



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Εφαρμογή ιστού για την πρόβλεψη τιμών υγρών
καυσίμων στην Ελληνική αγορά μέσω ανάλυσης
χρονοσειρών»

Του φοιτητή
Μαρινίδης Αντώνιος
Αρ. Μητρώου: 144276

Επιβλέπων
Ουγιάρογλου Στέφανος
Επ. Καθηγητής.

17 Ιουνίου 2023

Τίτλος Π.Ε. Εφαρμογή ιστού για την πρόβλεψη τιμών υγρών καυσίμων στην Ελληνική αγορά μέσω ανάλυσης χρονοσειρών

Κωδικός Π.Ε. 22328

Όνοματεπώνυμο φοιτητή : Μαρινίδης Αντώνιος

Όνοματεπώνυμο εισηγητή : Ουγιάρογλου Στέφανος

Ημερομηνία ανάληψης Π.Ε. : 21-10-2022

Ημερομηνία περάτωσης Π.Ε.

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Μαρινίδη Αντώνιου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίας στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Περίληψη

Στην παρούσα πτυχιακή εργασία, αναλύεται η διαδικασία με την οποία το υπουργείο Ανάπτυξης και Επενδύσεων καταγράφει καθημερινά τις τιμές των υγρών καυσίμων στη χώρα και δημοσιεύει ένα δελτίο επισκόπησης τιμών στην ιστοσελίδα του, <http://www.fuelprices.gr/>. Τα δεδομένα που παρέχονται μέσω αυτού του δελτίου περιλαμβάνουν τις καθημερινές παρατηρήσεις των τιμών υγρών καυσίμων ανά νομό της Ελλάδος και ανά τύπο καυσίμου, δημιουργώντας έτσι χρονοσειρές δεδομένων. Στο πλαίσιο προηγούμενης διπλωματικής εργασίας, αναπτύχθηκε ένας μηχανισμός για την αυτόματη λήψη και αποθήκευση αυτών των δεδομένων σε μια σχεσιακή βάση δεδομένων. Στόχος της παρούσας πτυχιακής εργασίας είναι να διερευνήσει τις βέλτιστες μεθόδους ανάλυσης χρονοσειρών, που θα επιτρέψουν την πρόβλεψη των μελλοντικών τιμών των υγρών καυσίμων. Επιπλέον, θα υλοποιηθεί μια ιστοσελίδα, όπου οι χρήστες θα έχουν τη δυνατότητα να παρακολουθούν τις χρονοσειρές των τιμών των υγρών καυσίμων ανά νομό και τύπο καυσίμου, καθώς και τις προβλέψεις για τις μελλοντικές τιμές στην Ελλάδα. Παράλληλα, θα παρουσιάζονται οι εκτιμήσεις των προβλέψεων που έχουν πραγματοποιηθεί από τις μεθόδους ανάλυσης χρονοσειρών, προκειμένου να αξιολογηθεί η αποτελεσματικότητα και η ακρίβεια τους. Στη βιβλιογραφική μελέτη, θα αναφερθούν οι κυριότερες μέθοδοι ανάλυσης χρονοσειρών, καθώς και οι τεχνικές και οι προσεγγίσεις που εφαρμόζονται στην πρόβλεψη των τιμών των καυσίμων. Συνολικά, η πτυχιακή εργασία θα συμβάλει στη βελτίωση της κατανόησης των μεθόδων ανάλυσης χρονοσειρών και της εφαρμογής τους στην πρόβλεψη των τιμών των υγρών καυσίμων. Επιπλέον, η δημιουργία της ιστοσελίδας θα προσφέρει στους χρήστες ένα εύχρηστο εργαλείο για την παρακολούθηση των τάσεων των τιμών και των προβλέψεων, βοηθώντας τους να προγραμματίζουν τις αγορές τους με βάση πιο αξιόπιστα και συστηματικά στοιχεία. Η πτυχιακή εργασία θα αποτελέσει μια σημαντική συμβολή στη βελτίωση των διαθέσιμων εργαλείων και τεχνικών για την πρόβλεψη των τιμών των υγρών καυσίμων και θα αποτελέσει ένα χρήσιμο παράδειγμα για μελλοντικές εργασίες και εφαρμογές στον τομέα της ενέργειας και της οικονομίας.

Abstract

In this thesis, the process by which the Ministry of Development and Investment records daily prices of liquid fuels in the country and publishes a price overview bulletin on its website, <http://www.fuelprices.gr/>, is analyzed. The data provided through this bulletin includes daily observations of liquid fuel prices by county in Greece and by fuel type, thus creating time series of data. As part of a previous thesis, a mechanism was developed to automatically download and store this data in a relational database. The aim of this thesis is to investigate the optimal methods of time series analysis, which will allow forecasting future liquid fuel prices. In addition, a website will be implemented where users will be able to monitor time series of liquid fuel prices by county and fuel type, as well as forecasts of future prices in Greece. At the same time, estimates of the forecasts made by the time series analysis methods will be presented in order to assess their effectiveness and accuracy. In the literature study, the main time series analysis methods will be mentioned, as well as the techniques and approaches applied in the forecasting of fuel prices. Overall, the thesis will contribute to an improved understanding of time series analysis methods and their application to the forecasting of liquid fuel prices. In addition, the creation of the website will provide users with an easy-to-use tool to monitor price trends and forecasts, helping them to plan their purchases based on more reliable and systematic data. The thesis will be an important contribution to improving the tools and techniques available for forecasting liquid fuel prices and will provide a useful example for future work and applications in the energy sector and the economy.

