών επιχειρηματικών αποφάσεων, τόσο κατά το στάδιο του σχεδιασμού, όσο και κατά το στάδιο του προγραμματισμού της λειτουργίας της. Σε αυτό πρωταρχικό ρόλο καλείται να επιτελέσει η γνώση μελλοντικών τιμών διαφόρων μεγεθών, που σχετίζονται με το περιβάλλον και τα αποτελέσματα από την καθημερινή λειτουργία της επιχείρησης, ήτοι ο όγκος εσόδων και πωλήσεων η ποσότητα των προϊόντων που χρειάζεται ή πρέπει να παραγγελθούν, τυχόν νέα προϊόντα που θα παραχθούν, οι τεχνολογικές ή ακόμη και ενδεχόμενες κοινωνικοοικονομικές εξελίξεις κ.ο.κ.

Είναι αυτόδηλο ότι μία επιχείρηση από τη φύση της είναι πολυσύνθετη, λαμβάνοντας υπόψιν ότι πολλές φορές θα χρειαστεί συνεργασία μεταξύ ενός η και περισσότερων τμημάτων της για οποιαδήποτε ενέργεια ή απόφαση, ανεξάρτητα από τη σημασία ή το βραχυπρόθεσμο αυτής. Το σύνηθες σε μία επιχείρηση η οποία δραστηριοποιείται με ταχείς ρυθμούς συνίσταται στο ότι πολλές φορές εκ των πραγμάτων δεν υπάρχουν χρονικά περιθώρια απόκρισης σε μία ενέργεια, επομένως προτιμάται μία γρήγορη και φαινομενικά αποδοτική δράση, στηριζόμενη κατά βάση στο γνωστικό υπόβαθρο, τις αναγκαίες πληροφορίες που υπάρχουν και την εμπειρία του υπεύθυνου για τη λήψη αυτής.

Σε μία περίοδο, ωστόσο, η οποία χαρακτηρίζεται από αστάθεια καθώς οι εξελίξεις όχι μόνο σε οικονομικό, αλλά και σε τεχνολογικό, νομοθετικό αλλά και κοινωνικό επίπεδο είναι ραγδαίες, η λήψη αποφάσεων με βασικό γνώμονα την εμπειρία ή το λεγόμενο «ένστικτο» κρίνεται αρκετά επισφαλής. Αντίθετα, κρίσιμη θεωρείται η δυνατότητα πρόβλεψης οικονομικών συνθηκών στο μέλλον, όχι μόνο αυτών που ενδεχομένως αφορούν την ίδια την επιχείρηση ή τον οργανισμό, αλλά και της γενικότερης αγοράς στην οποία αυτή ανήκει, εκτιμώντας τις τρέχουσες συνθήκες και δεδομένα, καθώς και τις τάσεις οι οποίες διαμορφώνονται. Προκειμένου να λυθεί το ζήτημα αυτό, έχει αναδυθεί και προκρίνεται η χρήση μαθηματικών μοντέλων και επιχειρηματικών τεχνών

οι οποίες στηρίζονται σε ανάλυση, με στόχο την αποτελεσματικότητα των λαμβανόμενων από οποιαδήποτε επιχείρηση αποφάσεων.

Συνεπώς, μείζονος σημασίας είναι να μπορέσει να επιτευχθεί μία όσο το δυνατόν σωστή και ακριβής πρόβλεψη. Υπάρχουν διάφορες μέθοδοι, η επιλογή εκ των οποίων επηρεάζεται από δομικούς παράγοντες, όπως το πόσα ιστορικά δεδομένα είναι ήδη διαθέσιμα. Με βάση αυτά μπορούν να πραγματοποιηθούν προβλέψεις για το μέλλον. Το σύνολο των δεδομένων που ταξινομούνται σε μία χρονολογική σειρά φέρει τον τίτλο «Χρονοσειρά». Η συγκεκριμένη μέθοδος είναι αυτή που θα μελετηθεί στη συγκεκριμένη εργασία.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου για τη συνεχή καθοδήγηση και υποστήριξή του.

2. Βασικές έννοιες πρόβλεψης

2.1 Τι είναι πρόβλεψη

Η πρόβλεψη είναι η διαδικασία εκτίμησης των μελλοντικών γεγονότων όσο το δυνατόν ακριβέστερα, λαμβάνοντας υπόψη όλες τις διαθέσιμες πληροφορίες, όπως τα ιστορικά ή εμπειρικά δεδομένα ή/και τη γνώση επερχόμενων γεγονότων που θα μπορούσαν να την επηρεάσουν. Για κάθε επιχείρηση ξεχωριστά, αποτελεί τροχοπέδη η διαδικασία της πρόβλεψης, καθώς επιβάλλεται να πετύχει κάποιον στόχο, ώστε να μην έχει ζημιές. Με άλλα λόγια, συνδράμει στη λήψη αποφάσεων για τις μελλοντικές τους δραστηριότητες και συνήθως διαδραματίζει αναπόσπαστο ρόλο σε διάφορους τομείς της.

Η ποιότητα των δεδομένων που χρησιμοποιούνται για τις μελλοντικές αποφάσεις είναι σημαντική και εξαρτάται από τον τρόπο συλλογής και ανάλυσής τους. Δεν υπάρχει ένας τέλειος τρόπος να προβλεφθεί το μέλλον. Η πρόβλεψη είναι μια εκτίμηση που υπόκειται σε έναν αριθμό άγνωστων παραγόντων. Παρότι υπάρχουν πολλοί τρόποι προβλέψεων κανένας δεν είναι άσφαλτος. Εφόσον πάντα υπάρχει η πιθανότητα άγνωστων παραγόντων, όπως η πανδημία των τελευταίων ετών, είναι λογικό ένα σφάλμα. Οι επιχειρήσεις έχουν επίγνωση αυτής της απόκλισης. Αυτός είναι και ο λόγος που συνηθίζουν να συγκρίνουν τις προβλέψεις για μια περίοδο, με τα πραγματικά αποτελέσματα όταν φτάσει η συγκεκριμένη χρονική περίοδος. Τέλος, οι υπεύθυνοι αναλυτές, βρίσκουν τον ή τους λόγους του σφάλματος, οι οποίοι θα ληφθούν υπόψη στις επόμενες προβλέψεις.

Μία επιχείρηση απαιτείται να δημιουργήσει ένα σύστημα πρόβλεψης που να ενσωματώνει μια ποικιλία μεθόδων για την πρόβλεψη αβέβαιων γεγονότων. Για τέτοια συστήματα πρόβλεψης απαιτείται η ανάπτυξη δεξιοτήτων στην αναγνώριση των προβλημάτων πρόβλεψης, στην εφαρμογή ποικίλων μεθόδων πρόβλεψης, στην επιλογή των κατάλληλων μεθόδων για κάθε πρόβλημα και στην αξιολόγηση και ανάλυση των προσεγγίσεων πρόβλεψης με την πάροδο του χρόνου. Οι αποφάσεις

σχετικά με το τι πρέπει να προβλεφθεί πρέπει να λαμβάνονται στις πρώτες φάσεις ενός έργου πρόβλεψης. Είναι επίσης σημαντικό να σκεφτούμε τον χρονικό ορίζοντα. Θα απαιτηθούν προβλέψεις ένα μήνα πριν, έξι μήνες πριν ή δέκα χρόνια πριν; Ανάλογα με τον ορίζοντα πρόβλεψης που είναι πιο ουσιαστικός, θα απαιτηθούν διάφοροι τύποι μοντέλων. Όταν πρόκειται για προβλέψεις, είναι συνήθως προτιμότερο να χρησιμοποιείται μια αυτοματοποιημένη προσέγγιση από ό,τι χειροκίνητη.

Προτού αναλάβουν ουσιαστικές εργασίες για την ανάπτυξη προβλέψεων, οι επιχειρήσεις θα πρέπει να μιλήσουν με τα άτομα που θα χρησιμοποιήσουν τις προβλέψεις για να διασφαλίσουν ότι κατανοούν τις ανάγκες τους και τον τρόπο με τον οποίο θα χρησιμοποιηθούν οι προβλέψεις. Αφού καθοριστεί ποιες προβλέψεις απαιτούνται, πρέπει να προσδιοριστούν ή να συλλεχθούν τα δεδομένα στα οποία θα βασιστούν οι προβλέψεις. Είναι πιθανό τα δεδομένα που απαιτούνται για την πρόβλεψη να υπάρχουν ήδη. Στις μέρες μας συλλέγονται πολλά δεδομένα και η δουλειά του οικονομικού αναλυτή είναι συχνά να βρει πού και πώς αποθηκεύονται τα βασικά δεδομένα. Τα αρχεία πωλήσεων μιας εταιρείας, η ιστορική ζήτηση για ένα προϊόν ή το ποσοστό ανεργίας σε μια συγκεκριμένη περιοχή αποτελούν παραδείγματα δεδομένων. Πριν από τη δημιουργία αποτελεσματικών μεθοδολογιών πρόβλεψης, ο αναλυτής μπορεί να δαπανήσει σημαντικό χρόνο για τον εντοπισμό και τη σύγκριση προσβάσιμων δεδομένων.

2.2 Αρχές προβλέψεων

Για τη σωστή χρήση των προβλέψεων συνιστάται να ακολουθούνται κάποιες αρχές:

1. Ανάλογα με τα δεδομένα, υπάρχει διαφορετική μέθοδος πρόβλεψης. Είναι αδύνατο μία συγκεκριμένη να χρησιμοποιείται συνεχώς για κάθε κατάσταση.
2. Είναι αδύνατο να υπάρξει άσφαλτη μέθοδος. Κάθε πρόβλεψη έχει κάποιο ποσοστό σφάλματος.
3. Σκοπός είναι η ελαχιστοποίηση του σφάλματος, όχι η εξαφάνισή του. Είναι αναγκαίο να συγκρίνονται τα πραγματικά αποτελέσματα με τις προβλέψεις, έτσι ώστε να αποκτά καλύτερη εικόνα ο αναλυτής και να μπορεί να μειώσει η διαφορά μεταξύ μελλοντικά.
4. Σε όσο κοντινότερη χρονική στιγμή αφορά η πρόβλεψη, τόσο ακριβέστερη είναι.
5. Η πολυπλοκότητα πρέπει να αποφεύγεται όταν δεν υπάρχει λόγος ύπαρξης. Ένα απλό μοτίβο γίνεται κατανοητό εύκολα. Ως αποτέλεσμα, αποφεύγονται λάθη υλοποίησης, αλλά και ανακρίβειες.
6. Σε περιόδους αβεβαιότητας χρειάζεται μετριοπάθεια. Σε τέτοιες περιπτώσεις η διακύμανση κοντά στο μέσο όρο είναι η ασφαλέστερη προσέγγιση.
7. Μερικές φορές χρειάζονται περισσότεροι από ένας τρόποι πρόβλεψης.

2.3 Μέθοδοι πρόβλεψης

Υπάρχουν τρεις μέθοδοι διαδικασίας πρόβλεψης:

1. Ποιοτικές μέθοδοι ή μέθοδοι κρίσης

Αυτή η κατηγορία περιλαμβάνει προσεγγίσεις στις οποίες ένας ή περισσότεροι εμπειρογνώμονες κάνουν μια πρόβλεψη με βάση τις γνώσεις, την εμπειρία και τη διαίσθησή τους. Αυτού του είδους οι προβλέψεις είναι υποκειμενικές και έχουν ένα στοιχείο προκατάληψης.

2. Αιτιοκρατικές ή μέθοδοι ή μέθοδοι ανάλυσης των δομικών παραγόντων

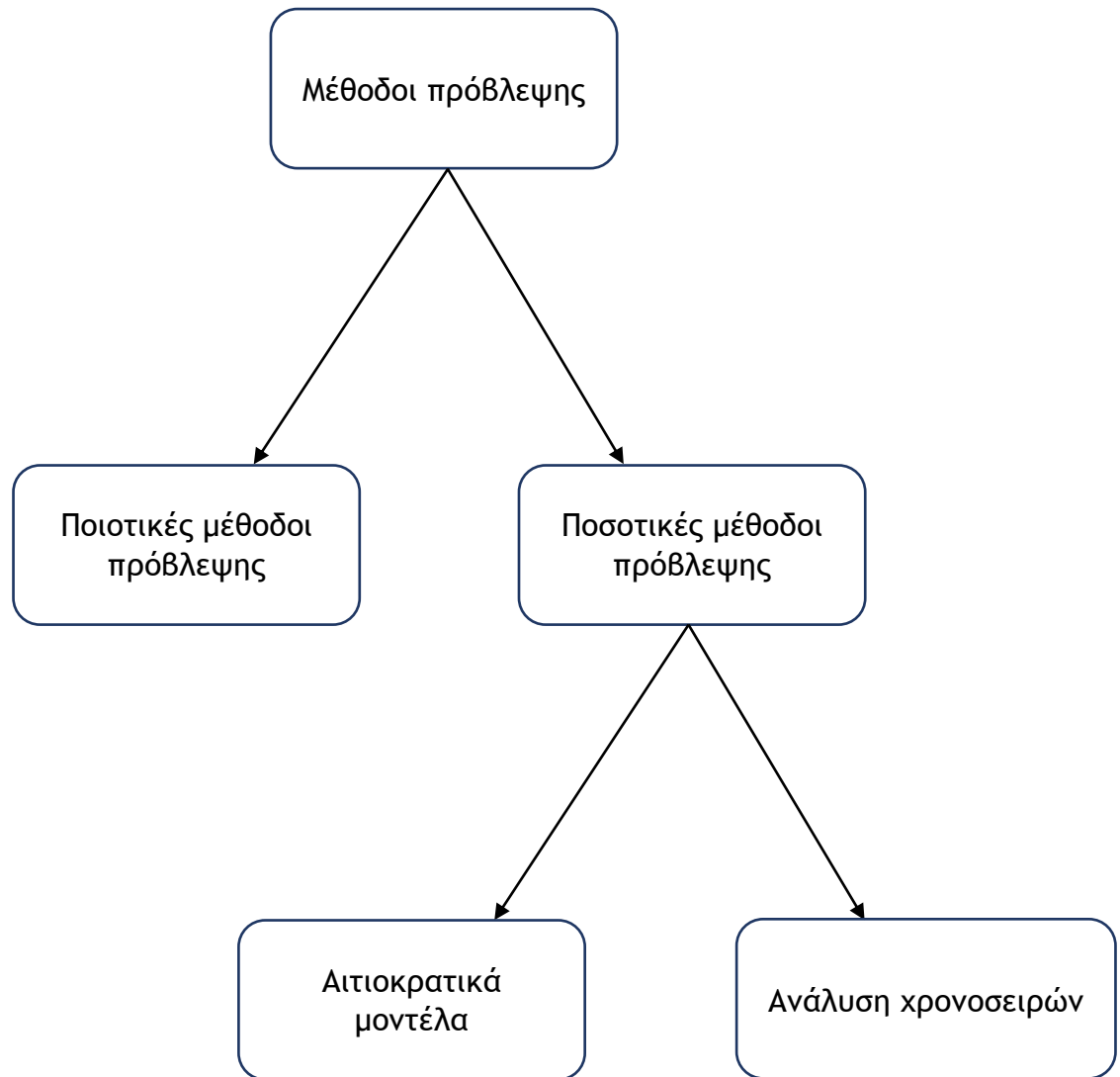
Οι διαδικασίες αυτές είναι αντικειμενικές και παράγουν το ίδιο αποτέλεσμα κάθε φορά που εισάγονται τα ίδια δεδομένα, καθώς βασίζονται σε μαθηματική μοντελοποίηση. Χρειάζονται ένα ευρύ φάσμα αριθμητικών δεδομένων. Αυτά ωστόσο δεν είναι πάντα διαθέσιμα ή αξιόπιστα. Η μεταβλητή για την οποία θέλουμε να προβλέψουμε επηρεάζεται από μία ή περισσότερες παραμέτρους με κάποιο τρόπο. Όταν λαμβάνονται υπόψη πολλαπλοί παράγοντες, το αιτιώδες μοντέλο είναι πιθανό να γίνει αρκετά σύνθετο.

Η διαφορά μεταξύ των δύο αυτών μεθόδων είναι ότι η πρώτη έχει ως βάση τα μαθηματικά, ενώ η δεύτερη την ανθρώπινη κρίση.

3. Μέθοδοι ανάλυσης χρονοσειρών και προβολής της τάσης

Σκοπός σε αυτήν την περίπτωση είναι η πρόβλεψη του μέλλοντος με τη χρήση δεδομένων του παρελθόντος. Προκειμένου να αναπτυχθεί ένα μοντέλο πρόβλεψης, η ανάλυση αυτή αναζητά μοτίβα, κυκλικότητα, περιοδικότητα και άλλα χαρακτηριστικά στα δεδομένα. Τα αιτιώδη μοντέλα χρησιμοποιούν μια πολύ διαφορετική προσέγγιση για την πρόβλεψη.

Για ευκολότερη κατανόηση, η κατηγοριοποίηση των μοντέλων προβλέψεων μπορεί να αναπαρασταθεί στο παρακάτω σχήμα:



2.3.1 Ποιοτικές μέθοδοι πρόβλεψης

Η πρώτη και πιο διαδεδομένη μέθοδος πρόβλεψης και συνίσταται στις λεγόμενες «ποιοτικές μεθόδους» (Qualitative Forecasting) ή αλλιώς αποκαλούμενες μεθόδους κρίσης. Σε αυτή την κατηγορία υπάγονται μέθοδοι πρόβλεψης κατά τις οποίες η εκάστοτε πρόβλεψη γίνεται από έναν ή περισσότερους ειδικούς με βασικό γνώμονα την κρίση, τις γνώσεις και την εμπειρία τους, ειδικά σε περιπτώσεις όπου δεν υπάρχουν αρκετά δεδομένα για την εξαγωγή ενός βέβαιου ή έστω αξιόπιστου συμπεράσματος- πρόβλεψης, όπως για παράδειγμα την τυχόν αποδοχή που θα είχε ένα νέο προϊόν στην αγορά ή τεχνολογικές εξελίξεις. Είναι προφανές ότι στην περίπτωση αυτή τόσο το υποκειμενικό στοιχείο, όσο και το στοιχείο της προκατάληψης (bias) έχει κυρίαρχο ρόλο. Αποτέλεσμα αυτής είναι η εξαγωγή κοινά αποδεκτών ποσοτικών εκτιμήσεων των μελλοντικών τιμών μίας μεταβλητής, με βάση τα χαρακτηριστικά των υπευθύνων που αναφέρονται παραπάνω, και βασίζονται τη δημοφιλία τους στο γεγονός ότι αυτές αποτελούν καρπό ατόμων ενημερωμένων πάνω στο εκάστοτε ζήτημα.

Οι πιο διαδεδομένες ποιοτικές μέθοδοι πρόβλεψης είναι οι εξής:

- **Προσωπική αντίληψη/ενόραση (personal insight):** Συνίσταται στη διατύπωση μίας πρόβλεψης από ένα άτομο το οποίο στηρίζεται μόνο στις γνώσεις, την επαγγελματική πείρα και το ένστικτο του. Μολονότι θεωρείται η πιο διαδεδομένη μέθοδος ποιοτικής πρόβλεψης, κρίνεται από τους θεωρητικούς αποφευκτέα, καθώς στερείται της απαιτούμενης αξιοπιστίας προκειμένου να σταθεί σαν μία αποτελεσματική μέθοδος.
- **Έρευνα αγοράς (market surveys):** Η μέθοδος αυτή έχει ως πρωταρχικό σκοπό τη συλλογή δεδομένων σχετικά με το ενδιαφέρον των καταναλωτών για ένα

προϊόν, τις βελτιώσεις που θα μπορούσε να έχει ή τον έλεγχο της αγοράς σχετικά με μία νέα ιδέα.

Η διεξαγωγή μίας έρευνας αγοράς συνήθως κατά βάση συνίσταται στο σχεδιασμό ερωτηματολογίων ή τη διεξαγωγή συνεντεύξεων (Naryan & Jaya, *Inventory Management: Principles & Practices*”, New Delhi, Excel Books, 2008), προκειμένου να επιτευχθεί η συλλογή όσο το δυνατόν περισσότερων δημογραφικών στοιχείων σχετικά με τον καταναλωτή ή οποιαδήποτε ομάδα - στόχο (target group), καθώς επίσης αναφορικά με το ενδιαφέρον που επέδειξαν για τα χαρακτηριστικά της υπηρεσίας ή του προϊόντος. Η έρευνα αυτή μπορεί να πραγματοποιηθεί με πληθώρα τρόπων, όπως ενδεικτικά τηλεφωνικά, με αποστολή του ερωτηματολογίου μέσω ταχυδρομείου ή μέσω προσωπικής συνέντευξης. Προκειμένου να εξασφαλιστεί η αποτελεσματικότητα της έρευνας, είναι σημαντικό να υπάρχει αντιπροσωπευτικό δείγμα από καταναλωτές ή νοικοκυριά μέσω τυχαίας επιλογής από τη συνολική αγορά της υπηρεσίας ή του προϊόντος για το οποίο πραγματοποιείται η έρευνα, ώστε να αποτελέσει ένα σημαντικό και ενδεικτικό δείγμα της κοινωνίας. Οι πληροφορίες που συλλέγονται αναλύονται με βάση στατιστικά εργαλεία, ενώ οι απαντήσεις ελέγχονται στο σύνολό τους και ερμηνεύονται προκειμένου να εξαχθεί ένα σωστό συμπέρασμα. Σε ό,τι αφορά την αποτελεσματικότητα της μεθόδου αυτής, η μεθοδολογία της έρευνας αγοράς μπορεί να χρησιμοποιηθεί για τη πρόβλεψη ζήτησης σε μικρό διάστημα (η πιο ακριβής), μεσοπρόθεσμα και μακροπρόθεσμα, ωστόσο λόγω της ραγδαίας εξέλιξης της αγοράς και του ανταγωνισμού, είναι προφανώς δύσκολο τα αποτελέσματα αυτής να είναι αξιόπιστα μακροπρόθεσμα.

Συμβούλιο στελεχών (panel consensus): Στη μέθοδο αυτή, τα μέλη μιας ομάδας ειδικών πάνω στο ζήτημα για το οποίο ζητείται μία πρόβλεψη

προβαίνουν σε μία ανοιχτή συζήτηση και ανταλλαγή εμπειριών, γνώσεων και ιδεών προκειμένου να οδηγηθούν σε μία ορθή πρόβλεψη για τα προβλεπόμενα μελλοντικά επίπεδα του μεγέθους (Anbuvelan, 2007). Η σύνθεση της ομάδας δύναται να περιέχει άτομα από οποιαδήποτε θέση, δηλαδή στελέχη όλων των επιπέδων μιας επιχείρησης. Τα δύο σημαντικότερα μειονεκτήματα της μεθόδου αυτής όμως βρίσκονται στο ότι αφενός και δεδομένου ότι παρότι καλούνται άτομα από όλες τις βαθμίδες μίας επιχείρησης, η ιεραρχία δεν παύει να ισχύει, επομένως η άποψη που εκφράζεται από στελέχη τα οποία πιθανώς να βρίσκονται εγγύτερα στον καταναλωτή και στο αντικείμενο προς πρόβλεψη, π.χ. άτομα που ενδεχομένως στελεχώνουν τμήματα πωλήσεων, υπάρχει πιθανότητα να μην επικρατήσει εκείνης των ανώτερων στελεχών, αφετέρου η συζήτηση μεταξύ πολλών ατόμων με διαφορετικές προσλαμβάνουσες είναι αναμενόμενο να οδηγήσει σε μη συγκλίνουσες απόψεις.

- **Μέθοδος Delphi (Delphi Method):** Λαμβάνοντας υπόψιν ότι στην περίπτωση χρήσης της μεθόδου του συμβουλίου στελεχών είναι πιθανότερο να επικρατήσει η άποψη ενός υπαλλήλου που βρίσκεται σε ιεραρχικά ανώτερη θέση σε σχέση με τους άλλους, υπερισχύει η Μέθοδος Delphi, βασικό χαρακτηριστικό της οποίας αποτελεί η «ανωνυμία», η οποία στηρίζεται στην απόκρυψη της ταυτότητας και των στοιχείων των συμμετεχόντων, προκειμένου κατά τη διάρκεια του σχηματισμού της πρόβλεψης όλες οι απόψεις να αντιμετωπιστούν ισάξια. Και σε αυτή τη μέθοδο πρόβλεψης κύριο σημείο αποτελεί ένα ερωτηματολόγιο, το οποίο διανέμεται από τον υπεύθυνο στους συμμετέχοντες- ειδικούς από διάφορους τομείς και θέσεις, προκειμένου να υπάρχει ένα ευρύ φάσμα γνώσεων. Οι απαντήσεις συλλέγονται και επαναπροωθούνται στους συμμετέχοντες μαζί με ένα νέο ερωτηματολόγιο, προκειμένου να βελτιωθεί και να αιτιολογηθεί η πρόβλεψή τους. Η συνηθέστερη πρακτική είναι η εμμονή των ειδικών στις απόψεις τους, είτε με λογική τεκμηρίωση είτε με ανάλογη μεταβολή τους. Το ως άνω επαναλαμβάνεται μέχρι να υπάρχει σύγκλιση απόψεων, όπου

και διανέμεται η τελική πρόβλεψη στους συμμετέχοντες. Σύμφωνα με τους θεωρητικούς, η μέθοδος αυτή φέρνει αποτελέσματα συνήθως σε τρεις με τέσσερις γύρους «αναδιανομών» του ερωτηματολογίου (feedback), με τον απαιτούμενο χρόνο για την ολοκλήρωση αυτής να τελεί σε συνάρτηση με το σύνολο των ειδικών που συμμετέχουν και το τον χρόνο και την δουλειά που χρειάζεται για να διαμορφώσουν τις προβλέψεις και να απαντήσουν επ' αυτών.

Μέθοδος Σεναρίων: Η μέθοδος αυτή υποστηρίζει ότι αποφάσεις μπορούν να εκτιμηθούν σε σχέση με σενάρια πιθανών μελλοντικών γεγονότων (plausible futures). Τα σενάρια επικεντρώνονται σε βασικές αβεβαιότητες και βεβαιότητες σχετικά με το μέλλον και χρησιμοποιούν με αυτό τον τρόπο προκειμένου να κατασκευαστούν πλήρεις πληροφοριών εικόνες (re-pictures) σαν να παρέχουν ζωντανές περιγραφές μελλοντικών κόσμων. Στην ουσία, τα σενάρια τονίζουν τον συλλογισμό που οδηγεί σε αποφάσεις και κρίσεις για το μέλλον και δίνουν μεγάλη προσοχή σε πηγές αβεβαιότητας χωρίς να προσπαθούν να μετατρέψουν την αβεβαιότητα σε πιθανότητα. Ο τρόπος σκέψης με σενάρια, μπορεί να χρησιμοποιηθεί επομένως για την εκτίμηση επιλογών αποφάσεων.

- **Μέθοδος Grass roots (Grass roots method):** Στη συγκεκριμένη μέθοδο η πρόβλεψη υλοποιείται βάσει δεδομένων προερχόμενων από όσους βρίσκονται σε άμεση επαφή στην αγορά με πελάτες και καταναλωτές. Οι πωλητές συγκεντρώνουν τα δεδομένα και τις τάσεις της περιοχής ευθύνης τους. Αυτή η μέθοδος έχει το πλεονέκτημα ότι λαμβάνονται υπόψη οι διαφοροποιήσεις της αγοραστικής τάσης και των προτιμήσεων των πελατών ή καταναλωτών ανά περιοχή.
- **Μέθοδος ιστορικής αναλογίας (historical analogy):** Η μέθοδος αυτή χρησιμοποιείται προκειμένου να προβλεφθεί η ζήτηση ενός προϊόντος σε περίπτωση μη ύπαρξης διαθέσιμων δεδομένων ζήτησης του παρελθόντος,

δηλαδή όταν πρόκειται για ένα νέο προϊόν, όπου παρουσιάζει ομοιότητες με άλλα που ενδεχομένως υφίστανται στην αγορά. Στη συγκεκριμένη μέθοδο υλοποιείται ένα είδος ιστορικής αναλογίας ανάμεσα στα δύο προϊόντα, με άντληση της ζήτησης του νέου προϊόντος, με τη χρήση των ιστορικών στοιχείων ζήτησης του υφιστάμενου προϊόντος. Το θέμα με τη μέθοδο αυτή ωστόσο έγκειται στο ότι η ομοιότητα που παρουσιάζουν οι συμπεριφορές της ζήτησης και της τοποθέτησης στην αγορά είναι ένα μέγεθος το οποίο έχει αμφισβητηθεί πολλές φορές.

2.3.2 Ποσοτικές μέθοδοι πρόβλεψης

Οι ποσοτικές μέθοδοι προβλέψεων (Quantitative forecasting) σε αντίθεση με τις ποιοτικές πραγματεύονται αριθμητικά δεδομένα και τόσο στο παρελθόν, όσο και στο μέλλον. Ανάλογα με τον τρόπο που ακολουθείται, οι συγκεκριμένες μέθοδοι πρόβλεψης χωρίζονται στα αιτιακά μοντέλα (casual models) και στις χρονοσειρές (time series models). Στα αιτιοκρατικά μοντέλα σκοπός είναι μια εξαρτημένη μεταβλητή να συσχετιστεί με μία ή περισσότερες ανεξάρτητες, ώστε να γίνει η επιθυμητή πρόβλεψη. Για παράδειγμα, οι εταιρείες συσχετίζουν την προσφορά ενός προϊόντος με τη ζητούμενη ποσότητα αλλά και την κατάσταση στην αγορά (π.χ. εισόδημα καταναλωτών).

Αντιθέτως, στις χρονοσειρές η μεταβλητή είναι μία, και οι επόμενες τιμές υπολογίζονται, βάσει των παρελθοντικών τιμών της ίδιας της μεταβλητής.

2.3.3 Μοντέλα χρονοσειρών

Η χρονοσειρά (time series) αναφέρεται σε μια πληθώρα δεδομένων, καταταγμένα με χρονολογική σειρά, που συλλέγονται με σκοπό να υποδείξουν την εξέλιξη μιας μεταβλητής στο χρόνο. Αρχικά, πραγματοποιούνται παρατηρήσεις κάποιου μεγέθους ανά συγκεκριμένες χρονικές περιόδους που απέχουν το ίδιο η μία από την άλλη. Στα συγκεκριμένα μοντέλα, τεράστια σημασία έχουν τα δεδομένα του παρελθόντος. Ο τρόπος λειτουργίας τους είναι ο εξής: Ο αναλυτής παρατηρεί τις μεταβολές μιας μεταβλητής έως μία συγκεκριμένη χρονική στιγμή, έτσι ώστε να μπορέσει να υπολογίσει τις επόμενες τιμές της μεταβλητής.

Η μεταβολή των τιμών μιας χρονοσειράς δεν είναι ποτέ τυχαία. Αντιθέτως, η ανάλυση των χρονοσειρών προϋποθέτει οι παρατηρήσεις να εξαρτώνται η μία από την άλλη, δηλαδή να μπορεί να δημιουργηθεί κάποιο μοτίβο βάσει της πρόβλεψης. Επειδή συνήθως τα δεδομένα μιας χρονοσειράς είναι πολλά και αυξάνονται με την πάροδο του χρόνου, ο αναλυτής τα αναπαριστά με δισδιάστατη γραφική μορφή συναρτήσεως του χρόνου.

Οι χρονοσειρές μπορούν να χωριστούν σε δύο κατηγορίες:

- **Συνεχείς:** Η τιμή ενός γεγονότος παρατηρείται και καταγράφεται σε όλα τα χρονικά διαστήματα, όπως η παρακολούθηση των σεισμών.
- **Διακριτές:** Η τιμή ενός γεγονότος παρατηρείται και καταγράφεται ανά τακτά χρονικά διαστήματα, όπως οι πωλήσεις ενός αυτοκινήτου.

2.3.3.1 Ανάλυση χρονοσειρών και χρησιμότητα δεδομένων

Η δουλειά ενός αναλυτή είναι πολύ σημαντική για την πορεία μιας επιχείρησης. Σκοπός είναι, οι προβλέψεις του να ανταποκρίνονται όσο περισσότερο γίνεται στην πραγματικότητα, έτσι ώστε να μειωθεί το ρίσκο και κατ' επέκταση οι ζημιές της επιχείρησης. Τα δεδομένα που απασχολούν έναν αναλυτή είναι οι πληροφορίες που συγκεντρώνονται στο οποιοδήποτε αντικείμενο μελέτης και το αντίστοιχο σημείο στο χρόνο, αλλά και οι παρατηρήσεις που πραγματοποιούνται διαχρονικά.

Κατ' αρχάς, χρειάζεται να ελέγξει πόσο ποιοτικές, αλλά και αξιόπιστες είναι οι πληροφορίες, πάνω στις οποίες θα γίνει η ανάλυση. Εάν οι πληροφορίες έχουν υπολογιστεί με μη ορθό τρόπο, πολύ πιθανό να είναι λανθασμένα τα αποτελέσματά της. Έπειτα επιλέγει τη μέθοδο που θεωρεί αποτελεσματικότερη και, αφού ολοκληρωθεί να αξιολογήσει τα αποτελέσματα.

Υπάρχουν ορισμένα κριτήρια, τα οποία μπορούν να υποδείξουν τη χρησιμότητα των δεδομένων:

- Τα δεδομένα χρειάζεται να είναι επίκαιρα και να έχουν συγκεντρωθεί σε διαδοχικές χρονικές περιόδους.
- Τα δεδομένα χρειάζεται να αναδιαμορφώνονται ανάλογα με τις επικρατούσες συνθήκες και να αντιπροσωπεύουν τα γεγονότα στα οποία αναφέρονται.
- Τα δεδομένα είναι απαραίτητα να είναι συνεπή και φερέγγυα, να έχουν συλλεχθεί με ορθό τρόπο και να προέρχονται από την αξιόπιστη πηγή.

2.3.3.2 Συνιστώσες χρονοσειρών

Υπάρχουν διαφορετικές μορφές συμπεριφοράς της μεταβλητής της χρονοσειράς, οι οποίες φέρουν το τίτλο «συνιστώσες». Υπάρχουν τέσσερις μορφές συνιστωσών:

1. Τάσης

Συχνά, οι τιμές των χρονοσειρών αυξομειώνονται με σταθερό ρυθμό για μεγάλο χρονικό διάστημα. Χρειάζεται μεγάλος αριθμός παρατηρήσεων ώστε να φανεί αν μια χρονοσειρά παρουσιάζει τάση. Σκοπός είναι να γίνουν μακροχρόνιες προβλέψεις. Σε περίπτωση που γραμμικά η χρονοσειρά κινείται παράλληλα στον άξονα του χρόνου, η τάση είναι ανύπαρκτη.

2. Κυκλική

Ονομάζεται έτσι, καθώς, γραφικά, εμφανίζονται ταλαντώσεις γύρω από την καμπύλη τάσης. Οι αλλαγές συνήθως διαρκούν για μεγάλο χρονικό διάστημα, περισσότερο από έναν χρόνο. Τις περισσότερες φορές αυτή η διάρκεια δεν είναι σταθερή. Η κυκλική κύμανση εμφανίζεται στο άνω σημείο καμψής που βρίσκεται αναμεταξύ δύο κάτω σημείων καμψής.

3. Εποχική

Οι τιμές επηρεάζονται από παράγοντες εποχικούς. Σε αυτή, την περίπτωση, εμφανίζεται επίσης κυκλική κίνηση, αλλά εμφανίζεται σε περίοδο ενός έτους. Συνήθως, κάθε φορά που πλησιάζει μια συγκεκριμένη περίοδος, η χρονοσειρά μεταβάλλεται με τον ίδιο τρόπο.

4. Τυχαία

Οι μεταβολές των τιμών δεν ακολουθούν κάποιο συγκεκριμένο μοντέλο, δηλαδή γίνονται τυχαία. Τις περισσότερες φορές, η τυχαία εμφανίζεται όταν έχουν προηγηθεί γεγονότα, τα οποία ήταν αδύνατο να προβλεφθούν προηγουμένως.

Προφανώς σε μία χρονοσειρά, δύνανται να συνυπάρξουν μία ή περισσότερες συνιστώσες. Οι τιμές της επηρεάζονται από τις συνιστώσες που την αποτελούν.

2.3.3.3 Αριθμητικά μέτρα και στατιστικοί δείκτες χρονοσειρών

Εφόσον μια χρονοσειρά αποτελείται από μια πληθώρα δεδομένων χρειάζονται αριθμητικά μέτρα και δείκτες χρονοσειρών προκειμένου υπάρξει πιο καθαρή εικόνα, ώστε να είναι ευκολότερη κατανομή αλλά και η διαχείρισή τους:

Μέση τιμή: Πρόκειται για τη μέση απόσταση ανάμεσα σε n μεταβλητές, δηλαδή το μέσο όρο. Εάν μια χρονοσειρά έχει αυξητική ή μειωτική τάση φαίνεται από τη μέση τιμή.

Μέση απόλυτη απόκλιση: Πρόκειται για το άθροισμα των αποκλίσεων κάθε μεταβλητής Y_i από τη μέση τιμή.

Μεταβλητότητα: Πρόκειται για το εύρος και η ταχύτητα διακύμανσης των τιμών. Μία απότομη αύξηση του δείχνει κίνδυνο στην αγορά.

Τυπική απόκλιση: Πρόκειται για το ποσό της μεταβολής ή της διασποράς ενός συνόλου τιμών δεδομένων.

Μέση τιμή	$\bar{Y} = \frac{1}{n} \sum_{i=1}^n Y_i$
Μέση απόλυτη απόκλιση	$MAD = \frac{1}{n} \sum_{i=1}^n Y_i - \bar{Y} $
Μεταβλητότητα	$s^2 = \frac{1}{n-1} \sum_{i=1}^n (Y_i - \bar{Y})^2$
Τυπική απόκλιση	$s = \sqrt{s^2} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (Y_i - \bar{Y})^2}$

Στασιμότητα χρονοσειράς: Όταν οι ιδιότητες μιας χρονοσειράς δεν επηρεάζονται από αλλαγές στην αρχή μέτρησης του χρόνου, τότε η χρονοσειρά χαρακτηρίζεται στάσιμη. Με άλλα λόγια, σε μία στάσιμη χρονοσειρά, η μέση τιμή και η διασπορά είναι σταθερές / δεν επηρεάζονται από το χρόνο.