Περιεχόμενα

Περίληψη	ii
Abstract	iii
Περιεχόμενα	iv
Κατάλογος Σχημάτων	v
Συντομογραφίες	vi
1 Εισαγωγή	1
1.1 Υγρά καύσιμα και οι τιμές	1
1.1.1 Παγκόσμιοι παράγοντες που επηρεάζουν τις τιμές των υγρών καυσίμων	1
1.1.2 Φυσικά φαινόμενα και κλιματικές αλλαγές	1
1.1.3 Ελληνική αγορά υγρών καυσίμων	2
1.1.4 Προσαρμογή της Ελλάδας στις μεταβαλλόμενες παγκόσμιες συνθήκες	3
1.2 Κίνητρο	4
1.3 Συνεισφορά	5
1.4 Οργάνωση	6
2 Το σύνολο δεδομένων	8
2.1 Πρωτογενή δεδομένα	8
2.2 Προ-επεξεργασία δεδομένων	10
2.3 Δημιουργία σχεσιακής Βάσης Δεδομένων	11
3 Ανάλυση χρονοσειρών και το μοντέλο ARIMA	13
3.1 Χρονοσειρές	13
3.2 Μέθοδοι ανάλυσης χρονοσειρών	13
3.3 Πρόβλεψη μελλοντικών τιμών χρονοσειρών	14
3.4 Το μοντέλο ARIMA	16
4 Τεχνολογίες	19
4.1 Τεχνολογίες Front-End	19
4.1.1 HTML5 και CSS	19
4.1.2 Javascript και βιβλιοθήκες της	20
4.2 Τεχνολογίες Back-End	22
4.2.1 Python	22
4.2.2 FastAPI	22
4.2.3 MySQL	23
5 Πρόβλεψη τιμών υγρών καυσίμων με το μοντέλο ARIMA	24
5.1 Προ-επεξεργασία των δεδομένων στην Python	24
5.2 Χρήση του μοντέλου ARIMA στην Python	29
5.3 Καθορισμός των παραμέτρων	32
5.4 Εκτίμηση των προβλέψεων	38
6 Υλοποίηση του FuelPricesApp	39
6.1 Αρχιτεκτονική του FuelPricesApp	39
6.2 Back-End	39
6.3 Front-End	48
7 Παρουσίαση του FuelPricesApp / Case studies	56
7.1 Αρχική σελίδα εφαρμογής	56
7.2 Παρουσίαση τιμών υγρών καυσίμων	57
7.3 Πρόβλεψη μελλοντικών τιμών	58
8 Συμπεράσματα - Μελλοντική εργασία	60
8.1 Συμπεράσματα	60
8.2 Μελλοντική εργασία	60
BIBΛΙΟΓΡΑΦΙΑ	62

Κατάλογος Σχημάτων

2.1	Παράδειγμα δελτίου με τις τιμές των υγρών καυσίμων ανά νομό της Ελληνικής επικράτειας . . .	9
2.2	Πίνακες σχεσιακού μοντέλου από άλλη διπλωματική εργασία	11
2.3	Πίνακες σχεσιακού μοντέλου για την παρούσα πτυχιακή εργασία	12
5.1	Dataframe του πίνακα prices	25
5.2	Γραφήματα των καυσίμων "αέριο" και "αμόλυβδη 100" για τον νομό Άρτης του πίνακα prices . .	25
5.3	Γραφήματα των καυσίμων "αέριο" και "αμόλυβδη 100" για τον νομό Άρτης του πίνακα interpolated_prices	28
5.4	Dataframe του πίνακα interpolated_prices	28
5.5	Dataframe της καθημερινής μέσης τιμής του αερίου στην Ελλάδα	33
5.6	Αποτελέσματα βέλτιστων παραμέτρων για το μοντέλο ARIMA	34
5.7	Οπτικοποίηση των αποτελεσμάτων - προβλέψεων	36
5.8	Αξιολόγηση των αποτελεσμάτων - προβλέψεων	36
5.9	Οπτικοποίηση των αποτελεσμάτων - προβλέψεων	37
6.1	Διάγραμμα αρχιτεκτονικής του FuelPricesApp	39
7.1	Αρχική σελίδα της εφαρμογής FuelPricesApp	57
7.2	Διεπαφή παρουσίασης τιμών υγρών καυσίμων της εφαρμογής FuelPricesApp	58
7.3	Διεπαφή προβλέψεων τιμών υγρών καυσίμων της εφαρμογής FuelPricesApp	59

Συνομογραφίες

Π.Ε. Πτυχιακή Εργασία
ΔΙΠΑΕ Διεθνές Πανεπιστήμιο Ελλάδος

Κεφάλαιο 1ο: Εισαγωγή

1.1 Υγρά καύσιμα και οι τιμές

Οι τιμές των υγρών καυσίμων αποτελούν ένα σημαντικό παράγοντα που επηρεάζει την παγκόσμια οικονομία και την καθημερινή ζωή των ανθρώπων. Τα τελευταία χρόνια, οι τιμές αυτές υφίστανται σημαντικές διακυμάνσεις και μεταβολές, καθώς επηρεάζονται από ποικίλους παράγοντες, όπως η πολιτική, οικονομική και γεωπολιτική κατάσταση, η ζήτηση και προσφορά, καθώς και τα φυσικά φαινόμενα. Η ελληνική αγορά υγρών καυσίμων δεν αποτελεί εξαίρεση σε αυτό το πλαίσιο, καθώς οι τιμές επηρεάζονται από τις ίδιες παγκόσμιες δυνάμεις και αντιμετωπίζει παρόμοιες προκλήσεις.

1.1.1 Παγκόσμιοι παράγοντες που επηρεάζουν τις τιμές των υγρών καυσίμων

Η πολιτική και η γεωπολιτική κατάσταση παίζουν κρίσιμο ρόλο στη διαμόρφωση των τιμών των καυσίμων. Συγκρούσεις και αστάθεια στις περιοχές παραγωγής πετρελαίου, όπως τη Μέση Ανατολή, μπορούν να οδηγήσουν σε διακοπές της παραγωγής και κατ'επέκταση στην αύξηση των τιμών. Αντίθετα, η συνεργασία μεταξύ χωρών-παραγωγών και οι πολιτικές αποφάσεις που αφορούν την παραγωγή και την εξαγωγή καυσίμων μπορεί να σταθεροποιήσουν τις τιμές [1].

Οι οικονομικές κρίσεις και η ανάκαμψη επηρεάζουν τη ζήτηση για καύσιμα, καθώς η οικονομική ανάπτυξη συχνά συνδέεται με αυξημένη κατανάλωση ενέργειας. Οικονομικές ύφεσης μπορούν να οδηγήσουν σε μείωση της ζήτησης και πτώση των τιμών, ενώ η ανάκαμψη και η ανάπτυξη μπορούν να αυξήσουν τη ζήτηση και τις τιμές.

Η προσφορά και η ζήτηση στην αγορά πετρελαίου είναι ένας άλλος σημαντικός παράγοντας που επηρεάζει τις τιμές των καυσίμων. Η αύξηση της παραγωγής πετρελαίου από μη παραδοσιακές πηγές, όπως η ξιστική εκμετάλλευση (fracking), μπορεί να οδηγήσει σε περισσότερο ανταγωνισμό και μειωμένες τιμές. Αντίθετα, οι περιορισμοί στην παραγωγή ή οι κυρώσεις κατά χωρών παραγωγών μπορεί να μειώσουν την προσφορά και να αυξήσουν τις τιμές [2].

Οι επενδύσεις και η ανάπτυξη υποδομών επίσης συμβάλλουν στη διαμόρφωση των τιμών των υγρών καυσίμων. Η κατασκευή νέων αγωγών, τερματικών και εγκαταστάσεων αποθήκευσης μπορεί να βελτιώσει την αποδοτικότητα και την ασφάλεια της παραγωγής, μεταφοράς και διανομής καυσίμων, ενώ ταυτόχρονα μπορεί να μειώσει το κόστος και τις τιμές.

1.1.2 Φυσικά φαινόμενα και κλιματικές αλλαγές

Φυσικά φαινόμενα, όπως τυφώνες, πλημμύρες, και ξηρασίες, μπορούν να προκαλέσουν σημαντικές διαταραχές στην παραγωγή, μεταφορά, και διανομή των υγρών καυσίμων. Τα καιρικά φαινόμενα μπορούν να κλείσουν πετρελαιοαγωγούς, να καταστρέψουν υποδομές, και να περιορίσουν τη μεταφορά καυσίμων μέσω πλοίων. Αυτές οι διαταραχές μπορούν να οδηγήσουν σε προσωρινές αυξήσεις των τιμών και να προκαλέσουν αβεβαιότητα στην αγορά.

Ένας ακόμα παράγοντας επιρροής των τιμών είναι η υγειονομική κατάσταση. Η πανδημία του COVID-19 για παράδειγμα, έχει σημαντικό αντίκτυπο στην παγκόσμια αγορά καυσίμων. Κατά τη διάρκεια της πανδημίας, οι κυβερνήσεις ανά τον κόσμο εφάρμοσαν αυστηρά μέτρα περιορισμού για να περιορίσουν την εξάπλωση του ιού, όπως την απαγόρευση κυκλοφορίας και το κλείσιμο των συνόρων. Αυτά τα μέτρα οδήγησαν σε μαζικές μειώσεις των μεταφορών και της βιομηχανικής δραστηριότητας, προκαλώντας μία απότομη πτώση της ζήτησης για υγρά καύσιμα [3].

Ως αποτέλεσμα, οι τιμές των καυσίμων σημείωσαν ραγδαίες πτώσεις, καθώς η ζήτηση ήταν χαμηλή, και οι προμηθευτές πετρελαίου αντιμετώπιζαν το πρόβλημα της υπερπαραγωγής σε συνδυασμό με τον περιορισμένο χώρο αποθήκευσης. Αυτό οδήγησε σε μια πρωτοφανή κατάσταση, όπου οι τιμές του πετρελαίου για σύντομο διάστημα έγιναν αρνητικές, καθώς οι προμηθευτές ήταν αναγκασμένοι να πληρώσουν για να απαλλαγούν από το περίσσειο πετρέλαιο [4].

Στη συνέχεια, καθώς οι κυβερνήσεις άρχισαν να χαλαρώνουν τα μέτρα περιορισμού και η οικονομία ανακάμπτει, η ζήτηση για υγρά καύσιμα αυξήθηκε σταδιακά. Ωστόσο, η ανάκαμψη της ζήτησης δεν ήταν ομοιόμορφη ανά τον κόσμο, με μερικές χώρες να επιστρέφουν σε κανονικά επίπεδα ζήτησης γρηγορότερα από άλλες. Η αβεβαιότητα σχετικά με τη διάρκεια της πανδημίας και την ταχύτητα της οικονομικής ανάκαμψης εξακολουθεί να επηρεάζει τις τιμές των καυσίμων.