Σε μια στάσιμη χρονοσειρά, τα δεδομένα εμφανίζονται γύρω από έναν μέσο. Με βάση τους ορισμούς των συνιστωσών, οι χρονοσειρές με κυκλικότητα και εποχικότητα μπορούν να επιδείξουν στασιμότητα, ενώ οι χρονοσειρές με τάση είναι αυστηρά μη στάσιμες, αφού έχουν παράγοντα το χρόνο.

Αυτοσυσχέτιση: Η συσχέτιση μιας μεταβλητής με καθυστέρηση μίας ή πολλών χρονικών περιόδων της μεταβλητής αυτής καθεαυτής, ονομάζεται αυτοσυσχέτιση. Η αυτοσυσχέτιση εμφανίζεται πάντα σε μία χρονοσειρά, αφού διάφοροι παράγοντες επηρεάζουν την ίδια τη μεταβλητή για την οποία γίνονται οι παρατηρήσεις. Μάλιστα, ο τρόπος με τον οποίο αλλάζουν αυτοί οι παράγοντες τη δεδομένη χρονική περίοδο μπορεί να εμφανίζει ομοιότητες με το παρελθόν.

Η αυτοσυσχέτιση δείχνει στον αναλυτή τη σχέση που έχουν μεταξύ τους δύο μετρήσεις που έχουν πραγματοποιηθεί σε δύο διαφορετικές περιόδους και απέχουν μία συγκεκριμένη απόσταση.

Ο συντελεστής αυτοσυσχέτισης θα μελετηθεί περισσότερο παρακάτω.

2.3.3.4 Διαδικασία πρόβλεψης χρονοσειρών

Η πρόβλεψη με τη χρήση χρονοσειρών απαιτεί να βρεθεί ο καταλληλότερος τρόπος με τον οποίο θα παρατηρούνται και θα συλλέγονται τα δεδομένα, αλλά και η διαδικασία που θα επακολουθήσει, έτσι ώστε οι τιμές των προβλέψεων να βρίσκονται κοντά στις πραγματικές.

Προκειμένου να πραγματοποιηθεί ορθά μία πρόβλεψη, είναι απαραίτητο να ακολουθούνται από τον αναλυτή τα εξής βήματα:

1. Να ορίσει το πρόβλημα. Πρόκειται για το δυσκολότερο αλλά και βασικότερο βήμα. Χρειάζεται να προσδιορίσει τον τρόπο με τον οποίο θα αξιοποιηθούν τα στοιχεία και θα γίνουν οι προβλέψεις.
2. Να συγκεντρώσει όλες τις απαραίτητες πληροφορίες, αποκλείοντας γεγονότα ελάσσονος σημασίας.
3. Να αναλύσει και να διερευνήσει τα συλλεχθέντα δεδομένα. Αρχικά, αναπαριστά τα δεδομένα με γραφική μορφή και υπολογίζει τους βασικούς στατιστικούς δείκτες. Με αυτόν τον τρόπο είναι σε θέση να γνωρίζει τη μορφή και το είδος των δεδομένων καθώς και τη συνιστώσα / τις συνιστώσες της χρονοσειράς. Τα στοιχεία αυτά είναι απολύτως απαραίτητα για την επιλογή της μεθόδου πρόβλεψης, ώστε τα αποτελέσματά της να είναι όσο το δυνατό ακριβέστερα.

4. Να επιλέξει την κατάλληλη μέθοδο και να την προσαρμόσει ανάλογα με τα δεδομένα και τους συντρέχοντες παράγοντες.

5. Να χρησιμοποιήσει το πρότυπο και να αξιολογήσει την ακρίβεια της πρόβλεψης. Σε αυτό το βήμα, πραγματοποιείται, τελικά, η πρόβλεψη. Κατά τη διαδικασία, είναι αναγκαίο να αξιολογήσει τα προτερήματα και τα ελαττώματα της μεθόδου, την οποία ακολούθησε. Εάν κριθεί απαραίτητο, ίσως επαναληφθεί κάποιο βήμα.

Η διαδικασία της πρόβλεψης δεν τερματίζει μετά το πέρας αυτών των βημάτων. Αφού περάσει ο χρόνος και φανούν οι πραγματικές τιμές, επιβάλλεται να συγκριθούν με τις προβλεπόμενες ώστε να κριθεί πόσο ακριβής ήταν η πρόβλεψη. Με βάση την ακρίβεια της πρόβλεψης, ο αναλυτής θα αποφασίσει εάν απαιτείται αλλαγή του προτύπου, της διαδικασίας πρόβλεψης ή και των δύο. Σκοπός είναι να υπάρξει ακόμη πιο αποτελεσματική μέθοδος πρόβλεψης την επόμενη φορά.

2.3.3.5 Αξιολόγηση υπολογισμού προβλέψεων

Η πρόβλεψη της επόμενης τιμής πρόκειται για το ουσιαστικότερο τμήμα της διαδικασίας ανάλυσης χρονοσειρών. Για να αξιολογηθεί η αποτελεσματικότητα μιας πρόβλεψης χρησιμοποιούνται κριτήρια αξιολόγησης. Καμία πρόβλεψη δεν μπορεί να είναι απολύτως ακριβής. Για αυτόν το λόγο, οι αναλυτές συγκρίνουν τις προβλέψεις με τα πραγματικά αποτελέσματα. Αρχικά, υπολογίζεται η απόκλιση της πραγματικής παρατηρούμενης τιμής από την προβλεπόμενη. Η διαφορά αυτή ονομάζεται σφάλμα πρόβλεψης. Έτσι ορίζονται τα παρακάτω μεγέθη:

Σφάλμα στην πρόβλεψη

$$e_t = Y_t - F_t$$

Μέσο σφάλμα

$$ME = \frac{1}{n} \sum_{t=1}^n e_t$$

Μέσο απόλυτο σφάλμα

$$MAE = \frac{1}{n} \sum_{t=1}^n |e_t|$$

Μέσο τετράγωνο σφάλματος

$$MSE = \frac{1}{n} \sum_{t=1}^n e_t^2$$

Σφάλμα στην πρόβλεψη: Η απόκλιση της πρόβλεψης από την πραγματική τιμή

Μέσο σφάλμα: Ο μέσος όρος των σφαλμάτων

Μέσο απόλυτο σφάλμα: Ο μέσος όρος της απόλυτης τιμής των σφαλμάτων

Μέσο τετράγωνο σφάλματος: Ο μέσος όρος του τετραγώνου των σφαλμάτων

Τα σφάλματα αυτά είναι συστηματικά. Επιπλέον, υπάρχουν και τα τυχαία σφάλματα. Ένα τυχαίο σφάλμα εμφανίζεται λόγω ύπαρξης μη αναμενόμενων παραγόντων. Όσο μικρότερες οι τιμές αυτών των μεγεθών, τόσο καλύτερη θεωρείται η μέθοδος.

Επιπλέον, για να έχει πιο ολοκληρωμένη εικόνα ο αναλυτής, συνηθίζει να υπολογίζει τα ποσοστά των παραπάνω μεγεθών με τη χρήση των τύπων:

Ποσοστιαίο σφάλμα	$PE_t = \frac{Y_t - F_t}{Y_t} * 100$
Μέσο ποσοστιαίο σφάλμα	$MPE = \frac{1}{n} \sum_{t=1}^n PE_t$
Μέσο απόλυτο ποσοστιαίο σφάλμα	$MAPE = \frac{1}{n} \sum_{t=1}^n PE_t $

Τέλος, η στατιστική U του Theil είναι ένα μέτρο σχετικής ακρίβειας που συγκρίνει τα προβλεπόμενα αποτελέσματα με τα αποτελέσματα της πρόβλεψης με ελάχιστα ιστορικά δεδομένα. Δίνει τεράστια βαρύτητα στα μεγάλα σφάλματα με σκοπό να αποφευχθούν στο μέλλον μέθοδοι που οδηγούν σε αυτά. Ο τρόπος υπολογισμού είναι ο εξής:

Στατιστική U του Theil:

$$U = \sqrt{\frac{\sum_{t=1}^{n-1} (FPE_{t+1} - APE_{t+1})^2}{\sum_{t=1}^{n-1} (APE_{t+1})^2}}$$

όπου

$$FPE_{t+1} = \frac{F_{t+1} - Y_t}{Y_t} \quad \text{και} \quad APE_{t+1} = \frac{Y_{t+1} - Y_t}{Y_t}$$

U = 1: όταν η συγκεκριμένη μέθοδος πρόβλεψης που χρησιμοποιούμε είναι ισοδύναμη με την απλούστερη δυνατή, όταν δηλαδή η πρόβλεψη για την επόμενη περίοδο είναι ίση με την τρέχουσα τιμή της υπό πρόβλεψη μεταβλητής. Όσο μικρότερη της μονάδας η στατιστική U τόσο καλύτερη η μέθοδος σε σχέση με την απλούστερη.

2.3.3.6 Μέθοδοι πρόβλεψης χρονοσειρών

Όπως είναι προφανές υπάρχουν μερικές διαφορετικές μέθοδοι πρόβλεψης με τη χρήση χρονοσειρών.

Αυτές είναι:

Απλός κινούμενος μέσος: Ο απλός κινούμενος μέσος όρος κάποιων k περιόδων είναι ουσιαστικά ο αριθμητικός μέσος των αυτών των k παρατηρήσεων της χρονοσειράς. Φέρει τον τίτλο κινούμενος, αφού μετά από κάθε προσθήκη μια νέας τιμής στη χρονοσειρά αλλάζει και η τιμή του. Συνήθως, οι k παρατηρήσεις είναι οι πιο πρόσφατες που υπάρχουν και όχι όλες. Εφόσον η τιμή γύρω από την οποία έχουν γίνει οι μετακινήσεις μεταβάλλεται και μετακινείται, οι παλιές παρατηρήσεις δε χρειάζεται να ληφθούν υπόψη. Ακολουθείται ο εξής τύπος:

$$F_{t+2} = \frac{Y_{t+1} + Y_t + Y_{t-1} + \dots + Y_{t-k+2}}{k}$$

$$F_{t+2} = F_{t+1} + \frac{1}{k}(Y_{t+1} - Y_{t-k+1})$$

όπου F η πρόβλεψη, Y η παρατήρηση k ο αριθμός των παρατηρήσεων και t η χρονική περίοδος.

Η μέση τιμή (μέσος όρος) των παρατηρήσεων αποτελεί τη βάση υπολογισμού της πρόβλεψης της επόμενης περιόδου t . Η κάθε πρόβλεψη δηλαδή είναι αποτέλεσμα

διόρθωσης της προηγούμενης πρόβλεψης κατά $1/k$. Όπως φαίνεται, η βαρύτητα που δίνεται σε κάθε παρατήρηση είναι ίση ακόμη κι αν απέχει αρκετά από την τωρινή. Για να αποφευχθεί αυτό το ελάττωμα και να βελτιωθεί η ακρίβεια της πρόβλεψης, μπορεί να χρησιμοποιηθεί μια διαφορετική μέθοδος πρόβλεψης.

Εκθετική εξομάλυνση: Σε αυτήν την περίπτωση, η προηγούμενη πρόβλεψη είναι ο σταθμικός μέσος όρος για τον υπολογισμό της πρόβλεψης που ακολουθεί. Δηλαδή η πρόβλεψη για την επόμενη χρονική στιγμή είναι η προηγούμενη διορθωμένη κατά μία παράμετρο α , με $0 < \alpha < 1$. Το α ονομάζεται σταθερά εξομάλυνσης.

$$F_{t+1} = F_t + \alpha(Y_t - F_t)$$

Έτσι, οι παλαιότερες τιμές, έχουν μικρότερη βαρύτητα, ενώ οι νεότερες μεγαλύτερη σε αντίθεση με την προηγούμενη περίπτωση.

Οι δύο αυτές μέθοδοι χρησιμοποιούνται σε χρονοσειρές με οριζόντια συνιστώσα. Για την περίπτωση χρονοσειρών με τάση έχουν οριστεί τρεις διαφορετικές μέθοδοι.

Διπλός κινούμενος μέσος: Με βάση τον απλό κινούμενο μέσο δημιουργείται ένας δεύτερος κινούμενος μέσος. Κατόπιν, χρησιμοποιούνται η οριζόντια τάση και η γραμμική τάση των παρατηρήσεων ώστε να βγει η πρόβλεψη. Ο υπολογισμός γίνεται σταδιακά ακολουθώντας τα παρακάτω βήματα:

1. Υπολογισμός απλού κινούμενου μέσου
2. Υπολογισμός του διπλού κινούμενου μέσου
3. Υπολογισμός του a_t
4. Υπολογισμός του συντελεστή τάσης (b_t)
5. Υπολογισμός της πρόβλεψης προσθέτοντας τις δύο προηγούμενες μεταβλητές a_t και b_t ($a_t + m \cdot b_t$)

$$F_{t+m} = a_t + b_t m$$

όπου

$$a_t = L_t + (L_t - {}_2L_t) = 2L_t - {}_2L_t$$

$$b_t = \frac{2}{k-1} (L_t - {}_2L_t)$$

Χάρη στη μέθοδο αυτή, μπορεί να πραγματοποιηθεί πρόβλεψη για πολλές μελλοντικές περιόδους. Συμβολίζοντας με $\tau > 1$ τις επόμενες μελλοντικές περιόδους, το τελευταίο βήμα μεταβάλλεται και ο τύπος παίρνει τη μορφή $F_{t+m} = a_t + m \cdot b_t$, όπου F_{t+m} η πρόβλεψη της εκάστοτε περιόδου.

Διπλή εκθετική εξομάλυνση μιας παραμέτρου: Σε αυτήν την περίπτωση υπολογίζεται ένας σταθμικός μέσος, και με βάση αυτόν ένας δεύτερος. Έπειτα λαμβάνονται υπόψη οι συντελεστές a , b (όπως στην εκθετική εξομάλυνση), με σκοπό τη δημιουργία της πρόβλεψης. Ο υπολογισμός γίνεται σταδιακά ακολουθώντας τα παρακάτω βήματα:

1. Εξομάλυνση με βάση το σταθμικό μέσο L_t
2. Εξομάλυνση εκ νέου με βάση το σταθμικό μέσο L_t
3. Υπολογισμός της σταθεράς εξομάλυνσης a_t , με $0 \leq a_t \leq 1$
4. Υπολογισμός του συντελεστή τάσης (b_t)
5. Υπολογισμός της πρόβλεψης ομοίως με την προηγούμενη μέθοδο.

$$F_{t+m} = a_t + b_t m$$

Όπου :

$$L_t = aY_t + (1-a)L_{t-1}$$

$${}_2L_t = aL_t + (1-a){}_2L_{t-1}$$

και

$$a_t = 2L_t - {}_2L_t$$

$$b_t = \frac{a}{1-a} (L_t - {}_2L_t)$$

Σε αυτή τη μέθοδο μπορούν επίσης να υπολογιστούν προβλέψεις από επόμενες μελλοντικές περιόδους. Συμβολίζοντάς τες με $t > 1$, το τελευταίο βήμα μεταβάλλεται και ο τύπος παίρνει τη μορφή $X_{t+\tau} = a_t + \tau \cdot b_t$, όπου $X_{t+\tau}$ η πρόβλεψη της εκάστοτε περιόδου.

Διπλή εκθετική εξομάλυνση δύο παραμέτρων: Σε αυτήν την περίπτωση χρησιμοποιούνται δύο διαφορετικές συναρτήσεις εξομάλυνσης. Η πρώτη (L_t) δείχνει την τιμή, ενώ η δεύτερη (b_t) την τάση. Για τον υπολογισμό τους, πρώτα ορίζονται δύο σταθερές εξομάλυνσης α , β , με $0 \leq \alpha \leq 1$ και $0 \leq \beta \leq 1$. Τα βήματα που ακολουθούνται είναι τα παρακάτω:

1. Υπολογισμός της συνάρτησης (εξομάλυνση) των τιμών.
2. Υπολογισμός της συνάρτησης (εξομάλυνση) της τάσης.
3. Υπολογισμός των μελλοντικών m προβλέψεων προσθέτοντας τις δύο εξισώσεις μεταβλητές L_t και b_t ($L_t + m \cdot b_t$)

$$F_{t+m} = L_t + b_t m$$

Όπου :

$$L_t = \alpha Y_t + (1 - \alpha)(L_{t-1} + b_{t-1})$$

$$b_t = \beta(L_t - L_{t-1}) + (1 - \beta)b_{t-1}$$

Ουσιαστικά, όταν γνωστοποιηθεί η τελευταία παρατήρηση, πραγματοποιείται ο υπολογισμός της νέας τιμής και με βάση τη διαφορά της με την προηγούμενη, υπολογίζεται ξανά η κλίση.

Συναρτήσει των δύο εξισώσεων υπολογίζεται η πρόβλεψη.