1.1.3 Ελληνική αγορά υγρών καυσίμων

Η ελληνική αγορά υγρών καυσίμων είναι μια σημαντική και σύνθετη αγορά που περιλαμβάνει πολλούς παράγοντες, όπως την πρόσβαση σε πηγές πετρελαίου, τη διακίνηση και την επεξεργασία, καθώς και τη διανομή και πώληση στο κοινό. Τα κύρια είδη υγρών καυσίμων που καταναλώνονται στην Ελλάδα είναι το πετρέλαιο θέρμανσης, η αμόλυβδη βενζίνη και το πετρέλαιο κίνησης.

Η ελληνική αγορά καυσίμων επηρεάζεται σε μεγάλο βαθμό από τις παγκόσμιες τιμές του πετρελαίου, τις οποίες καθορίζουν οι διεθνείς αγορές. Επιπλέον, οι φορολογικές πολιτικές της χώρας, όπως οι ειδικοί φόροι κατανάλωσης (ΕΦΚ) και ο ΦΠΑ, παίζουν σημαντικό ρόλο στον καθορισμό των τιμών στην αγορά. Η Ελληνική κυβέρνηση έχει επίσης θεσπίσει διάφορους κανονισμούς και προδιαγραφές ποιότητας για τα καύσιμα, προκειμένου να εξασφαλίσει την προστασία του περιβάλλοντος και της δημόσιας υγείας.

Στην Ελλάδα, η ζήτηση για υγρά καύσιμα κυμαίνεται ανάλογα με την εποχή του έτους, την οικονομική κατάσταση και τις πολιτικές αποφάσεις. Ο τουρισμός είναι ένας από τους κύριους παράγοντες που επηρεάζουν τη ζήτηση για καύσιμα, καθώς η Ελλάδα είναι ένας δημοφιλής τουριστικός προορισμός. Κατά τους καλοκαιρινούς μήνες, η ζήτηση για βενζίνη και πετρέλαιο κίνησης αυξάνεται λόγω του μεγαλύτερου αριθμού τουριστών που επισκέπτονται τη χώρα.

Η ανάκαμψη της ελληνικής οικονομίας μετά τη χρηματοπιστωτική κρίση της προηγούμενης δεκαετίας έχει επίσης συμβάλει στην αύξηση της ζήτησης για καύσιμα. Εντούτοις, οι τιμές των καυσίμων παραμένουν υψηλές, καθώς η Ελλάδα εξακολουθεί να αντιμετωπίζει οικονομικές προκλήσεις και περιορισμούς.

Όσον αφορά την πανδημία του COVID-19, η ελληνική αγορά καυσίμων επηρεάστηκε όπως και οι άλλες αγορές σε όλο τον κόσμο. Κατά τη διάρκεια της πανδημίας, η ζήτηση για καύσιμα έπεσε αισθητά λόγω των περιοριστικών μέτρων που εφαρμόστηκαν από την κυβέρνηση, όπως το lockdown και η περιορισμένη κυκλοφορία. Αυτό οδήγησε σε μειωμένη κατανάλωση καυσίμων από τόσο τον ιδιωτικό όσο και τον επαγγελματικό τομέα.

Ωστόσο, καθώς οι περιορισμοί άρχισαν να χαλαρώνουν και η οικονομία άρχισε να ανακάμπτει, η ζήτηση για υγρά καύσιμα στην Ελληνική αγορά αυξήθηκε σταδιακά. Παρόλα αυτά, η ανάκαμψη δεν ήταν ομοιόμορφη, καθώς ορισμένοι τομείς, όπως ο τουρισμός, επηρεάστηκαν σε μεγαλύτερο βαθμό από την πανδημία και χρειάστηκε περισσότερος χρόνος για να ανακάμψουν.

1.1.4 Προσαρμογή της Ελλάδας στις μεταβαλλόμενες παγκόσμιες συνθήκες

Η Ελλάδα, όπως και πολλές άλλες χώρες, αντιμετωπίζει την ανάγκη να προσαρμοστεί στις συνεχώς μεταβαλλόμενες παγκόσμιες συνθήκες, που περιλαμβάνουν τόσο οικονομικές όσο και περιβαλλοντικές προκλήσεις. Η ανάλυση που ακολουθεί εξετάζει τις προσπάθειες της χώρας για προσαρμογή και τις προοπτικές της σε αυτό το πλαίσιο.

Αρχικά, στον οικονομικό τομέα, η Ελλάδα αναπτύσσει στρατηγικές για τη διασφάλιση της αειφορίας της ενέργειας και της ενεργειακής αυτόρκειας. Μετά την κρίση του χρέους και την εφαρμογή των μνημονίων, η χώρα αναγκάστηκε να επαναπροσδιορίσει τις προτεραιότητές της και να επικεντρωθεί στη βελτίωση της ανταγωνιστικότητας και της παραγωγικότητας. Η Ελλάδα επιδιώκει επίσης να διευρύνει τις εξαγωγές και να προωθήσει την ιδιωτική επένδυση, προκειμένου να δημιουργήσει νέες θέσεις εργασίας και να ανακάμψει από την οικονομική ύφεση.

Στο περιβαλλοντικό επίπεδο, η Ελλάδα αναγνωρίζει τη σημασία της μετάβασης σε μια πιο πράσινη οικονομία και της αντιμετώπισης των κλιματικών αλλαγών. Επομένως, έχουν ληφθεί μέτρα για την ενθάρρυνση της χρήσης ανανεώσιμων πηγών ενέργειας, τη βελτίωση της ενεργειακής απόδοσης και τη μείωση των εκπομπών αερίων θερμοκηπίου. Επιπλέον, η κυβέρνηση εργάζεται για την ανάπτυξη πράσινων υποδομών και την ενίσχυση της κυκλοφορίας ηλεκτρικών οχημάτων.

Η προσαρμογή στις μεταβαλλόμενες παγκόσμιες συνθήκες περιλαμβάνει επίσης την αντιμετώπιση των προκλήσεων που παρουσιάζονται από την τεχνολογική εξέλιξη. Η Ελλάδα προσπαθεί να αξιοποιήσει τις τεχνολογικές καινοτομίες για να βελτιώσει την παραγωγικότητα, την εκπαίδευση, την ιατρική περίθαλψη και τις υπηρεσίες κοινωνικής πρόνοιας. Επιπρόσθετα, η χώρα υποστηρίζει ενεργά την επέκταση της ψηφιακής οικονομίας, καθώς και την ολοκληρωμένη επένδυση σε πληροφοριακά και επικοινωνιακά συστήματα, προς όφελος της καθημερινής εμπειρίας των πολιτών και της λειτουργικότητας των επιχειρήσεων. Αυτό θα συμβάλει στη βελτίωση της ποιότητας ζωής και της ανταγωνιστικότητας της χώρας στην παγκόσμια αγορά.

1.2 Κίνητρο

Η απόφαση για την ασχολία με την συγκεκριμένη πτυχιακή εργασία προήλθε από την επιθυμία εμβάθυνσης στον τομέα της ανάλυσης χρονοσειρών και της πρόβλεψης τιμών, ιδίως στον κλάδο των υγρών καυσίμων. Σημαντικές πτυχές της κοινωνίας και της παγκόσμιας οικονομίας αποτελούν η οικονομική ανάπτυξη και η βιώσιμη ενέργεια, ενώ οι τιμές των καυσίμων επηρεάζουν σημαντικά τη ζωή των ανθρώπων, τις επιχειρήσεις και τη βιομηχανία.

Αναγνωρίζοντας τη σημασία των τιμών των καυσίμων, υπήρξε επιθυμία για περαιτέρω εξερεύνηση των μεθόδων πρόβλεψης τιμών και κατανόησης του πώς μπορούν να εφαρμοστούν αποτελεσματικά σε πρακτικό επίπεδο. Ένα από τα κύρια κίνητρα για αυτήν την επιλογή ήταν η επιθυμία να αξιοποιηθούν οι γνώσεις και οι δεξιότητες που αποκτήθηκαν κατά τη διάρκεια των σπουδών, σε ένα πεδίο με μεγάλη σημασία και επιρροή στην καθημερινή ζωή και την ανάπτυξη της χώρας. Υπήρξε επίσης η επιθυμία να συμβάλει στη βελτίωση της πρόσβασης σε πληροφορίες και την ενημέρωση των καταναλωτών, ώστε να λαμβάνουν πιο ενημερωμένες αποφάσεις σχετικά με την αγορά και τη χρήση καυσίμων.

Επιπρόσθετα, υπήρξε ενδιαφέρον να διερευνηθούν οι επιπτώσεις των περιβαλλοντικών παραγόντων και των πολιτικών αποφάσεων στις τιμές των καυσίμων, με σκοπό την ανάπτυξη μιας πιο ολοκληρωμένης κατανόησης των διαδικασιών και των μηχανισμών που διαμορφώνουν την αγορά των υγρών καυσίμων. Αυτό θα μπορούσε να συμβάλει στη διαμόρφωση πιο αποτελεσματικών πολιτικών και στρατηγικών για την αντιμετώπιση των προκλήσεων της ενεργειακής ασφάλειας, της περιβαλλοντικής προστασίας και της βιώσιμης ανάπτυξης.

Ένα επιπλέον κίνητρο για την επιλογή του συγκεκριμένου θέματος για την πτυχιακή εργασία προέκυψε από την προοπτική της εφαρμογής των αποτελεσμάτων της έρευνας σε πρακτικές εφαρμογές, όπως η δημιουργία ενός ιστοχώρου ή μιας εφαρμογής, που θα προσφέρει στους χρήστες εύκολη πρόσβαση στις χρονοσειρές των τιμών ανά νομό και τύπο καυσίμου και τις προβλέψεις για τις τιμές των υγρών καυσίμων.

Η ανάπτυξη μιας τέτοιας εφαρμογής θα μπορούσε να βοηθήσει τους καταναλωτές να επωφεληθούν από τις προβλέψεις των τιμών και να λάβουν πιο ενημερωμένες αποφάσεις, ενώ ταυτόχρονα θα συμβάλει στη διάδοση της γνώσης και της κατανόησης των μεθόδων ανάλυσης χρονοσειρών και των επιπτώσεων των διάφορων παραγόντων στη διαμόρφωση των τιμών των καυσίμων.

Η πτυχιακή αυτή εργασία, αναμένεται να συνεισφέρει στη βελτίωση της κατανόησης των παραγόντων που επηρεάζουν τις τιμές των υγρών καυσίμων και των μεθόδων πρόβλεψης τιμών, ώστε να υποστηρίξει τη λήψη πιο αποτελεσματικών αποφάσεων από τους καταναλωτές, τις επιχειρήσεις και τους κυβερνητικούς οργανισμούς. Ο στόχος είναι η δημιουργία ενός χρήσιμου εργαλείου που θα βοηθήσει στην αντιμετώπιση των προκλήσεων που αντιμετωπίζουν οι κοινωνίες και οι οικονομίες από τις διακυμάνσεις των τιμών των καυσίμων, την ενεργειακή ασφάλεια, την περιβαλλοντική προστασία και την βιώσιμη ανάπτυξη.