Τριπλή εκθετική εξομάλυνση: Η τριπλή εκθετική εξομάλυνση χρησιμοποιείται για παρατηρήσεις σε διαδοχικές μικρές περιόδους, μικρότερες ακόμη και από ένα έτος, που διακυμαίνονται εποχικά. Σε αυτήν τη περίπτωση ορίζονται τρεις διαφορετικές εξισώσεις εξομάλυνσης. Ομοίως με την προηγούμενη μέθοδο, οι πρώτες δύο υπολογίζουν την τιμή και την τάση, ενώ η τρίτη την εποχικότητα. Για τον υπολογισμό τους, ορίζονται επίσης τρεις σταθερές εξομάλυνσης, α , β , γ , με $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$ και $0 \leq \gamma \leq 1$. Για να υπολογιστεί η πρόβλεψη απαιτούνται τέσσερα βήματα:

1. Υπολογισμός της συνάρτησης (εξομάλυνση) των τιμών
2. Υπολογισμός της συνάρτησης (εξομάλυνση) της τάσης
3. Υπολογισμός (εξομάλυνση) της εποχικότητας
4. Υπολογισμός των μελλοντικών m προβλέψεων προσθέτοντας πρώτα τις δύο προηγούμενες εξισώσεις L_t και b_t και πολλαπλασιάζοντας με την τρίτη ($L_t + m \cdot b_t$) $\cdot S_{t+\tau-s}$

$$L_t = \alpha P_t + (1 - \alpha) Q_t$$

$$b_t = \beta R_t + (1 - \beta) b_{t-1}$$

$$S_t = \gamma T_t + (1 - \gamma) S_{t-s}$$

με τα P_t , Q_t , R_t , T_t , να διαμορφώνονται ανάλογα την περίπτωση ως εξής:

Συνιστώσα τάσης	Συνιστώσα εποχικότητας		
	1 Χωρίς επίδραση	2 Προσθετική επίδραση	3 Πολ/στική επίδραση
A Χωρίς επίδραση	$P_t = Y_t$ $Q_t = L_{t-1}$ $F_{t+m} = L_t$	$P_t = Y_t - S_{t-s}$ $Q_t = L_{t-1}$ $T_t = Y_t - L_t$ $F_{t+m} = L_t + S_{t+m-s}$	$P_t = Y_t / S_{t-s}$ $Q_t = L_{t-1}$ $T_t = Y_t / L_t$ $F_{t+m} = L_t S_{t+m-s}$
B Προσθετική επίδραση	$P_t = Y_t$ $Q_t = L_{t-1} + b_{t-1}$ $R_t = L_t - L_{t-1}$ $F_{t+m} = L_t + mb_t$	$P_t = Y_t - S_{t-s}$ $Q_t = L_{t-1} + b_{t-1}$ $R_t = L_t - L_{t-1}$ $T_t = Y_t - L_t$ $F_{t+m} = L_t + mb_t + S_{t+m-s}$	$P_t = Y_t / S_{t-s}$ $Q_t = L_{t-1} + b_{t-1}$ $R_t = L_t - L_{t-1}$ $T_t = Y_t / L_t$ $F_{t+m} = (L_t + mb_t) S_{t+m-s}$
Γ Πολ/στική επίδραση	$P_t = Y_t$ $Q_t = L_{t-1} b_{t-1}$ $R_t = L_t / L_{t-1}$ $F_{t+m} = L_t b_t^m$	$P_t = Y_t - S_{t-s}$ $Q_t = L_{t-1} b_{t-1}$ $R_t = L_t / L_{t-1}$ $T_t = Y_t - L_t$ $F_{t+m} = L_t b_t^m + S_{t+m-s}$	$P_t = Y_t / S_{t-s}$ $Q_t = L_{t-1} b_{t-1}$ $R_t = L_t / L_{t-1}$ $T_t = Y_t / L_t$ $F_{t+m} = L_t b_t^m S_{t+m-s}$

Υπάρχουν δύο διαφορετικά μοντέλα, τα οποία επιδρούν στο πώς προσδιορίζονται οι παρατηρήσεις μιας χρονοσειράς:

1. Προσθετική (additive): Η μεταβλητή μεταβάλλεται στο χρόνο κατά κάποιο σταθερό ποσό και με τον ίδιο τρόπο. Ονομάζεται προσθετική, διότι, ουσιαστικά, οι υπαρκτές συνιστώσες προστίθενται προκειμένου να υπολογιστεί η πρόβλεψη.
2. Πολλαπλασιαστική (multiplicative): Η μεταβλητή μεταβάλλεται στο χρόνο κατά κάποιο σταθερό ποσοστό. Ονομάζεται πολλαπλασιαστική, επειδή οι υπαρκτές συνιστώσες πολλαπλασιάζονται για τον υπολογισμό της πρόβλεψης. Το μοντέλο αυτό είναι ευκολότερο και, συνεπώς χρησιμοποιείται πιο συχνά.

Και στις δύο περιπτώσεις, με την επίδραση της εποχικής συνιστώσας, η τιμή της προβλεπόμενης μεταβλητής διαμορφώνεται σε διαφορετικά επίπεδα από τη μέση τιμή του έτους, κατά κάποιο σταθερό ποσό ή ποσοστό αντίστοιχα.

2.3.3.7 Άλλες μέθοδοι πρόβλεψης

Παλινδρόμηση: Παλινδρόμηση ονομάζεται μία συσχέτιση-συνάρτηση μεταξύ δύο ή περισσότερων μεταβλητών με σκοπό την πρόβλεψη των τιμών της μίας με τη χρησιμοποίηση της άλλης / των άλλων. Επομένως, όπως είναι προφανές υπάρχουν ανεξάρτητες και εξαρτημένες μεταβλητές. Συνήθως, οι όταν οι τιμές μιας μεταβλητής μπορούν να καθοριστούν με ευκολία, αυτή ορίζεται ως ανεξάρτητη, οι μεταβολές των τιμών της οποίας επηρεάζουν και συμβάλλουν στον υπολογισμό των τιμών της εξαρτημένης / των εξαρτημένων.

Όταν υπάρχει ένα ζεύγος μεταβλητών που μπορούν να απεικονιστούν γραμμικά τότε πρόκειται για απλή γραμμική παλινδρόμηση. Σε περίπτωση που οι παρατηρήσεις έχουν γίνει την ίδια χρονική στιγμή ή με κάποια χρονολογική σειρά, είναι προφανές πως πρόκειται για παλινδρόμηση με χρονοσειρές με ανεξάρτητη μεταβλητή το χρόνο

Έτσι, με τη βοήθεια της μεταβλητής a , υπολογίζεται η μεταβλητή b . Εάν μετά από γραφική αναπαράσταση των δεδομένων, παρατηρηθεί γραμμική σχέση τότε πρόκειται για γραμμική παλινδρόμηση με εξίσωση: $Y = a + bX$, όπου,

Y : εξαρτημένη μεταβλητή, X : ανεξάρτητη μεταβλητή, a : σταθερά, b : κλίση/συντελεστής διεύθυνσης. Καθώς, κατά τον υπολογισμό του Y , είναι πιθανό να παρατηρηθεί σφάλμα μέτρησης (e) είναι αναγκαίο να συμπεριληφθεί και αυτό ως όρος της

$$b = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2}$$

και

$$a = \bar{Y} - b\bar{X}$$

σχέσης, δηλαδή $Y = a + bX + e$.

Για τον υπολογισμό της σταθεράς b , ουσιαστικά, ελαχιστοποιείται το άθροισμα του τετραγώνου των αποκλίσεων των παρατηρήσεων.

Σε κάθε περίπτωση η σχέση ή μη δύο μεταβλητών (X και Y) μπορεί να γίνει γνωστή με τη χρήση του συντελεστή συσχέτισης (r_{XY} , με $-1 \leq r_{XY} \leq 1$):

$$r_{XY} = \frac{Cov_{XY}}{S_X S_Y} = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2} \sqrt{\sum (Y_i - \bar{Y})^2}}$$

Αν $0 < r_{XY} \leq 1$ τότε οι μεταβολές των μεταβλητών αυξάνονται ή μειώνονται ταυτόχρονα, με την εξάρτηση να εντείνεται όσο πλησιάζει ο συντελεστής στο 1.

Αν $-1 \leq r_{XY} < 0$ τότε οι μεταβολές των μεταβλητών αυξάνονται ή μειώνονται με αντίθετη φορά, με την εξάρτηση να εντείνεται όσο πλησιάζει ο συντελεστής στο -1.

Τέλος να ο συντελεστής είναι ίσος με το μηδέν, τότε ότι δεν υπάρχει εξάρτηση μεταξύ των μεταβλητών X , Y .

Σε περίπτωση μη γραμμικής παλινδρόμησης, ο συντελεστής δεν μπορεί να δείξει την εξάρτηση μεταξύ των δύο μεταβλητών.

Τέλος, ο συντελεστής προσδιορισμού δείχνει το ποσοστό του συνόλου του αθροίσματος των τετραγώνων των αποκλίσεων της εξαρτημένης μεταβλητής από τη μέση τιμή της, η οποία ερμηνεύεται από τη γραμμή παλινδρόμησης και την ανεξάρτητη μεταβλητή. Συμβολίζεται με R^2 και ορίζεται:

$$R^2 = r^2_{Y\hat{Y}} = \frac{\sum (\hat{Y}_i - \bar{Y})^2}{\sum (Y_i - \bar{Y})^2}$$

Ουσιαστικά ο συντελεστής προσδιορισμού ισούται με το τετράγωνο του συντελεστή συσχέτισης των μεταβλητών.

Η γραμμική παλινδρόμηση χρησιμοποιείται στις μακροπρόθεσμες προβλέψεις. Η γραμμική παλινδρόμηση χρησιμοποιείται, ωστόσο, και στις αιτιακές (ποσοτικές) σχέσεις. Η διαφορά συναντάται στο πώς μεταβάλλεται η μεταβλητή που μελετάται. Αν ευθύνεται αποκλειστικά μία δεύτερη μεταβλητή, τότε πρόκειται για αιτιακή σχέση. Αν υπαίτιος παράγοντας είναι ο χρόνος τότε πρόκειται για χρονοσειρά.

Εάν σε κάποια περίπτωση η εξαρτημένη μεταβλητή επηρεάζεται από k ανεξάρτητες μεταβλητές (δηλαδή περισσότερες από μία), η παλινδρόμηση ονομάζεται πολλαπλή

$$Y_i = b_0 + b_1 X_{1i} + b_2 X_{2i} + \dots + b_k X_{ki} + e$$

γραμμική παλινδρόμηση. Το σχεσιακό μοντέλο να αποκτά τη μορφή:

Όπου Y_i η εξαρτημένη μεταβλητή, b_i οι σταθερές παράμετροι, X_i οι ανεξάρτητες μεταβλητές και e το σφάλμα.

Η μέθοδος των Box - Jenkins:

Η μέθοδος ARIMA: Τα μοντέλα ARIMA (Autoregressive Integrated Moving Average) χρησιμοποιούνται για την πρόβλεψη των τιμών κάποιου μεγέθους και την εξέλιξη τους στο χρόνο. Αναπαριστούν χρονοσειρές ανεξαρτήτως στασιμότητας. Ωστόσο, δεν υπάρχουν ανεξάρτητες μεταβλητές, αλλά οι πληροφορίες των χρονοσειρών αξιοποιούνται για την πραγματοποίηση των προβλέψεων. Βασίζεται επίσης σε μοντέλα αυτοσυσχέτισης.

Αρχικά, ορίζεται ένα τυχαίο σφάλμα και αξιοποιούνται οι τιμές του μεγέθους που απασχολούν τον αναλυτή, ώστε να δημιουργήσουν οι δύο αυτοί παράγοντες μαζί μια γραμμική σχέση. Το μοντέλο αυτό φέρει τον τίτλο «μονοδιάστατο μοντέλο ARIMA».

Εάν συνυπάρξουν πολλοί παράγοντες, τότε το μοντέλο ARIMA γίνεται πολυδιάστατο και περιλαμβάνει χρονοσειρές ανεξάρτητων μεταβλητών.

Ένα μοντέλο ARIMA αποτελεί και η μέθοδος των Box - Jenkins. Έγινε δημοφιλής τη δεκαετία του 70 και πλέον χρησιμοποιείται από πολλές επιχειρήσεις. Μπορεί να ελέγχει την αποτελεσματικότητα του μοντέλου πρόβλεψης που επέλεξε, κάτι απολύτως χρήσιμο για τους αναλυτές των επιχειρήσεων.

Συνήθως, οι μέθοδοι πρόβλεψης επιλέγονται χωρίς να μπορεί να ελεγχθεί αν αρχικά το μοντέλο είναι το κατάλληλο προς υλοποίηση. Η μέθοδος των Box - Jenkins, όμως, συντονίζει προσεκτικά το μοντέλο και τη διαδικασία. Χρησιμοποιείται μια συστηματική προσέγγιση για τον προσδιορισμό ενός κατάλληλου μοντέλου. Με βάση τα ιστορικά δεδομένα ελέγχεται η εγκυρότητα. Παράγεται επίσης μια κατάλληλη διαδικασία πρόβλεψης.

Πρόκειται για μια επαναληπτική μέθοδο. Δε χρειάζεται κάποιο υπάρχον μοτίβο ιστορικών δεδομένων. Με βάση ένα μοτίβο αυτοσυσχέτισης, επιλέγεται ο καταλληλότερος τρόπος πρόβλεψης των δεδομένων. Μια αυτοσυσχέτιση μετρά τη συσχέτιση μεταξύ των τιμών της χρονοσειράς που διαχωρίζονται από περιόδους. Κατόπιν χρειάζεται να εκτιμηθούν οι παράμετροι που σχετίζονται με το υπόδειγμα αξιοποιώντας τα ιστορικά δεδομένα. Τέλος, υπολογίζονται τα σφάλματα πρόβλεψης. Εάν τόσο τα σφάλματα και οι εκτιμήσεις συμπεριφέρονται όπως αναμένεται βάσει του υποτιθέμενου προτύπου, το πρότυπο είναι σωστό και διατηρείται. Εάν όχι, επιβάλλεται να αλλάξει και να επαναληφθεί η διαδικασία μέχρι να επικυρωθεί ένα πρότυπο. Σε αυτό το σημείο, εμφανίζεται μια πραγματική πρόβλεψη για την επόμενη περίοδο.

Ωστόσο, για να επιτευχθούν όλα αυτά, η μέθοδος Box-Jenkins απαιτεί μεγάλο όγκο παρελθοντικών δεδομένων (τουλάχιστον 50 χρονικές περιόδους), γι' αυτό και χρησιμοποιείται μόνο για μεγάλες εφαρμογές.

2.4 Η γλώσσα Python

Η γλώσσα Python είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου που λειτουργεί με διερμηνευτή (interpreter). Το υψηλό επίπεδο των δομών δεδομένων που βρίσκονται ενσωματωμένες, μαζί με τη δυναμική τυποποίηση και τη δυναμική δέσμευση, την καθιστούν κατάλληλη για τη γρήγορη ανάπτυξη εφαρμογών, καθώς και για χρήση ως γλώσσα scripting ή για σύνδεση υφιστάμενων στοιχείων μεταξύ τους. Δημιουργήθηκε το 1989 στο ερευνητικό κέντρο Centrum Wiskunde & Informatica (CWI) από τον Ολλανδό Guido van Rossum, αρχικά στα συστήματα Amoeba. Δύναται να χρησιμοποιηθεί για αντικειμενοστραφή και για διαδικαστικό προγραμματισμό, ενώ πρόκειται για μια ανοιχτού κώδικα γλώσσα (open-source).

Η Python 2.0 κυκλοφόρησε το 2000, ενώ το Δεκέμβριο του 2008 κυκλοφόρησε η έκδοση 3.0. Πρόκειται για την πρώτη γλώσσα (έκδοση) προγραμματισμού που είναι ασύμβατη με εκδόσεις του παρελθόντος, ενώ υπάρχει κατάλληλο πρόγραμμα που επιτρέπει την μετατροπή από 2.0 σε 3.0. Ο λόγος είναι πως η νέα έκδοση είναι ασύμβατη με τις παλαιότερες. Αυτό συμβαίνει, διότι η έκδοση 3.0 περιέχει πολλές διορθώσεις λαθών που υπήρχαν/υπάρχουν στις προηγούμενες. Σήμερα, η τελευταία υπάρχουσα έκδοση είναι η 3.10.5.

Χαρακτηριστικό της γλώσσας είναι η χρήση αλλαγής σειράς και εσοχών για τη δημιουργία εμφωλευμένων εντολών. Οι περισσότερες γλώσσες προγραμματισμού χρησιμοποιούν τα άγκιστρα ({, }) για το συγκεκριμένο σκοπό. Το απλό, εύχρηστο συντακτικό της Python συμβάλλει στην εύκολη αναγνωσιμότητά της. Ως αποτέλεσμα, το κόστος συντήρησης των προγραμμάτων είναι μικρό.

Εφόσον δεν υπάρχει βήμα μεταγλώττισης, η εκσφαλμάτωση των προγραμμάτων εκσφαλμάτωση γίνεται ταχεία. Όταν μάλιστα ο διερμηνέας βρει κάποιο σφάλμα, δημιουργεί εξαίρεση (exception). Αν το πρόγραμμα αδυνατήσει να πιάσει την εξαίρεση (exception catch), ο διερμηνέας εκτυπώνει ένα ίχνος στοίβας. Ένας αποσφραματωτής

(γραμμένος επίσης σε Python) σε επίπεδο πηγής επιτρέπει την επιθεώρηση τοπικών και παγκόσμιων μεταβλητών, αξιολογεί τις αυθαίρετες εκφράσεις, ενώ μπορεί, επιπλέον, να εκτελεί τον κώδικα σειρά προς σειρά.

Η συγκεκριμένη γλώσσα υποστηρίζει πακέτα, κάτι που επηρεάζει θετικά τη σχεδίαση του προγράμματος και την επαναχρησιμοποίηση του κώδικα. Ο διερμηνευτής της Python, ο οποίος είναι απαραίτητος για την εκτέλεση προγραμμάτων Python, αλλά και η εκτεταμένη πρότυπη βιβλιοθήκη είναι διαθέσιμα σε μορφή πηγαίου ή δυαδικού κώδικα για όλες τις κύριες πλατφόρμες, ενώ διανέμονται ελεύθερα. Για τη δημιουργία του κώδικα χρησιμοποιείται ένας κειμενογράφος ή ακόμα κάποιο περιβάλλον ανάπτυξης (Integrated Development Environment - IDE) ειδικά φτιαγμένο για τη γλώσσα. Η Python έρχεται μαζί με το δικό της περιβάλλον ανάπτυξης (IDLE).

3. Η εφαρμογή

3.1 Ανάπτυξη της εφαρμογής

Σκοπός της εφαρμογής είναι να δίνει ο χρήστης τα δεδομένα και αφού επιλέξει ποια μέθοδο ανάλυσης χρονοσειρών επιθυμεί να γίνεται η πρόβλεψη για την επόμενη περίοδο.

Η εφαρμογή ξεκινάει και δίνει στην επιλογή στον χρήστη να επιλέξει πως επιθυμεί να εισάγει τα δεδομένα στο πρόγραμμα. Μπορεί είτε μέσω αρχείου txt είτε χειροκίνητα να εισάγει τις τιμές των δεδομένων(παρατηρήσεις). Στην πρώτη περίπτωση ο χρήστης θα πρέπει να δημιουργήσει ένα αρχείο txt όπου θα εισάγει της παρατηρήσεις. Θα πρέπει να έχει ένα txt για κάθε περίπτωση που θέλει να εξετάσει και τα δεδομένα να βρίσκονται σε μία σειρά και να χωρίζονται με ένα κόμμα(,) μεταξύ τους.

Αφού ο χρήστης επιλέξει, στη συνέχεια το πρόγραμμα εμφανίζει μια γραφική παράσταση με τα δεδομένα και του δίνεται η επιλογή να διαλέξει ποια μέθοδο θέλει να χρησιμοποιήσει.

Αφού διαλέξει, το πρόγραμμα αξιοποιεί όλα αυτά τα δεδομένα για να υπολογίσει την πρόβλεψη (Ft), το λάθος et όπως επίσης το Μέσο σφάλμα, το Μέσο απόλυτο σφάλμα και το Μέσο τετράγωνο σφάλματος. Στο τέλος εμφανίζει έναν πίνακα με όλα τα δεδομένα, τόσο τα ήδη γνωστά, όσο και τα υπολογισμένα και εμφανίζει γραφικά την καμπύλη των παρατηρήσεων και των προβλέψεων.

3.2 Εισαγωγή δεδομένων και έλεγχος εγκυρότητας

Ο κώδικας ξεκινάει με την εισαγωγή τεσσάρων ενοτήτων (modules).

```
import math
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

1. Math: Αυτή η ενότητα παρέχει πρόσβαση στις μαθηματικές συναρτήσεις που ορίζονται από το πρότυπο C και χρησιμοποιείται με πραγματικούς αριθμούς. Όλες οι συναρτήσεις αυτής της ενότητας επιστρέφουν πραγματικό αριθμό 32 ψηφίων (bits), δηλαδή float. Στο συγκεκριμένο πρόγραμμα έχουν χρησιμοποιηθεί οι συναρτήσεις round, η οποία στρογγυλοποιεί έναν αριθμό, pow, η οποία υψώνει έναν αριθμό σε δύναμη και η abs, η οποία υπολογίζει την απόλυτη τιμή ενός αριθμού.

2. Pandas: Πρόκειται για ένα πακέτο Python ανοικτού κώδικα που χρησιμοποιείται συνήθως για την ανάλυση δεδομένων και μηχανική μάθηση. Βασίστηκε και δημιουργήθηκε χτισμένο πάνω σε ένα άλλο πακέτο, το NumPy, το οποίο παρέχει υποστήριξη για πολυδιάστατους πίνακες. Το Pandas συνεργάζεται καλά με πολλές άλλες ενότητες επιστήμης δεδομένων μέσα σε ένα πρόγραμμα της Python και περιλαμβάνεται σε κάθε διανομή Python. Στο συγκεκριμένο πρόγραμμα, χρησιμοποιείται κυρίως για, οπτικοποίηση, φόρτωση και αποθήκευση δεδομένων. Έχει χρησιμοποιηθεί η συνάρτηση DataFrame για την εμφάνιση των δεδομένων σε μορφή πίνακα.
3. Matplotlib.pyplot: Είναι μια συλλογή συναρτήσεων που «φέρνουν» το MATLAB στο περιβάλλον Python. Κάθε συνάρτηση pyplot σχεδιάζει ή μεταβάλλει ένα σχήμα, όπως μια γραφική παράσταση. Χρησιμοποιείται για τη γραμμική απεικόνιση των παρατηρήσεων και των προβλέψεων.
4. NumPy: Πρόκειται για μια βιβλιοθήκη Python γραμμένη κυρίως σε C++. Χρησιμοποιείται για εργασία με πίνακες, ενώ αποτελεί χρήσιμο εργαλείο και στη γραμμική άλγεβρα. Είναι ανοικτού κώδικα και διατίθεται δωρεάν. Επειδή οι λίστες της Python είναι πιο αργές, το NumPy δημιουργεί ένα αντικείμενο πίνακα. Η ενότητα numpy είναι απαραίτητη για τη χρήση του Pandas, που έχει χρησιμοποιηθεί στο συγκεκριμένο πρόγραμμα.

Έπειτα, ορίζονται κάποιες συναρτήσεις, οι οποίες θα φανούν χρήσιμες στη συνέχεια του κώδικα.

1. Η συνάρτηση calce (calculate e) υπολογίζει το σφάλμα πρόβλεψης. Δέχεται ως είσοδο τη λίστα με τις παρατηρήσεις (y), τη λίστα με τις προβλέψεις (ma), το πλήθος των περιόδων (n), καθώς πόσα σφάλματα θέλει ο χρήστης να

υπολογίσει. Έπειτα, αρχικοποιεί τη λίστα των σφαλμάτων πρόβλεψης, τοποθετώντας στην πρώτη θέση το NaN. Αυτό συμβαίνει διότι στην πρώτη περίοδο δεν υπάρχει πρόβλεψη και, επομένως, ούτε και σφάλμα. Σε κάθε περίοδο, αν δεν υπάρχει πρόβλεψη τοποθετεί στη θέση της λίστας e το NaN, διαφορετικά υπολογίζει το σφάλμα με τον κατάλληλο τύπο και στρογγυλοποιεί την τιμή στα τρία δεκαδικά ψηφία. Τέλος, για όσες περιόδους δεν επιθυμούνται προβλέψεις, τοποθετεί το NaN στις αντίστοιχες θέσεις της λίστας και επιστρέφει ολόκληρη τη λίστα.

```
def calce(y, ma, n, sfalmata):  
    e = [np.NaN]  
    for i in range(1, n):  
        if math.isnan(ma[i]):  
            e.append(np.NaN)  
        else:  
            a = y[i] - ma[i]  
            e.append(round(a, 3))  
    e.append(np.NaN)  
    for i in range(1, sfalmata):  
        e[i] = np.NaN  
    return e
```

2. Η συνάρτηση calcm (calculate m) υπολογίζει το μέσο σφάλμα πρόβλεψης, το μέσο απόλυτο σφάλμα και το μέσο σφάλμα τετραγώνου. Αρχικά, εκχωρείται η τιμή 0 στον μετρητή (sma). Στη συνέχεια εκτελείται μια for όπου γίνεται έλεγχος για κάθε περίοδο και όπου δεν έχει υπολογιστεί κάποιο σφάλμα ο μετρητής αυξάνεται κατά μια μονάδα. Υπάρχει επίσης και μήνυμα αν τα δεδομένα είναι λιγότερα από τα ζητούμενα σφάλματα. Έπειτα εκχωρούνται στις μεταβλητές s(απλός μετρητής), me(μέσο σφάλμα), mae(μέσο απόλυτο σφάλμα) και mse(μέσο τετράγωνο σφάλματος) η τιμή 0 και με τους κατάλληλους τύπους υπολογίζεται η κάθε μεταβλητή και εκχωρείται η τιμή της στρογγυλοποιημένη

στα τρία δεκαδικά ψηφία. Τελικά, δημιουργείται η λίστα m, σε κάθε μία θέση της οποίας υπάρχει η τιμή του κάθε σφάλματος. Επιστρέφεται η λίστα m.

```
def calcm(e, n):
    snan=0
    for i in range(1, len(e)):
        if math.isnan(e[i]):
            snan=snan+1
    if snan == len(e):
        print("Λίγα δεδομένα για σφάλματα")
        return 0
    else:
        s = 0
        me = 0
        mae = 0
        mse = 0
        for i in range(n):
            if not math.isnan(e[i]):
                s += 1
                me = round((e[i]+me), 3)
                mae = round((abs(e[i])+mae), 3)
                mse = round((pow(e[i], 2)+mse), 3)
        m = [me / s, mae / s, mse / s]
    return m
```

Οι συναρτήσεις που εμφανίζονται στη συνέχεια, πραγματοποιούν τις προβλέψεις με βάση την κάθε μέθοδο (μία συνάρτηση για μία διαφορετική μέθοδο).

1. Η συνάρτηση `series1` λειτουργεί με βάση το μοντέλο του απλού κινούμενου μέσου. Παίρνει ως ορίσματα από τον χρήστη δύο λίστες, μια λίστα περιέχει τον αριθμό των περιόδων και μια δεύτερη που περιέχει τις παρατηρήσεις. Στη συνέχεια προστίθεται μία επιπλέον σειρά στο τέλος της λίστας της περιόδου με την επόμενη περίοδο και στο τέλος της λίστας με τις παρατηρήσεις μια κενή σειρά όπου θα είναι η επόμενη πρόβλεψη. Έπειτα ζητείται από τον χρήστη να εισάγει τον αριθμό k που επιθυμεί. Κατόπιν, δημιουργείται το `DataFrame`. Αμέσως μετά τη δημιουργία του, η αρίθμηση μετακινείται μια θέση μπροστά. Αυτό συμβαίνει διότι το μέτρημα στην Python ξεκινά από το 0, αλλά στον πραγματικό κόσμο δεν υπάρχει περίοδος 0. Έπειτα υπολογίζεται ο κινούμενος μέσος. Η βιβλιοθήκη `pandas` διαθέτει ήδη συνάρτηση εύρεσης του κινητού μέσου, επομένως δε χρειάζονται περαιτέρω υπολογισμοί. Ωστόσο χρειάζεται ολίσθηση (`shift`) μία θέση μπροστά, έτσι ώστε η πρόβλεψη να εμφανίζεται δίπλα στην περίοδο, στην οποία αναφέρεται και όχι δίπλα σε αυτήν, στην οποία υπολογίζεται. Μετά από αυτή τη διαδικασία ζητείται και πάλι από τον χρήστη να εισάγει τον αριθμό των σφαλμάτων που επιθυμεί να παραλείψει ώστε να υπολογιστούν τα υπόλοιπα σφάλματα. Στη συνέχεια, κάθε τιμή αποθηκεύεται σε μία θέση της λίστας `ma`, για να μπορούν να πραγματοποιηθούν οι υπολογισμοί. Έτσι, καλείται η `calce`, στην οποία αποστέλλονται οι λίστες των παρατηρήσεων και των προβλέψεων, καθώς και ο αριθμός των περιόδων και τα σφάλματα προς παράληψη. Κατόπιν καλείται η `calcm` όπου υπολογίζονται και επιστρέφει τις υπόλοιπες μεταβλητές (Μέσο Σφάλμα, Μέσο Απόλυτο Σφάλμα, Μέσο Σφάλμα Τετραγώνου) και εμφανίζονται κάτω από τα δεδομένα. Τέλος εμφανίζεται το γράφημα, δηλαδή απεικονίζονται οι παρατηρήσεις και οι προβλέψεις και ο χρήστης έχει την επιλογή να αποθηκεύσει τα αποτελέσματα σε αρχείο `excel`.

```

def series1(weeks,y):
    y.append(np.NaN)
    weeks.append(len(weeks)+1)
    print("Δώσε το K\n")
    k=int(input())
    df = pd.DataFrame({'t': weeks, 'yt': y})
    df.index = np.arange(1, len(df) + 1)
    df['Ft'] = round(df['yt'].rolling(window=k).mean(),2)
    df['Ft'] = df['Ft'].shift(1)
    ma = df['Ft'].tolist()
    sfalmata=int(input("Αριθμός σφαλμάτων για παράβλεψη"))
    e = calce(y, ma, len(weeks)-1,sfalmata)
    df['et'] = e
    print(df)
    m = calcm(e, len(y))
    if m != 0:
        print("Μέσο σφάλμα ", round(m[0], 2), "\nΜέσο απόλυτο σφάλμα ", round(m[1], 2),
              "\nΜέσο τετράγωνο σφάλματος ", round(m[2], 2))
    df.plot.line(x='t', y=['yt', 'Ft'])
    pl.show()
    save = str(input("Αποθήκευση αποτελέσματος? Y/N\n"))
    if save == "Y":
        name = str(input("Όνομα αρχείου?\n"))
        df.to_csv(name + '.csv', encoding='utf-8')

```

Διάλεξε χρονοσειρά

1. Απλός κινούμενος μέσων
2. Απλή εκθετική εξομάλυνση
3. Μέθοδος Holt δυο παραμέτρων
4. Μέθοδος Holt-Winter's για τάση και εποχικότητα

3

1

Δώσε το K

3

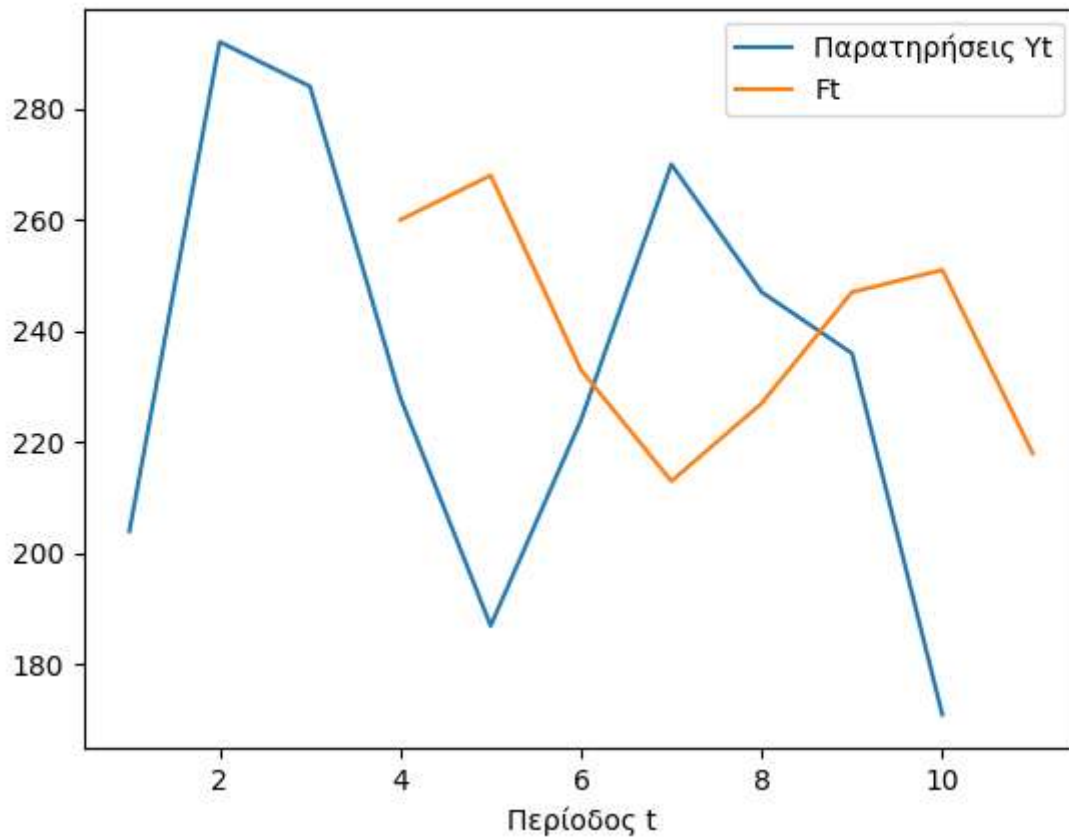
Αριθμός σφαλμάτων για παράβλεψη

Περίοδος t	Παρατηρήσεις Yt	Ft	et
1	204.0	NaN	NaN
2	292.0	NaN	NaN
3	284.0	NaN	NaN
4	228.0	260.0	NaN
5	187.0	268.0	NaN
6	224.0	233.0	-9.0
7	270.0	213.0	57.0
8	247.0	227.0	20.0
9	236.0	247.0	-11.0
10	171.0	251.0	-80.0
11	NaN	218.0	NaN

Μέσο σφάλμα -4.6

Μέσο απόλυτο σφάλμα 35.4

Μέσο τετράγωνο σφάλματος 2050.2



2. Η συνάρτηση `series2` λειτουργεί με βάση το μοντέλο της απλής εκθετικής εξομάλυνσης. Ομοίως με την προηγούμενη συνάρτηση, παίρνει σαν ορίσματα τις περιόδους και τις παρατηρήσεις που τις εισάγει ο χρήστης, όπως επίσης ζητάει και τον αριθμό των σφαλμάτων που θέλει ο χρήστης να παραλείψει ώστε να υπολογιστούν τα σφάλματα των υπόλοιπων περιόδων. Στη συνέχεια καλείται η συνάρτηση `Yalrha` που δέχεται ως ορίσματα τις περιόδους, τις παρατηρήσεις και τα σφάλματα προς παράληψη. Η συγκεκριμένη συνάρτηση μας επιστρέφει την κατάλληλη τιμή της σταθεράς εξομάλυνσης α που ελαχιστοποιεί το μέσο τετράγωνο σφάλματος. Έπειτα δημιουργείται η λίστα `nexty1` όπου στην πρώτη σειρά εισάγεται κενό και στη δεύτερη η τιμή της πρώτης περιόδου και από την τρίτη σειρά και μετά εισάγεται η τιμή της πρόβλεψης της επόμενης περιόδου που υπολογίζεται

χρησιμοποιώντας το α από την συνάρτηση γ alrha. Το αποτέλεσμα των προβλέψεων που βγαίνουν από τους υπολογισμούς περιέχει μια απόκλιση 0,1% από τα πραγματικά αποτελέσματα για δύο λόγους, ένας είναι ότι έχουν επιλεγεί μέχρι και δύο δεκαδικά ψηφία για κάθε αριθμό και ο δεύτερος είναι ότι η συνάρτηση γ alrha λόγω του μεγάλου όγκου πράξεων και επαναλήψεων επιλέχθηκε να κάνει την αναζήτηση του αριθμού α σε ένα μικρότερο εύρος αριθμών. Αφού γίνει ο υπολογισμός για κάθε παρατήρηση καλείται η συνάρτηση calce για τον υπολογισμό των σφαλμάτων. Η επιστροφή των σφαλμάτων, μαζί με τα τους υπολογισμούς και τις περιόδους εισάγονται στο Dataframe που έχει δημιουργηθεί. Τέλος εμφανίζεται το γράφημα, δηλαδή απεικονίζονται οι παρατηρήσεις και οι προβλέψεις και ο χρήστης έχει την επιλογή να αποθηκεύσει τα αποτελέσματα σε αρχείο excel.