Τέλος, η ανάπτυξη των δεξιοτήτων στην ανάλυση δεδομένων και την επεξεργασία χρονοσειρών αποτελεί επίσης σημαντικό στόχο, καθώς οι ικανότητες αυτές θα είναι πολύτιμες για την επαγγελματική εξέλιξη.

1.3 Συνεισφορά

Αυτή η πτυχιακή εργασία αποτελεί μια προσπάθεια για την συνεισφορά στην κατανόηση και την πρόβλεψη των τιμών των υγρών καυσίμων στην Ελλάδα. Σημαντικός χρόνος και πόροι έχουν διατεθεί για την ανάλυση των δεδομένων και την αξιοποίηση των μεθόδων ανάλυσης χρονοσειρών. Η κύρια συνεισφορά στην πτυχιακή αυτή εργασία είναι η εφαρμογή των αποκτηθέντων γνώσεων και δεξιοτήτων στην επίλυση ενός πραγματικού προβλήματος που αφορά την πρόβλεψη των τιμών των καυσίμων.

Έχει γίνει επεξεργασία των διαθέσιμων δεδομένων, αφαιρώντας τα από θορύβους και ανεπιθύμητες παραμορφώσεις, ενώ έχουν επιλεγεί κατάλληλες μέθοδοι ανάλυσης χρονοσειρών για την πρόβλεψη των μελλοντικών τιμών. Επιπλέον, έχει σχεδιαστεί και αναπτυχθεί μια εφαρμογή ιστού, η οποία παρουσιάζει με σαφή και ευανάγνωστο τρόπο σε μορφή γραφήματος τις τιμές των υγρών καυσίμων ανά νομό και ανά τύπο καυσίμου, τις π

ροβλέψεις και τις χρονοσειρές των τιμών αυτών, καθώς και την αξιολόγηση των μεθόδων που χρησιμοποιήθηκαν για την πρόβλεψη των μελλοντικών τιμών. Μέσω της εφαρμογής αυτής, γίνονται πιο προσιτά τα αποτελέσματα της έρευνας στο ευρύ κοινό και στους ενδιαφερόμενους φορείς.

Τέλος, η ενημέρωση για τις πιο πρόσφατες τάσεις και εξελίξεις στον τομέα της ανάλυσης χρονοσειρών, καθώς και η επαφή με τη βιβλιογραφία, έχει συμβάλλει στον εμπλουτισμό των γνώσεων και την κατανόηση των εφαρμογών της ανάλυσης χρονοσειρών σε διάφορους τομείς. Αυτό έχει επιτρέψει την προσαρμογή και βελτίωση των μεθόδων που χρησιμοποιήθηκαν στην πτυχιακή εργασία, καθιστώντας την ακόμα πιο αποτελεσματική και αξιόπιστη.

1.4 Οργάνωση

Στο κεφάλαιο 2, αναλύουμε τη διαδικασία συλλογής πρωτογενών δεδομένων από διάφορες πηγές, την προ-επεξεργασία και καθαρισμό των δεδομένων, καθώς και τη δημιουργία της σχεσιακής βάσης δεδομένων για την αποθήκευση, διαχείριση και ανάλυση των τιμών των υγρών καυσίμων. Επιπλέον, παρουσιάζουμε τεχνικές και εργαλεία που χρησιμοποιήθηκαν για τη διαχείριση των δεδομένων και τη βελτίωση της απόδοσης των ερωτημάτων.

Στο κεφάλαιο 3 εστιάζουμε στην ανάλυση χρονοσειρών, μια σημαντική μέθοδος στη στατιστική και την επεξεργασία δεδομένων που επιτρέπει την παρακολούθηση και την πρόβλεψη των τάσεων των δεδομένων με την πάροδο του χρόνου. Αρχικά, εξηγούμε τις βασικές έννοιες και τη θεωρία πίσω από τις χρονοσειρές, καθώς και τις διάφορες μεθόδους που χρησιμοποιούνται για την ανάλυση και την πρόβλεψή τους. Στη συνέχεια, παρουσιάζουμε το μοντέλο ARIMA (Autoregressive Integrated Moving Average), μια δημοφιλή και αποτελεσματική τεχνική για την πρόβλεψη χρονοσειρών, που συνδυάζει την αυτοπαλινδρόμηση (autoregressive - AR), την ολοκλήρωση (integrated - I) και τη μεταβλητή κινητή μέση τιμή (moving average - MA). Περιγράφουμε λεπτομερώς το μαθηματικό υπόβαθρο του μοντέλου ARIMA, τις διάφορες παραμέτρους που εμπλέκονται και πώς επιλέγονται για να βελτιστοποιήσουν την πρόβλεψη.

Στο κεφάλαιο 4, αναλύουμε τις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής FuelPricesApp, τόσο στο Front end (HTML5, CSS, JavaScript και βιβλιοθήκες της), όσο και στο Back end (Python, FastAPI, MySQL). Εξηγούμε τη λειτουργία και τα πλεονεκτήματα κάθε τεχνολογίας, καθώς και τον τρόπο με τον οποίο συνεργάζονται για την παροχή των υπηρεσιών της εφαρμογής.