```
def series2(weeks, y):
    y.append(np.NaN)
    weeks.append(len(weeks) + 1)
    sfalmata=int(input("Αριθμός σφαλμάτων για παράβλεψη"))
    alpha=Yalrha(weeks, y, sfalmata)
    nexty1 = [np.NaN, y[0]]
    print("Απλή εκθετική εξομάλυνση με  $\alpha =$  ", alpha, "\n")
    for i in range(1, len(weeks)-1):
        nexty1.append(round(nexty1[i] + alpha * (y[i] - nexty1[i]), 2))
        nexty1[i] = float("{:.2f}".format(round(nexty1[i], 2)))
    e = calce(y, nexty1, len(weeks)-1, sfalmata)
    df = pd.DataFrame({'t': weeks, 'yt': y, 'Ft': nexty1, 'et': e})
    pd.set_option('display.max_columns', 15)
    df.index = np.arange(1, len(df) + 1)
    print(df)
    m = calcm(e, len(y))
    print("Μέσο σφάλμα ", round(m[0], 2), "\nΜέσο απόλυτο σφάλμα ", round(m[1], 2),
          "\nΜέσο τετράγωνο σφάλματος ", round(m[2], 2))
    df.plot.line(x='t', y=['yt', 'Ft'])
    pl.show()
    save = str(input("Αποθήκευση αποτελέσματος? Y/N\n"))
    if save == "Y":
        name = str(input("Όνομα αρχείου?\n"))
        df.to_csv(name + '.csv', encoding='utf-8')
```

Διάλεξε χρονοσειρά

1. Απλός κινούμενος μέσων
2. Απλή εκθετική εξομάλυνση
3. Μέθοδος Holt δυο παραμέτρων
4. Μέθοδος Holt-Winter's για τάση και εποχικότητα

2

2

Αριθμός σφαλμάτων για παράβλεψη 3

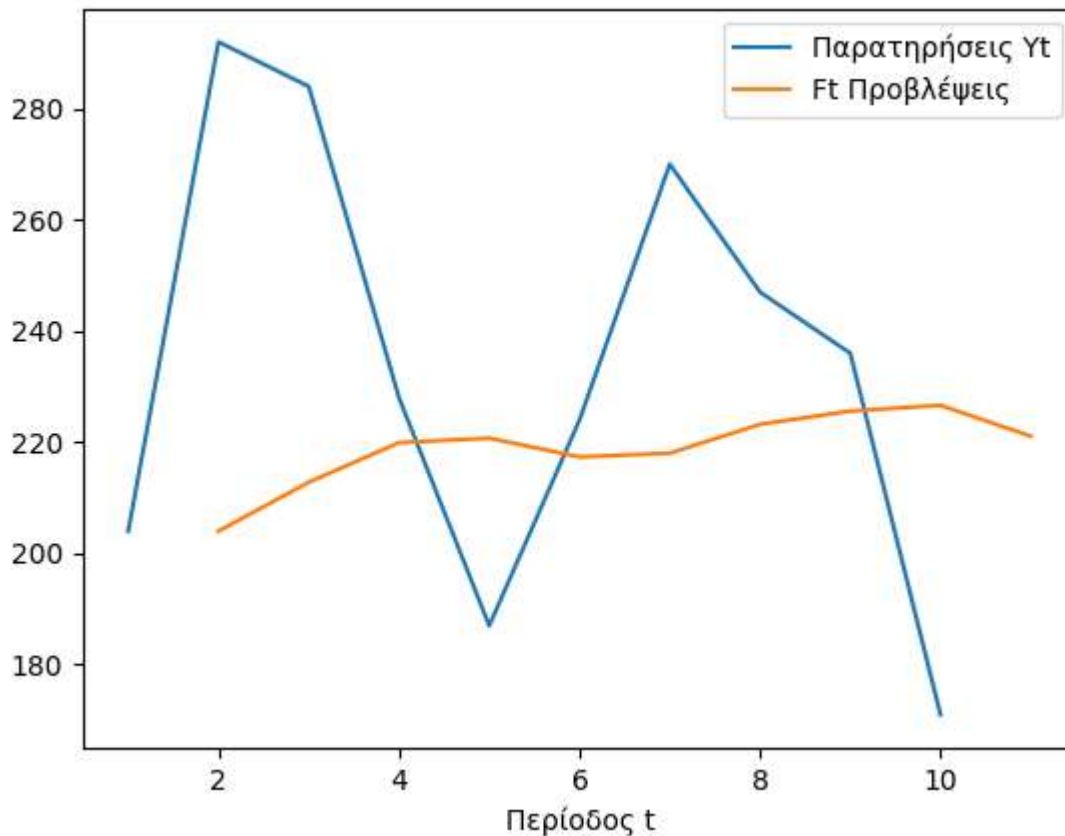
Απλή εκθετική εξομάλυνση με $\alpha = 0.1$

	Περίοδος t	Παρατηρήσεις Yt	Ft	Προβλέψεις	et
1	1.0	204.0		NaN	NaN
2	2.0	292.0		204.00	NaN
3	3.0	284.0		212.80	NaN
4	4.0	228.0		219.92	8.08
5	5.0	187.0		220.73	-33.73
6	6.0	224.0		217.36	6.64
7	7.0	270.0		218.02	51.98
8	8.0	247.0		223.22	23.78
9	9.0	236.0		225.60	10.40
10	10.0	171.0		226.64	-55.64
11	11.0	NaN		221.08	NaN

Μέσο σφάλμα 1.64

Μέσο απόλυτο σφάλμα 27.18

Μέσο τετράγωνο σφάλματος 1102.64



3. Η συνάρτηση `series3` λειτουργεί με βάση το μοντέλο της διπλής εκθετικής εξομάλυνσης δύο παραμέτρων που είναι γνωστή ως μέθοδος Holt (Holt's two parameter linear method). Όπως και στις προηγούμενες συναρτήσεις έτσι και εδώ η `series3` δέχεται ως ορίσματα τις περιόδους και τις παρατηρήσεις που εισάγει ο χρήστης και δημιουργεί αντίστοιχα δύο λίστες: τη `weeks` για τις περιόδους και τη `y` για τις παρατηρήσεις. Έπειτα δημιουργείται ένα `DataFrame` όπου και εδώ η αρίθμηση θα μετακινηθεί μια θέση μπροστά. Στη συνέχεια ζητείται από τον χρήστη να εισάγει τον αριθμό των σφαλμάτων που θέλει να παραλείψει, ώστε να υπολογιστούν τα σφάλματα των υπόλοιπων περιόδων και έπειτα αρχικοποιούνται οι εξής λίστες: η `b` όπου αρχικοποιείται με το μέσο βήμα της τάσης στις κοντινές

για την έναρξη της μεθόδου παρατηρήσεις και θα περιέχει τα δεδομένα για την εκτίμηση του βήματος της τάσης από περίοδο σε περίοδο, η lt όπου αρχικοποιείται με την τιμή της πρώτης παρατήρησης. Μετά καλείται η συνάρτηση $YAlphaBeta$ όπου παίρνει σαν ορίσματα τις περιόδους, τις παρατηρήσεις, τον αριθμό των σφαλμάτων καθώς και τις λίστες b και lt . Η συγκεκριμένη συνάρτηση επιστρέφει μία λίστα που περιέχει δύο παραμέτρους, το a και το β . Αυτές οι δύο παράμετροι ονομάζονται σταθερές εκθετικής εξομάλυνσης και η συνάρτηση $YAlphaBeta$ υπολογίζει την κατάλληλη τιμή τους ώστε το μέσο τετράγωνο σφάλματος των προβλέψεων να γίνει το ελάχιστο. Αφού επιστραφούν αυτές οι παράμετροι μπορούν να χρησιμοποιηθούν στη συνέχεια του προγράμματος όπου υπολογίζονται τα lt και b με βάση τους τύπους τους. Μόλις υπολογιστούν, δημιουργείται η λίστα f που θα αποθηκεύονται οι προβλέψεις και στις επόμενες δύο σειρές είναι ο υπολογισμός των προβλέψεων αυτών. Έπειτα καλείται η $calc$ για τα σφάλματα και οι τιμές που επιστρέφονται εισάγονται στο `DataFrame`. Στη συνέχεια καλείται η $calc_m$ όπου υπολογίζει και επιστρέφει το Μέσο Σφάλμα, το Μέσο Απόλυτο Σφάλμα και το Μέσο Τετράγωνο Σφάλματος. Τέλος η συνάρτηση εμφανίζει τα δεδομένα από το `DataFrame` δηλαδή τις περιόδους, τις παρατηρήσεις, τις προβλέψεις, τα lt και b και τα σφάλματα σε κάθε περίοδο και κατόπιν το γράφημα.

```

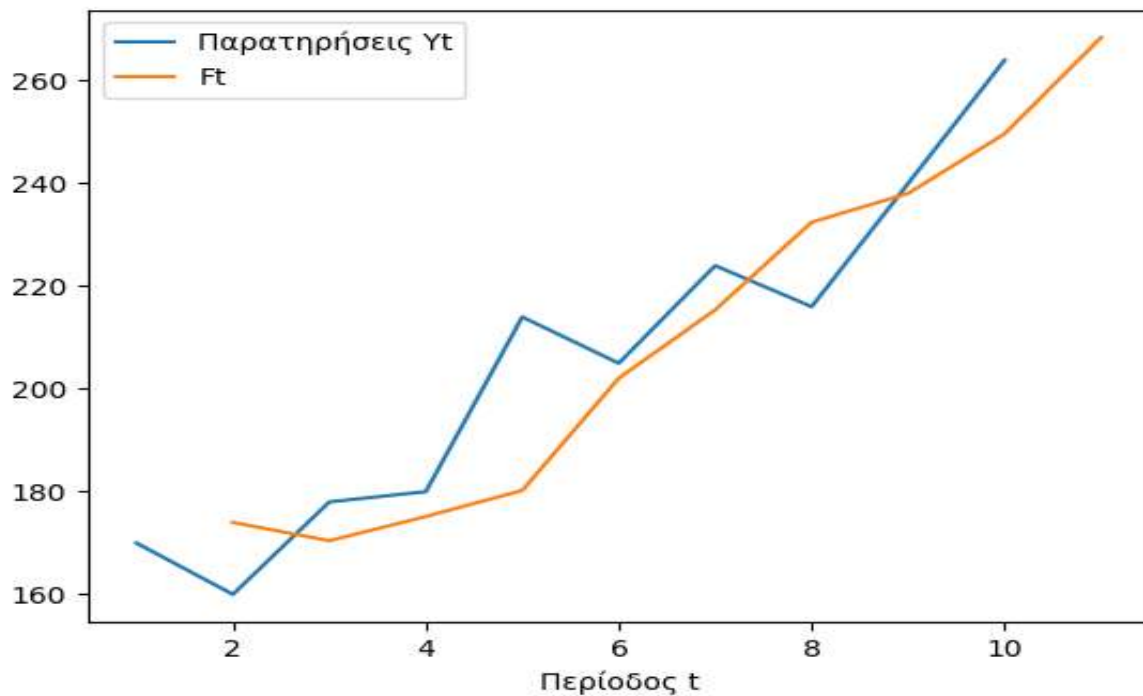
def series3(weeks, y):
    weeks.append(len(weeks)+1)
    y.append(np.NaN)
    pd.set_option('display.max_columns', 10)
    df = pd.DataFrame({'t': weeks, 'yt': y})
    sfalmata=int(input("Αριθμός σφαλμάτων για παράβλεψη"))
    df.index = np.arange(1, len(df) + 1)
    b=[(y[2]-y[0])/2]
    lt = [y[0]]
    Yp=YAlphaBeta(weeks, y, sfalmata, lt[0], b[0])
    print("Μέθοδος Holt δύο παραμέτρων με α=", Yp[0], "β=", Yp[1], "\n")
    for i in range(1, len(y)-1):
        lt.append(Yp[0]*y[i] + (1-Yp[0])*(lt[i-1]+b[i-1]))
        lt[i] = round(lt[i], 3)
        b.append(Yp[1]*(lt[i] - lt[i - 1]) + (1 - Yp[1])*b[i - 1])
        b[i] = round(b[i], 3)
    b.append(np.NaN)
    lt.append(np.NaN)
    df['Lt'] = lt
    f = [np.NaN]
    for i in range(1, len(y)):
        f.append(lt[i-1] + b[i-1])
    df['bt'] = b
    df['Ft'] = f
    e = calce(y, f, len(y)-1, sfalmata)
    df['et'] = e
    print(df)
    m = calcm(e, len(y))
    print("Μέσο σφάλμα ", round(m[0], 3), "\nΜέσο απόλυτο σφάλμα ", round(m[1], 3),
          "\nΜέσο τετράγωνο σφάλματος ", round(m[2], 3))
    df.plot.line(x='t', y=['yt', 'Ft'])
    pl.show()
    save = str(input("Αποθήκευση αποτελέσματος? Y/N\n"))
    if save == "Y":
        name = str(input("Όνομα αρχείου?\n"))
        df.to_csv(name + '.csv', encoding='utf-8')

```

Διάλεξε χρονοσειρά
 1. Απλός κινούμενος μέσος
 2. Απλή εκθετική εξομάλυνση
 3. Μέθοδος Holt δυο παραμέτρων
 4. Μέθοδος Holt-Winter's για τάση και εποχικότητα
 5
 3
 Αριθμός σφαλμάτων για παράβλεψη 5
 Μέθοδος Holt δυο παραμέτρων με $\alpha = 0.3$ $\beta = 0.8$

	t	yt	Lt	bt	Ft	et
1	1	170.0	170.000	4.000	NaN	NaN
2	2	160.0	169.800	0.640	174.000	NaN
3	3	178.0	172.708	2.454	170.440	NaN
4	4	180.0	176.613	3.615	175.162	NaN
5	5	214.0	190.360	11.721	180.228	NaN
6	6	205.0	202.957	12.422	202.081	2.919
7	7	224.0	217.965	14.491	215.379	8.621
8	8	216.0	227.519	10.541	232.456	-16.456
9	9	240.0	238.642	11.007	238.060	1.940
10	10	264.0	253.954	14.451	249.649	14.351
11	11	NaN	NaN	NaN	268.405	NaN

Μέσο σφάλμα 2.275
 Μέσο απόλυτο σφάλμα 8.857
 Μέσο τετράγωνο σφάλματος 112.672



4. Η συνάρτηση `series4` λειτουργεί με βάση το μοντέλο της Εκθετικής εξομάλυνσης για γραμμική τάση και εποχικότητα (Holt-Winter's trend and seasonality method). Η συνάρτηση καλείται με ορίσματα δύο λίστες, τη `weeks` και την `y` όπως σε κάθε συνάρτηση. Αρχικά ο χρήστης εισάγει τιμή για τη μεταβλητή `es`, που είναι εύρος της εποχικότητας που επιθυμεί να υπολογίσει. Στη συνέχεια δημιουργείται το `DataFrame` που θα περιέχει τα δεδομένα όπως οι περίοδοι, οι παρατηρήσεις, οι προβλέψεις και τα σφάλματα. Έπειτα ζητείται από τον χρήστη να εισάγει τα σφάλματα που θέλει να παραληφθούν ώστε να υπολογιστούν τα υπόλοιπα και μετά δημιουργούνται οι λίστες για τα `lt`, `b`, `f`, και `s`. Μετά γίνεται η αρχικοποίηση των συγκεκριμένων λιστών με τις κατάλληλες τιμές. Όσο είναι το `es` (εύρος εποχικότητας) τόσες θα είναι και οι κενές πρώτες σειρές των λιστών `lt` και `b`. Μετά το πέρας των `es` σειρών οι λίστες παίρνουν τις ανάλογες τιμές αρχικοποίησης. Όλη αυτή η διαδικασία πραγματοποιείται μέσα σε ένα βρόγχο(`for loop`) που επαναλαμβάνεται όσες φορές όσο είναι το `es`. Αμέσως μετά καλείται η συνάρτηση `YAlphaBetaGama` που παίρνει ως ορίσματα τις περιόδους, τις παρατηρήσεις, τα σφάλματα, το `lt`, το `b`, το `s` και το `es`. Η συνάρτηση αυτή επιστρέφει μια λίστα με τρεις παραμέτρους α , β και γ όπου ονομάζονται σταθερές εκθετικής εξομάλυνσης και οι τιμές των παραμέτρων αυτών έχουν υπολογιστεί και είναι οι κατάλληλες που ελαχιστοποιούν το μέσο τετράγωνο σφάλματος. Αφού έχουν επιστρέψει οι σταθερές εκθετικής εξομάλυνσης, το πρόγραμμα είναι έτοιμο να ξεκινήσει τις διαδικασίες για την εύρεση των προβλέψεων. Αρχικά τρέχει ένας βρόγχος (`for loop`) για όσες περιόδους υπάρχουν, όπου βρίσκει τα `lt`, `b` και `s`. Μετά και αφού έχουν πάρει τις κατάλληλες τιμές οι συγκεκριμένες λίστες γίνονται εισαγωγή στο `DataFrame`. Στη συνέχεια υπολογίζεται, μια πρόβλεψη `f` αν ο χρήστης έχει δώσει μικρό εύρος εποχικότητας ζητώντας απλά την επόμενη τιμή ή πολλές προβλέψεις `f` αν ο χρήστης ζήτησε μεγαλύτερο εύρος εποχικότητας. Υστέρα από τη αυτή τη διαδικασία η λίστα `f` που περιέχει τις προβλέψεις εισάγεται και αυτή στο

DataFrame και καλούνται η συναρτήσεις `calce` και `calcm` όπως στις συναρτήσεις `series1`, `series2`, `series3` που μας επιστρέφουν τα Σφάλματα, το Μέσο σφάλμα, το Μέσο απόλυτο σφάλμα και το Μέσο τετράγωνο σφάλματος. Τέλος εμφανίζεται το γράφημα.

```

def series4(weeks, y):
    es=int(input("Δώσε το εύρος της εποχικότητας\n"))
    for ep in range(es):
        weeks.append(len(weeks) + 1)
        y.append(np.NaN)
    pd.set_option('display.max_columns', 10)
    df = pd.DataFrame({'Περίοδος t': weeks, 'Παρατηρήσεις Yt': y})
    sfalmata = int(input("Αριθμός σφαλμάτων για παράβλεψη"))
    df.index = np.arange(1, len(df) + 1)

    sum=0
    lt=[]
    f=[]
    b=[]
    s=[]
    Yp=[]
    for i in range(es):
        if i!=es-1:
            lt.append(np.NaN)
            b.append(np.NaN)
        else:
            lt.append(np.sum([y[0:es]])/es)
            for j in range(es):
                if math.isnan(y[es+j])==False:
                    sum=abs(((y[es+j]-y[j])/es))+sum

```

```

        s.append(round(y[j]/lt[es-1],3))
    else:
        s.append(s[j-1])
    b.append(sum/es)
Yp=YAlphaBetaGama(weeks, y, sfalmata, lt, b, s, es)
print("Μέθοδος Holt-Winter's με α= ", Yp[0], "β= ", Yp[1], "γ= ", Yp[2], "\n")
for i in range(es, len(y)-1):
    lt.append(Yp[0]*(y[i]/s[i-es]) + (1-Yp[0])*(lt[i-1]+b[i-1]))
    lt[i] = round(lt[i], 2)
    b.append(Yp[1]*(lt[i] - lt[i - 1]) + (1 - Yp[1])*b[i - 1])
    b[i] = round(b[i], 2)
    s.append(Yp[2]*y[i]/lt[i] + (1-Yp[2])*s[i-es])
    s[i] = round(s[i], 2)
b.append(np.NaN)
lt.append(np.NaN)
s.append(np.NaN)
df['Lt'] = lt
df['bt'] = b
df['St'] = s
for k in range(es):
    f.append(np.NaN)
if es == 1:
    for i in range(es, len(weeks)):
        f.append((lt[i - 1] + b[i - 1]) * s[i - es])
        f[i] = round(f[i], 2)
else:
    for i in range(es, len(weeks)-es+1):
        f.append((lt[i-1] + b[i-1])*s[i-es])
        f[i] = round(f[i], 2)
    for i in range(2, es+1):
        f.append((lt[len(lt) - es-1] + i * b[len(b) - es-1]) * s[len(s) - 2*es+i-1])
df['Ft'] = f
e = calce(y, f, len(y)-1, sfalmata)
df['et'] = e
print(df)
m = calcm(e, len(y))
print("Μέσο σφάλμα ", round(m[0], 2), "\nΜέσο απόλυτο σφάλμα ", round(m[1], 2),
      "\nΜέσο τετράγωνο σφάλματος ", round(m[2], 2))
df.plot.line(x='t', y=['yt', 'Ft'])
pl.show()
save = str(input("Αποθήκευση αποτελέσματος? Y/N\n"))
if save == "Y":
    name = str(input("Όνομα αρχείου?\n"))
    df.to_csv(name + '.csv', encoding='utf-8')

```

Δώσε το εύρος της εποχικότητας

3

Αριθμός σφαλμάτων για παράβλεψη 6

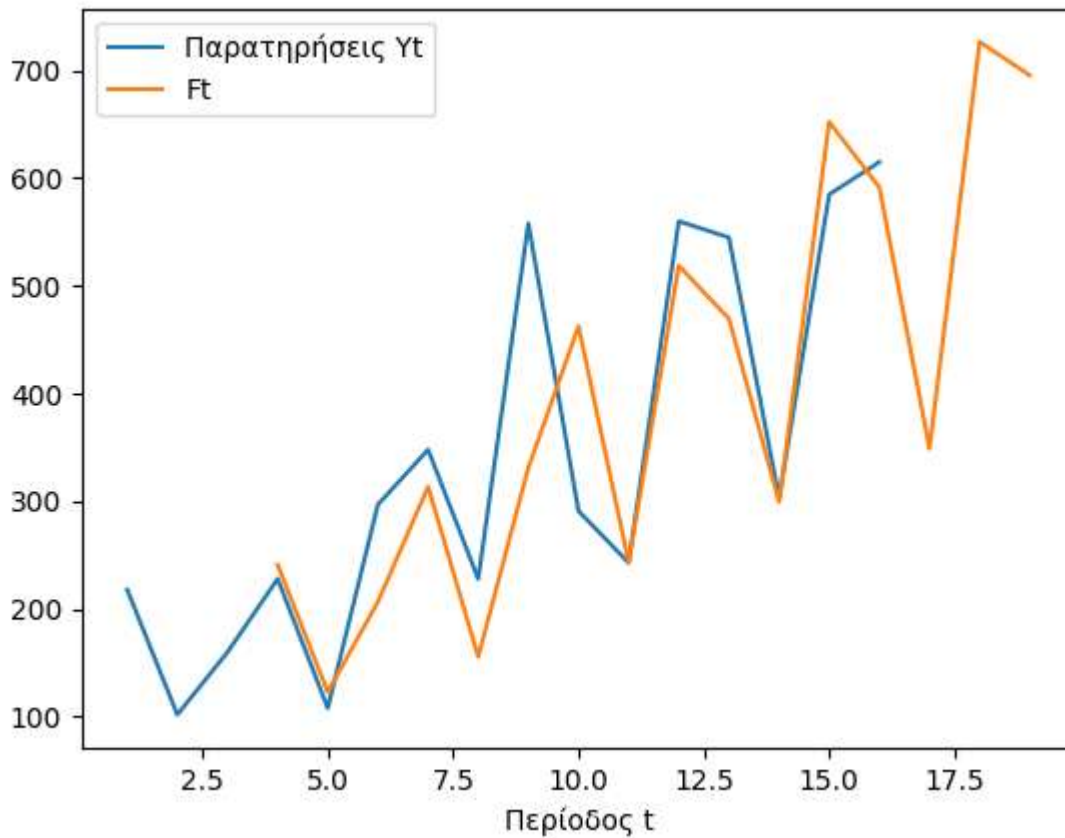
Μέθοδος Holt-Winter's με $\alpha = 0.1$ $\beta = 0.2$ $\gamma = 0.4$

	t	yt	Lt	bt	St	Ft	et
1	1	218.0	NaN	NaN	1.362	NaN	NaN
2	2	102.0	NaN	NaN	0.638	NaN	NaN
3	3	160.0	160.00	17.00	1.000	NaN	NaN
4	4	228.0	176.04	16.81	1.340	241.0700	NaN
5	5	108.0	190.49	16.34	0.610	123.0400	NaN
6	6	297.0	215.85	18.14	1.150	206.8300	NaN
7	7	348.0	236.56	18.65	1.390	313.5500	34.45
8	8	228.0	267.07	21.02	0.710	155.6800	72.32
9	9	558.0	307.80	24.96	1.420	331.3000	226.70
10	10	291.0	320.42	22.49	1.200	462.5400	-171.54
11	11	243.0	342.84	22.48	0.710	243.4700	-0.47
12	12	560.0	368.22	23.06	1.460	518.7500	41.25
13	13	545.0	397.57	24.32	1.270	469.5400	75.46
14	14	303.0	422.38	24.42	0.710	299.5400	3.46
15	15	585.0	442.19	23.50	1.410	652.3300	-67.33
16	16	615.0	467.55	23.87	1.290	591.4300	23.57
17	17	NaN	NaN	NaN	NaN	348.9100	NaN
18	18	NaN	NaN	NaN	NaN	726.5589	NaN
19	19	NaN	NaN	NaN	NaN	695.5164	NaN

Μέσο σφάλμα 23.79

Μέσο απόλυτο σφάλμα 71.66

Μέσο τετράγωνο σφάλματος 9973.27



5. Οι συναρτήσεις Y_{α} , $Y_{\alpha\beta}$ και $Y_{\alpha\beta\gamma}$ δέχονται τα ίδια ορίσματα που είναι: περίοδοι, παρατηρήσεις, τα σφάλματα. Επιπλέον η $Y_{\alpha\beta}$ δέχεται τις λίστες με τα l και u ενώ η $Y_{\alpha\beta\gamma}$ τις λίστες με τα l_t , b , s , e_s όπου εξηγούνται στη συνάρτηση `series4`. Η κάθε συνάρτηση βρίσκει τις σταθερές εκθετικής εξομάλυνσης (α, β, γ). Η Y_{α} πραγματοποιεί ελέγχους για το κατάλληλο α για κάθε δεκαδικό από το 0 μέχρι το 1, με ένα δεκαδικό ψηφίο και βήμα 0,1 έως ότου βρεθεί α που θα δώσει την ελάχιστη τιμή του μέσου τετραγώνου σφάλματος. Η $Y_{\alpha\beta}$ κάνει αντίστοιχα ότι η Y_{α} , αλλά επιπλέον συνδυάζει το β που παίρνει τιμές από 0 μέχρι 1 με ένα δεκαδικό ψηφίο και βήμα 0,1 με το α έως ότου βρεθεί α και β μαζί που να δώσουν την ελάχιστη τιμή του μέσου τετραγώνου σφάλματος. Τέλος η $Y_{\alpha\beta\gamma}$ κάνει ότι η $Y_{\alpha\beta}$ αλλά επιπλέον συνδυάζει το γ που παίρνει

τιμές από το 0 μέχρι το 1 με ένα δεκαδικό ψηφίο και βήμα 0,1, με το α και το β έως ότου βρεθεί α, β, γ μαζί δώσουν την ελάχιστη τιμή του μέσου τετραγώνου σφάλματος.

Yalpha:

```
def Yalpha(weeks, y, sfaImata):
    alpha = []
    for i in np.arange(0, 1.1, 0.1):
        alpha.append(round(i, 3))
    min1 = []
    temp = []
    A=[]
    for i in alpha:
        next = [np.NaN, y[0]]
        e = [np.NaN]
        for j in range(1, len(y) - 1):
            next.append(round(next[j] + i * (y[j] - next[j]), 3))
            next[j] = round(next[j], 1)
        for k in range(1, len(next) - 1):
            if math.isnan(next[k]):
                e.append(np.NaN)
            else:
                et = round(y[k] - next[k], 3)
                e.append(round(et, 3))
        for k in range(1, sfaImata):
            e[k] = np.NaN
        s = 0
        me = 0
        mae = 0
        mse = 0
        for z in range(len(weeks) - 1):
            if not math.isnan(e[z]):
                s += 1
                me = round((e[z] + me), 3)
                mae = round((abs(e[z]) + mae), 3)
                mse = round((pow(e[z], 2) + mse), 3)
        m = [round(me / s, 3), round(mae / s, 3), round(mse / s, 3)]
        min1.append(round(m[2], 3))
        temp.append(round(i, 3))
        df = pd.DataFrame({'MSE': min1, 'A': temp})
    A=df[df['MSE'] ==min(min1)]['A']
    return min(A)
```

YAlphaBeta:

```
def YAlphaBeta(weeks, y, sfalmata, l, u):
    alpha = []
    beta = []
    ret = []
    for i in np.arange(0, 1.1, 0.1):
        alpha.append(round(i, 3))
    for bi in np.arange(0, 1.1, 0.1):
        beta.append(round(bi, 3))

    A = []
    B = []
    min1 = []
    temp = []
    temp2 = []
    for i in alpha:
        for q in beta:
            next=[1]
            bb=[u]
            e = [np.NaN]
            for j in range(1, len(y) - 1):
                next.append(i * y[j] + (1 - i) * (next[j - 1] + bb[j - 1]))
                next[j] = round(next[j], 3)
                bb.append(q * (next[j] - next[j - 1]) + (1 - q) * bb[j - 1])
                bb[j] = round(bb[j], 3)
            next.append(np.NaN)
            f = [np.NaN]
            for sy in range(1, len(y)):
                f.append(next[sy - 1] + bb[sy - 1])
            for k in range(1, len(f) - 1):
                if math.isnan(f[k]):
                    e.append(np.NaN)
                else:
                    et = round(y[k] - f[k], 3)
                    e.append(round(et, 3))
            for k in range(1, sfalmata):
                e[k] = np.NaN
            s = 0
            me = 0
            mae = 0
            mse = 0
            msep = 0
            for z in range(len(weeks) - 1):
                if not math.isnan(e[z]):
                    s += 1
                    me = round((e[z] + me), 3)
                    mae = round((abs(e[z]) + mae), 3)
                    mse = round((pow(e[z], 2) + mse), 3)
            m = [round(me / s, 3), round(mae / s, 3), round(mse / s, 3)]
            min1.append(round(m[2], 3))
            temp.append(round(i, 3))
```

YAlphaBetaGama:

```
def YAlphaBetaGama(weeks, y, sfalmata, lt, b, s, es):  
  
    alpha = []  
    bita = []  
    es=es  
    gama=[]  
    ret = []  
    for i in np.arange(0, 1.1, 0.1):  
        alpha.append(round(i, 3))  
    for bi in np.arange(0, 1.1, 0.1):  
        bita.append(round(bi, 3))  
    for gi in np.arange(0, 1.1, 0.1):  
        gama.append(round(gi, 3))  
  
    A = []  
    B = []  
    I=[]  
    fi=[]  
    for u in range(0, es):  
        fi.append(np.NaN)  
  
    bb=[]  
    ss=[]  
    min1 = []  
    temp = []  
    temp2 = []  
    temp3=[]
```



```

for i in alpha:
    for q in bits:
        for g in gama:
            l= lt.copy()
            bb= b.copy()
            ss= s.copy()
            e = [np.NaN]
            for j in range(es, len(y) - 1):
                l.append((1 * (y[j] / ss[j - es]) + (1 - 1) * (l[j - 1] + bb[j - 1]))
                l[j] = round(l[j], 2)
                bb.append(q * (l[j] - l[j - 1]) + (1 - q) * bb[j - 1])
                bb[j] = round(bb[j], 3)
                ss.append(g * (y[j] / l[j]) + (1 - g) * ss[j - es])
                ss[j] = round(ss[j], 2)
            bb.append(np.NaN)
            l.append(np.NaN)
            ss.append(np.NaN)
            f=f1.copy()
            if es==1:
                for kt in range(es, len(y) - es+1):
                    f.append((l[kt - 1] + bb[kt - 1]) * ss[kt - es])
                    f[kt] = round(f[kt], 2)
                    f.append(np.NaN)
            else:
                for kt in range(es, len(y) - es+1):
                    f.append((l[kt - 1] + bb[kt - 1]) * ss[kt - es])
                    f[kt] = round(f[kt], 2)
            for p in range(2, es+1):
                f.append((lt[len(lt) - es - 1] + p * b[len(b) - es - 1]) * s[len(s) - 2 * es + p - 1])
            for k in range(1, len(y)-1):
                if math.isnan(f[k]):
                    e.append(np.NaN)
                else:
                    et = round(y[k] - f[k], 3)
                    e.append(round(et, 3))
            for k in range(1, sfalmata):
                e[k] = np.NaN
            si = 0
            me = 0
            mae = 0
            mse = 0
            msep = 0
            for z in range(len(weeks) - 1):
                if not math.isnan(e[z]):
                    si += 1
                    me = round((e[z] + me), 3)
                    mae = round((abs(e[z]) + mae), 3)
                    mse = round((pow(e[z], 2) + mse), 3)
            m = [round(me / si, 3), round(mae / si, 3), round(mse / si, 3)]
            min1.append(round(m[2], 3))

```

Το τελευταίο κομμάτι του κώδικα πρόκειται για το κύριο πρόγραμμα. Ουσιαστικά, πρόκειται για το τμήμα που θα τρέξει πρώτο μόλις εκτελεστεί το πρόγραμμα. Εμφανίζεται ένα μήνυμα που ενημερώνει το χρήστη για τις διαθέσιμες επιλογές. Μπορεί να επιλέξει να εισάγει τα δεδομένα που θέλει χειροκίνητα ή να τα φορτώσει μέσω αρχείου txt. Αφού επιλέξει τι θέλει στη συνέχεια επιλέγει με ποια μέθοδο θέλει να κάνει πρόβλεψη για τα δεδομένα του. Έπειτα ανάλογα με την επιλογή του παίρνουν σειρά οι ανάλογες συναρτήσεις.

```
print("Πάτησε 1 για να φορτώσεις δεδομένα από αρχείο TXT ή 2 για να φορτώσεις τα δεδομένα χειροκίνητα\n")
epilogi=int(input())
if epilogi == 2:
    paratiriseis=[]
    periodos=[]
    print("Δώσε περίοδο και πάτησε -1")
    temperiodos = float(input())
    while temperiodos != -1:
        periodos.append(temperiodos)
        temperiodos = float(input())
    print("Δώσε τις παρατηρήσεις και πάτησε -1")
    temparatiriseis= float(input())
    while temparatiriseis != -1:
        paratiriseis.append(temparatiriseis)
        temparatiriseis = float(input())
    df=pd.DataFrame({'Περίοδος': periodos, 'Παρατηρήσεις': paratiriseis})
    print(periodos,paratiriseis)
    df.plot.line(x='Περίοδος', y=['Παρατηρήσεις'])
    pl.show()
if user == 1:
    series1(periodos,paratiriseis)
elif user == 2:
    series2(periodos,paratiriseis)
elif user == 3:
    series3(periodos,paratiriseis)
else:
    series4(periodos,paratiriseis)
```

Τα παραπάνω παραδείγματα έχουν ληφθεί από το βιβλίο Διοίκηση παραγωγικών συστημάτων- Βασικές θεωρητικές αρχές και εφαρμογές στη λήψη διοικητικών αποφάσεων(2^η έκδοση) των κυρίων Δημητριάδη Γ. Σωτήρη και Μιχιώτη Ν. Αθανάσιο.

Πιο συγκεκριμένα χρησιμοποιήθηκαν τα παραδείγματα:

Πίνακας 2.10 σελ 46,

Σχήμα 2.8 σελ 58

Πίνακας 2.16 σελ 63

Πίνακας 2.18 σελ 68

Οι δύο πρώτες συναρτήσεις έχουν ακρίβεια στα αποτελέσματα σχεδόν 100%. Ενώ οι επόμενες δυο έχουν απόκλιση κοντά στο 5% και αυτό λόγω των δεκαδικών ψηφίων των πράξεων και των συναρτήσεων εύρεσης των σταθερών εξομάλυνσης που κάνουν εξαντλητική αναζήτηση από το 0 μέχρι το 1 αλλά μόνο στα νούμερα με ένα δεκαδικό.

4. Σχεδιασμός περιβάλλοντος και οδηγίες χρήσης

Για την ανάπτυξη της εφαρμογής έχει χρησιμοποιηθεί το πρόγραμμα PyCharm. Πρόκειται για ένα ολοκληρωμένο περιβάλλον ανάπτυξης Python (IDE). Περιέχει ένα τεράστιο φάσμα βασικών εργαλείων για τους προγραμματιστές. Είναι διαθέσιμο για Windows, macOS και Linux. Υποστηρίζει την έκδοση Python 2.7, καθώς και όλες τις εκδόσεις από Python 3.6 έως 3.11.

Το πρόγραμμα κυκλοφορεί σε τρεις διαφορετικές εκδόσεις:

Community: Η έκδοση αυτή διατίθεται δωρεάν και είναι ανοικτού κώδικα. Συμπεριλαμβάνει βοήθεια κώδικα, αναδιαμορφώσεις και οπτική αποσφαλμάτωση.

Professional: Η έκδοση αυτή διατίθεται επί πληρωμή και κυκλοφορεί για επαγγελματική ανάπτυξη Python. Επίσης, μπορεί να χρησιμοποιηθεί για ανάπτυξη εφαρμογών Django, Flask και Pyramid. Επιπλέον υποστηρίζει ανάπτυξη ιστοσελίδων μέσω των γλωσσών HTML5, CSS, JavaScript και XML με ήδη ενεργοποιημένα plug-ins. Περιλαμβάνει βοήθεια κώδικα, αναδιαμορφώσεις, οπτική αποσφαλμάτωση και

απομακρυσμένες διαμορφώσεις. Επιπρόσθετα, χρησιμοποιείται για ανάπτυξη και υποστήριξη βάσεων δεδομένων αλλά και επιστημονικών εργαλεία και εργαλείων μεγάλων δεδομένων.

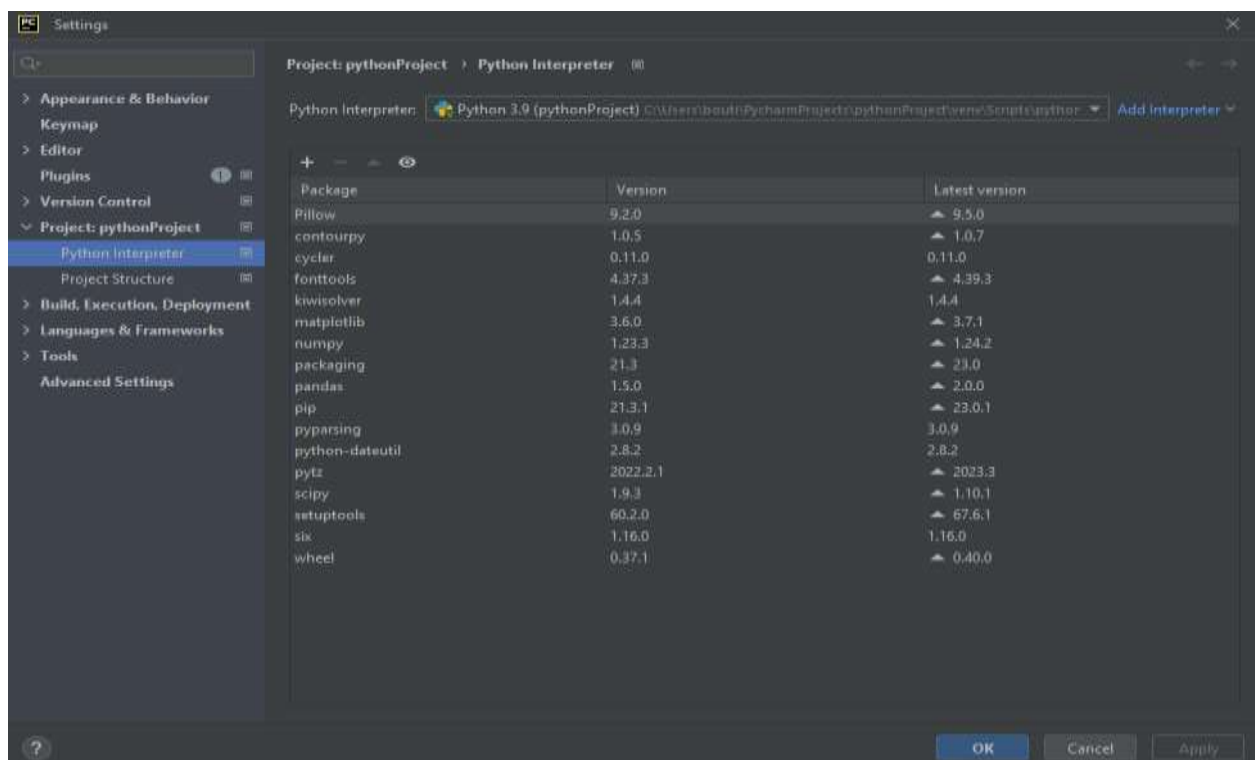
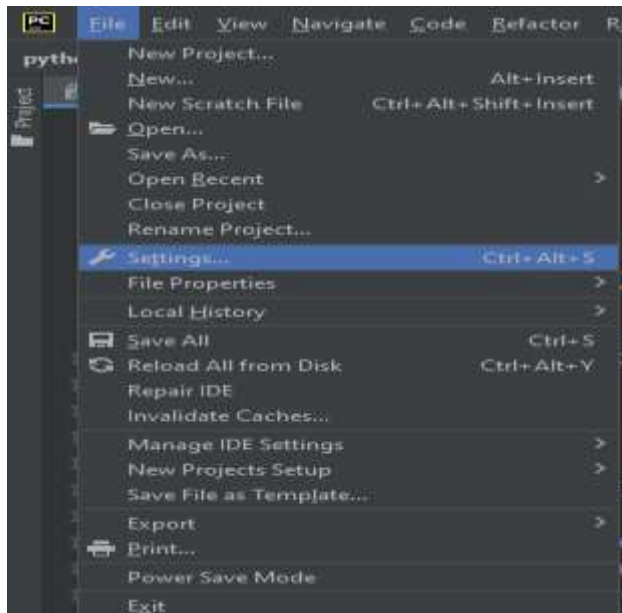
Edu: Διατίθεται επίσης δωρεάν και είναι ανοιχτού κώδικα. Πρόκειται για την εκπαιδευτική έκδοση του προγράμματος. Περιέχει εκπαιδευτικά εργαλεία και χρησιμοποιείται για την εκμάθηση γλωσσών προγραμματισμού. και συναφών τεχνολογιών με ενσωματωμένα εκπαιδευτικά εργαλεία.

Παρακάτω δίνεται αναλυτικά ο τρόπος που γίνεται εγκατάσταση το περιβάλλον pyCharm και πως τρέχει το πρόγραμμα.

Αρχικά ο χρήστης κατεβάζει το πρόγραμμα pyCharm για windows από τον παρακάτω σύνδεσμο: <https://www.jetbrains.com/pycharm/download/#section=windows>

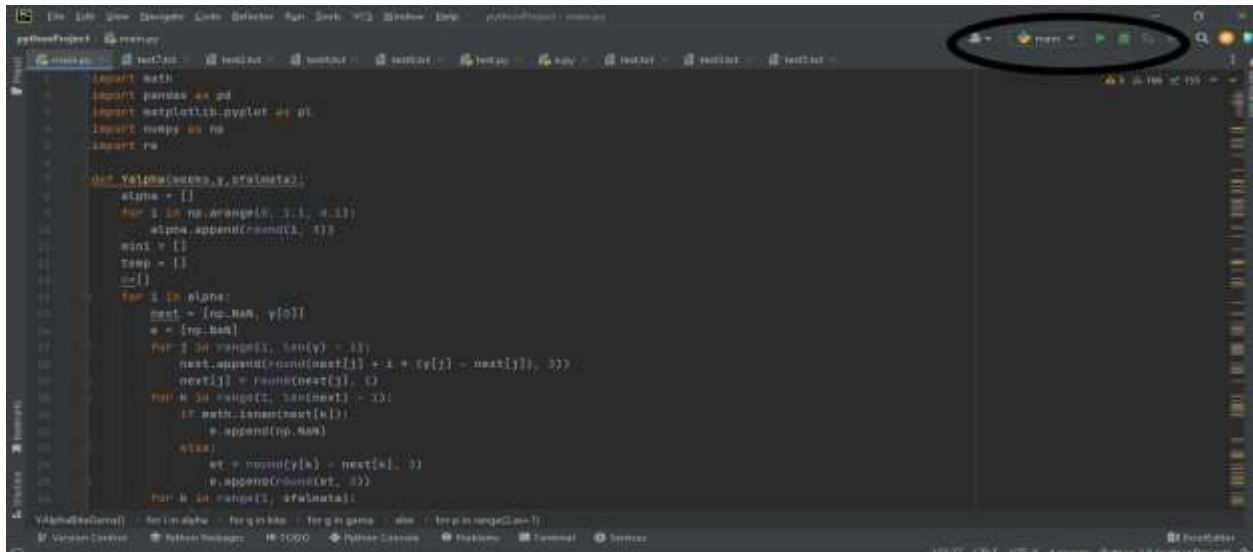


Αφού ο χρήστης επιλέξει την έκδοση community εκτελεί την εγκατάσταση και ακολουθεί όλα τα βήματα που του προτείνει το πρόγραμμα. Όταν τελειώσει η εγκατάσταση επιλέγει νέο project. Μόλις δημιουργηθεί το project, πρέπει να πάει στις ρυθμίσεις να επιλέξει τις κατάλληλες βιβλιοθήκες (libraries) που χρειάζεται το project για να δουλέψει επιτυχώς.



Το project έχει φορτωμένες τις βασικές βιβλιοθήκες, αλλά χρειάζονται και κάποιες επιπλέον για τους σωστούς υπολογισμούς και για τις γραφικές παραστάσεις. Αυτές είναι: (numpy, pip, pandas, matplotlib, scipy) όπως φαίνονται στην παραπάνω εικόνα. Έπειτα το project είναι έτοιμο να δουλέψει. Ο χρήστης κάνει αντιγραφεί το αρχείο(main.py) με τον κώδικα στον φάκελο του project και αν έχει και τα αρχεία με

τα δεδομένα σε μορφή txt. Τα αρχεία των δεδομένων πρέπει να βρίσκονται στον ίδιο φάκελο με το main.py Στη συνέχεια πατάει την επιλογή play για να ξεκινήσει το πρόγραμμα.



Αμέσως του εμφανίζεται η επιλογή να διαλέξει αν θέλει να φορτώσει δεδομένα από αρχείο txt ή να τα εισάγει ο ίδιος.



Αν διαλέξει την πρώτη επιλογή θα πρέπει το αρχείο txt να έχει μια σειρά, όπου θα βρίσκονται οι παρατηρήσεις και κάθε τιμή να χωρίζεται με κόμμα.