Στο κεφάλαιο 5, υλοποιούμε την προ-επεξεργασία των δεδομένων και εφαρμόζουμε το μοντέλο ARIMA στην Python καθώς διεξάγουμε και μια πειραματική μελέτη για τον καθορισμό των βέλτιστων παραμέτρων του μοντέλου. Παρουσιάζουμε τα αποτελέσματα της εφαρμογής του μοντέλου στα δεδομένα των τιμών των υγρών καυσίμων, καθώς και τη διαδικασία βελτιστοποίησης των παραμέτρων και της αξιολόγησης της απόδοσης του μοντέλου.

Στο κεφάλαιο 6, παρουσιάζουμε την υλοποίηση του FuelPricesApp, με επικέντρωση στην αρχιτεκτονική της εφαρμογής, το Back end και το Front end. Εξηγούμε τη δομή και τις λειτουργίες της εφαρμογής, καθώς και τον τρόπο με τον οποίο οι τεχνολογίες που αναφέρθηκαν στο Κεφάλαιο 4 ενσωματώνονται για την υλοποίησή της. Επιπλέον, περιγράφουμε τη διαδικασία ανάπτυξης και τα βήματα που ακολουθήθηκαν για την υλοποίηση της εφαρμογής.

Στο κεφάλαιο 7, παρουσιάζουμε το FuelPricesApp μέσω διαφορετικών case studies, αναλύοντας την παρουσίαση των τιμών υγρών καυσίμων και την πρόβλεψη των μελλοντικών τιμών με τη βοήθεια του μοντέλου ARIMA. Επικεντρωνόμαστε στα αποτελέσματα που παρέχει η εφαρμογή και στη χρηστικότητα της για τους χρήστες που ενδιαφέρονται για τις τάσεις των τιμών των υγρών καυσίμων.

Τέλος, στο κεφάλαιο 8, παρουσιάζουμε τα συμπεράσματα που προκύπτουν από την πτυχιακή εργασία και συζητάμε τη μελλοντική εργασία που θα μπορούσε να επεκτείνει και να βελτιώσει το FuelPricesApp. Εξετάζουμε τα κύρια ευρήματα της εργασίας, τις επιτυχίες και τις προκλήσεις που αντιμετωπίστηκαν κατά τη διάρκεια της υλοποίησης του έργου, καθώς και τις πιθανές βελτιώσεις που θα μπορούσαν να γίνουν για να αυξήσουν την αποτελεσματικότητα και τη χρηστικότητα της εφαρμογής. Προτείνουμε επίσης πιθανές κατευθύνσεις για μελλοντική έρευνα, όπως η εξέταση εναλλακτικών μεθόδων πρόβλεψης, η ανάπτυξη πρόσθετων λειτουργιών για το FuelPricesApp ή η εφαρμογή των προσεγγίσεων αυτής της εργασίας σε άλλους τομείς.

Συνοψίζοντας, η πτυχιακή εργασία παρέχει μια λεπτομερή ανάλυση του προβλήματος της πρόβλεψης των τιμών των υγρών καυσίμων και της υλοποίησης μιας εφαρμογής που αντιμετωπίζει αυτό το πρόβλημα. Μέσω της εξέτασης των τεχνολογιών, της ανάπτυξης ενός μοντέλου πρόβλεψης και της υλοποίησης του FuelPricesApp, δίνουμε στους χρήστες ένα εργαλείο για να παρακολουθούν και να προβλέπουν τις τιμές των καυσίμων, βοηθώντας τους να λαμβάνουν πιο ενημερωμένες αποφάσεις για την κατανάλωση καυσίμων. Η εργασία αυτή επικεντρώνεται στην εφαρμογή της ανάλυσης χρονοσειρών και του μοντέλου ARIMA για την πρόβλεψη των τιμών και εξετάζει τις επιδόσεις του μοντέλου μέσω πειραματικών μελετών.

Επιπλέον, η εργασία εξετάζει την αρχιτεκτονική και τις τεχνολογίες που χρησιμοποιούνται για την υλοποίηση του FuelPricesApp, όπως οι γλώσσες προγραμματισμού, τα πρότυπα σχεδίασης και οι βάσεις δεδομένων. Τέλος, παρουσιάζει περιπτώσεις χρήσης της εφαρμογής, προβάλλοντας τις δυνατότητες και τη χρησιμότητα του FuelPricesApp για τους τελικούς χρήστες.

Κεφάλαιο 2ο: Το σύνολο δεδομένων

2.1 Πρωτογενή δεδομένα

Τα πρωτογενή δεδομένα αποτελούν ένα θεμελιώδες στοιχείο για την κατανόηση και την ανάλυση των τιμών των υγρών καυσίμων, παρέχοντας τη βάση για την ανάπτυξη μοντέλων πρόβλεψης και την παρακολούθηση των τάσεων της αγοράς. Για την πτυχιακή αυτή, τα δεδομένα λαμβάνονται καθημερινά από το Υπουργείο Ανάπτυξης και Επενδύσεων στην ιστοσελίδα τους <http://www.fuelprices.gr/> σε μορφή PDF, και περιλαμβάνουν ένα δελτίο με τις τιμές των υγρών καυσίμων ανά νομό της Ελληνικής επικράτειας.

Στα πλαίσια μιας άλλης διπλωματικής εργασίας, έχει αναπτυχθεί ένας εξελιγμένος μηχανισμός για την αυτόματη ανάκτηση και επεξεργασία αυτών των δεδομένων. Μέσω αυτού του μηχανισμού, τα δεδομένα μετατρέπονται από τη μορφή PDF σε μια πιο εύχρηστη μορφή και αποθηκεύονται σε μια σχεσιακή βάση δεδομένων. Αυτή η διαδικασία επιτρέπει την ταχύτερη και πιο αποτελεσματική ανάλυση των δεδομένων.