```
Run: main
C:\Users\bouti\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\bouti\PycharmProjects\pythonProject\main.py
Πάτησε 1 για να φορτώσεις δεδομένα από αρχείο TXT ή 2 για να φορτώσεις τα δεδομένα χειροκίνητα
Όνομα αρχείου
|
```

```
1 204, 292, 284, 228, 187, 224, 278, 247, 230, 171
```

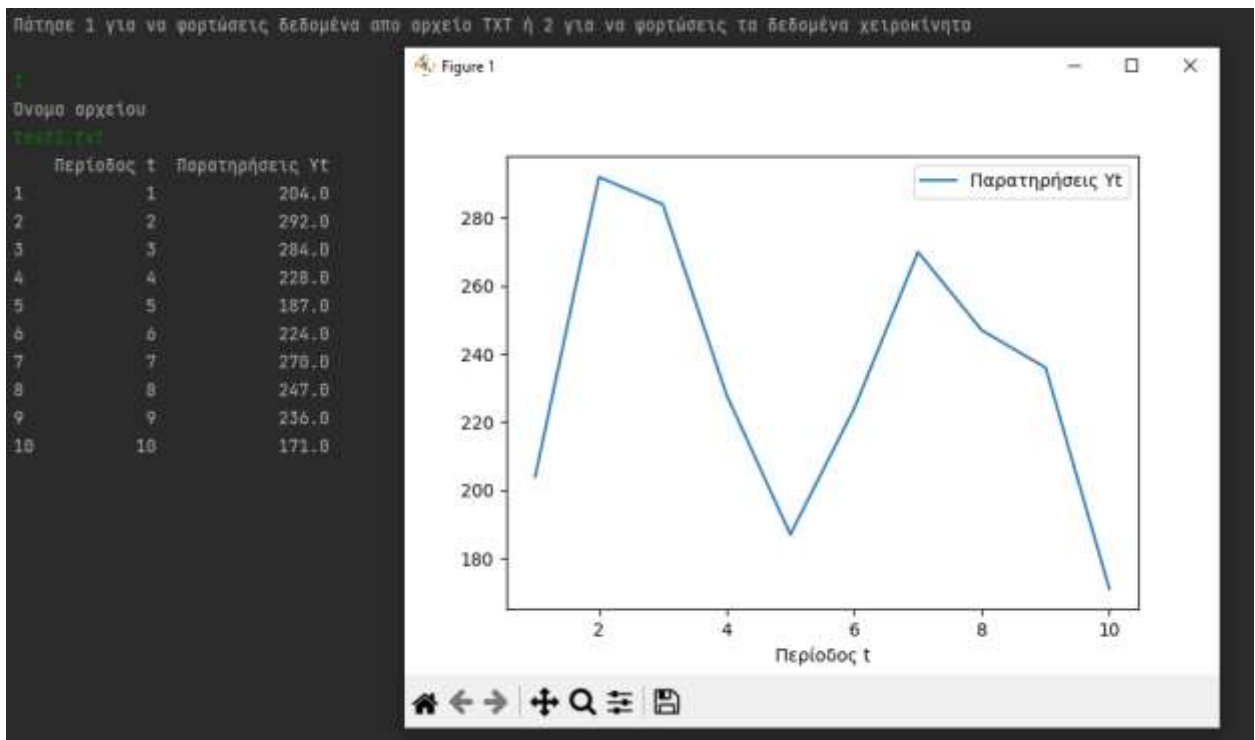
Αν επιλέξει τη δεύτερη επιλογή τότε, εισάγει τις παρατηρήσεις όπου πατάει enter μετά από κάθε τιμή και στο τέλος το -1 που σηματοδοτεί ότι τελείωσε την εισαγωγή.

```
Πάτησε 1 για να φορτώσεις δεδομένα από αρχείο TXT ή 2 για να φορτώσεις τα δεδομένα χειροκίνητα
2
Δώσε τις παρατηρήσεις και πάτησε -1
204
292
284
228
187
224
276
247
236
171
-1

```

Περίοδος	Παρατηρήσεις
1	204.0
2	292.0
3	284.0
4	228.0
5	187.0
6	224.0
7	276.0
8	247.0
9	236.0
10	171.0

Στη συνέχεια, αφού έχουν εισαχθεί επιτυχώς τα δεδομένα, εμφανίζεται η γραφική παράσταση.



Ο χρήστης αφού δει τη γραφική παράσταση και καταλάβει τη χρονοσειρά, κλείνει το παράθυρο της παράστασης και του εμφανίζονται οι 4 επιλογές με τις μεθόδους πρόβλεψης.

```
Πάτησε 1 για να φορτώσεις δεδομένα απο αρχείο TXT ή 2 για να φορτώσεις τα δεδομένα χειροκίνητα
```

```
1
```

```
Όνομα αρχείου
```

```
test1.txt
```

	Περίοδος t	Παρατηρήσεις Yt
1	1	204.0
2	2	292.0
3	3	284.0
4	4	228.0
5	5	187.0
6	6	224.0
7	7	270.0
8	8	247.0
9	9	236.0
10	10	171.0

```
Διάλεξε χρονοσειρά
```

1. Απλός κινούμενος μέσος
2. Απλή εκθετική εξομάλυνση
3. Μέθοδος Holt δυο παραμέτρων
4. Μέθοδος Holt-Winter's για τάση και εποχικότητα

Έστω ότι επιλέγει την πρώτη επιλογή του απλού κινούμενου μέσου. Τότε του ζητείται να επιλέξει τον αριθμό k.

```
Πάτησε 1 για να φορτώσεις δεδομένα απο αρχείο TXT ή 2 για να φορτώσεις τα δεδομένα χειροκίνητα
```

```
1
```

```
Όνομα αρχείου
```

```
test1.txt
```

	Περίοδος t	Παρατηρήσεις Yt
1	1	204.0
2	2	292.0
3	3	284.0
4	4	228.0
5	5	187.0
6	6	224.0
7	7	270.0
8	8	247.0
9	9	236.0
10	10	171.0

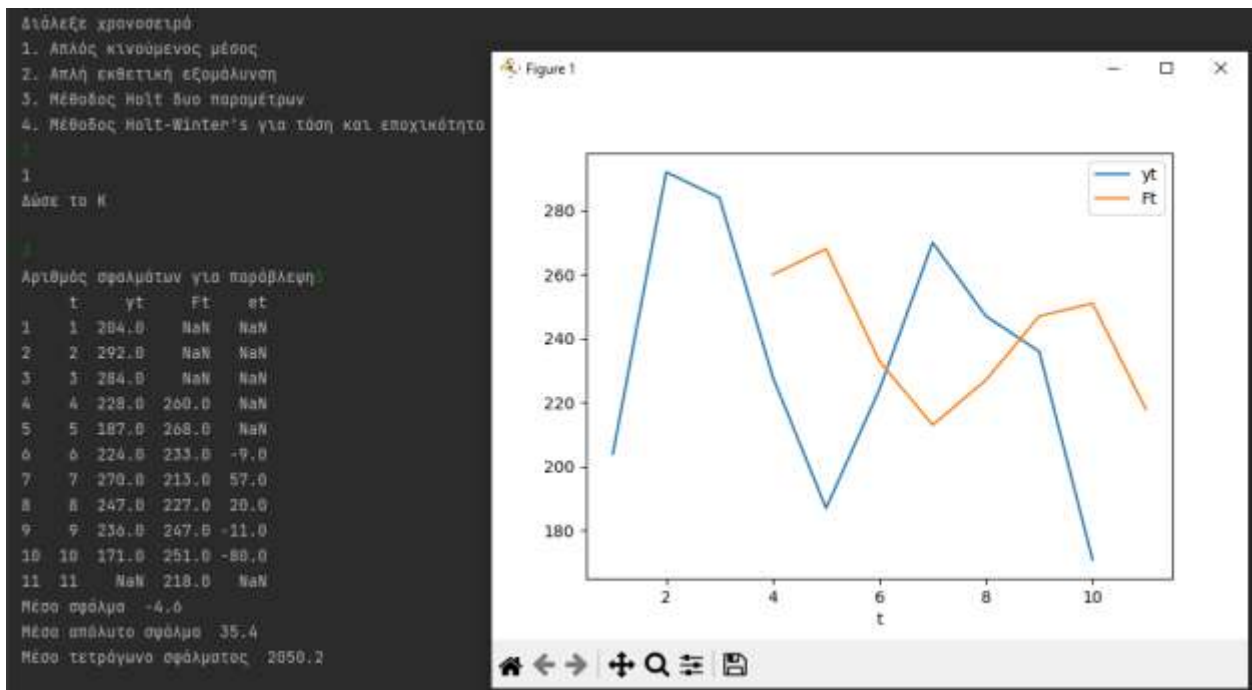
```
Διάλεξε χρονοσειρά
```

1. Απλός κινούμενος μέσος
2. Απλή εκθετική εξομάλυνση
3. Μέθοδος Holt δυο παραμέτρων
4. Μέθοδος Holt-Winter's για τάση και εποχικότητα

```
1
```

```
Δώσε το k
```

Ύστερα ο χρήστης δίνει τον αριθμό των σφαλμάτων. Εδώ επιλέγει τον αριθμό των περιόδων που θα προσπεράσει το πρόγραμμα ώστε να υπολογίσει τα σφάλματα των υπόλοιπων περιόδων. Πχ αν έχει 10 περιόδους, ο χρήστης δίνει σφάλματα 5, τότε θα υπολογιστούν τα σφάλματα από την 6^η σειρά μέχρι τη 10^η. Τέλος γίνονται οι υπολογισμοί του προγράμματος και εμφανίζει έναν πίνακα που περιέχει τις περιόδους, τις παρατηρήσεις, τις προβλέψεις και τα σφάλματα. Στο τέλος του πίνακα εμφανίζει το μέσο σφάλμα, το μέσο απόλυτο σφάλμα και το μέσο τετράγωνο σφάλματος όπως επίσης και τη γραφική παράσταση των προβλέψεων με τις παρατηρήσεις.



Τέλος αφού ο χρήστης δει τα αποτελέσματα, το πρόγραμμα του δίνει την επιλογή να τα αποθηκεύσει αν επιθυμεί σε αρχείο excel.

```
Αποθήκευση αποτελέσματος? Y/N
Y
Όνομα αρχείου?
0011201
```

5. Συμπεράσματα

Συνεπώς, η πρόβλεψη είναι ζωτικής σημασίας για μια επιχείρηση. Στην εργασία αυτή αναφέρθηκαν οι μέθοδοι προβλέψεων και μελετήθηκαν αναλυτικά αυτές που βασίζονται σε ιστορικά δεδομένα καταταγμένα με χρονολογική σειρά. Κατά τη διαδικασία ανάπτυξης της πρόβλεψης ακολουθούνται συγκεκριμένα βήματα. Αρχικά είναι απαραίτητο να κατανοηθεί ο σκοπός για τον οποίο γίνεται η πρόβλεψη. Έπειτα, τα μοντέλα που θα αναπτυχθούν θα πρέπει να εφαρμοστούν στα δεδομένα αφού πληρούν, βέβαια, συγκεκριμένες προϋποθέσεις. Τα αποτελέσματα θα πρέπει να αξιολογηθούν σχετικά με την απόδοση τους με χρήση διάφορων κριτηρίων και ελέγχων ακρίβειας. Τα κριτήρια αυτά συνήθως περιγράφουν τα αποτελέσματα μαθηματικών εξισώσεων που, σαν εισόδους, δέχονται τα σφάλματα των προβλέψεων, δηλαδή τις αποκλίσεις των προβλεπόμενων δεδομένων από τα πραγματικά. Για την τελική επιλογή του κατάλληλου μοντέλου πρόβλεψης, η ανάπτυξη πολλαπλών μοντέλων από διαφορετικές μεθόδους και η σύγκριση αυτών έχει θεμελιώδη σημασία. Με αυτόν τον τρόπο, οι αναλυτές λαμβάνουν τις σωστότερες αποφάσεις, με σκοπό την ελαχιστοποίηση του ρίσκου και τη μεγιστοποίηση του κέρδους.

5.1 Πιθανοί τρόποι εξέλιξης

Κάποιοι πιθανοί τρόποι εξέλιξης θα ήταν εξής: Να γίνει πιο απλούστερος ο κώδικας όσον αφορά τις συναρτήσεις εύρεσης των σταθερών εξομάλυνσης (α, β, γ), ώστε να

επιταχυνθεί η εύρεση των τιμών αλλά και η ακρίβεια τους. Επίσης μπορεί να δημιουργηθεί ένα γραφικό περιβάλλον που μπορεί να αλληλεπιδρά ο χρήστης με την εφαρμογή για την εισαγωγή των δεδομένων, την εξαγωγή και την επεξεργασία των αποτελεσμάτων.

6.Βιβλιογραφία

1. The Analysis of Time Series: An Introduction with R (Chapman & Hall/CRC Texts in Statistical Science) - Chris Chatfield, Haipeng Xing (7th Edition)
2. Time Series Analysis - James D. Hamilton
3. Introduction to Time Series Analysis and Forecasting - Douglas C. Montgomery, Cheryl L. Jennings, and Murat Kulahci
4. Τεχνικές Ανάλυσης και Πρόβλεψης Τηλεπικοινωνιακών Αγορών - Μιχαήλ Κυριακίδης
5. Διοίκηση παραγωγικών συστημάτων: Βασικές θεωρητικές αρχές και εφαρμογές στη λήψη διοικητικών αποφάσεων - Δημητριάδης Γ. Σωτήριος, Μιχιώτης Ν. Αθανάσιος (2^η έκδοση)
6. <https://docs.python.org/3/faq/>
7. https://pandas.pydata.org/docs/user_guide/dsintro.html
8. https://numpy.org/doc/stable/user/absolute_beginners.html
9. <https://www.jetbrains.com/pycharm/features/>