Για τις ανάγκες της παρούσας πτυχιακής εργασίας, έχουμε πρόσβαση σε αυτήν τη σχεσιακή βάση δεδομένων και ανακτούμε τον πίνακα που περιέχει τις καθημερινές τιμές των υγρών καυσίμων ανά νομό της Ελλάδας και ανά τύπο καυσίμου. Αυτό το σύνολο δεδομένων αποτελεί τη βάση για την ανάλυση και την εξαγωγή συμπερασμάτων σχετικά με τη δυναμική των τιμών των καυσίμων και την επίδραση διαφόρων παραγόντων, όπως η γεωγραφική θέση και οι εποχιακές τάσεις.

Η ανάκτηση και επεξεργασία αυτών των δεδομένων αποτελεί ένα κρίσιμο βήμα για την κατανόηση του προβλήματος και την ανάπτυξη ενός αξιόπιστου μοντέλου πρόβλεψης των τιμών των καυσίμων. Επίσης, η διαθεσιμότητα αυτών των δεδομένων επιτρέπει την αξιολόγηση της απόδοσης του μοντέλου και την προσαρμογή του σε περιπτώσεις όπου απαιτούνται βελτιώσεις.



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΑΝΑΠΤΥΞΗΣ ΚΑΙ ΕΠΕΝΔΥΣΕΩΝ

Αθήνα, 05-05-2023

**Διυπηρεσιακή Μονάδα Ελέγχου Αγοράς
Διεύθυνση Διαχείρισης Δεδομένων Στατιστικής
Επεξεργασίας & Εποπτείας Ηλεκτρονικού
Εμπορίου
Τμήμα Διαχείρισης Πληροφοριών & Εποπτείας
Ηλεκτρονικού Εμπορίου**

Ταχ. Δ/ση : Πλ. Κάνιγγος, Αθήνα ΠΡΟΣ : ΑΝΑΡΤΗΤΕΟ ΣΤΟ FUELPRICES
Ταχ. Κώδικας : 106 77
Πληροφορίες : Νικολοπούλου Μαρία
Τηλέφωνο : 213 1514 284
Fax : 210 3815 317
Email : fuelprices@mindev.gov.gr
Ιστοσελίδα : www.mindev.gov.gr

ΘΕΜΑ: ΚΑΘΗΜΕΡΙΝΟ ΔΕΛΤΙΟ ΕΠΙΣΚΟΠΗΣΗΣ ΤΙΜΩΝ ΥΓΡΩΝ ΚΑΥΣΙΜΩΝ

Μέσες τιμές λιανικής πώλησης καυσίμων ανά Νομό, για την
Πέμπτη, 4 Μαΐου 2023
(τιμές σε €/λτ, συμπ. ΦΠΑ)

ΝΟΜΟΣ	Αμόλυβδη 95 οκτ.	Αμόλυβδη 100 οκτ.	Diesel Κίνησης	Υγραέριο κίνησης (Autogas)
ΝΟΜΟΣ ΑΤΤΙΚΗΣ	1,822	2,037	1,536	0,832
ΝΟΜΟΣ ΑΙΤΩΛΙΑΣ ΚΑΙ ΑΚΑΡΝΑΝΙΑΣ	1,913	2,123	1,615	1,023
ΝΟΜΟΣ ΑΡΓΟΛΙΔΟΣ	1,879	2,073	1,580	0,909
ΝΟΜΟΣ ΑΡΚΑΔΙΑΣ	1,894	2,124	1,593	0,953
ΝΟΜΟΣ ΑΡΤΗΣ	1,883	2,087	1,584	0,983
ΝΟΜΟΣ ΑΧΑΪΑΣ	1,868	2,079	1,579	0,975
ΝΟΜΟΣ ΒΟΙΩΤΙΑΣ	1,871	2,093	1,575	0,968
ΝΟΜΟΣ ΓΡΕΒΕΝΩΝ	1,919	2,139	1,616	1,015
ΝΟΜΟΣ ΔΡΑΜΑΣ	1,856	2,064	1,558	0,920
ΝΟΜΟΣ ΔΩΔΕΚΑΝΗΣΟΥ	2,023	2,174	1,706	1,119
ΝΟΜΟΣ ΕΒΡΟΥ	1,881	2,059	1,589	1,005
ΝΟΜΟΣ ΕΥΒΟΙΑΣ	1,885	2,086	1,589	0,921
ΝΟΜΟΣ ΕΥΡΥΤΑΝΙΑΣ	1,947	2,188	1,656	0,968
ΝΟΜΟΣ ΖΑΚΥΝΘΟΥ	1,914	2,085	1,606	1,081
ΝΟΜΟΣ ΗΛΕΙΑΣ	1,885	2,085	1,582	0,920
ΝΟΜΟΣ ΗΜΑΘΙΑΣ	1,836	2,050	1,548	0,892
ΝΟΜΟΣ ΗΡΑΚΛΕΙΟΥ	1,929	2,084	1,627	1,008
ΝΟΜΟΣ ΘΕΣΠΡΩΤΙΑΣ	1,881	2,108	1,596	0,965
ΝΟΜΟΣ ΘΕΣΣΑΛΟΝΙΚΗΣ	1,818	2,031	1,532	0,870
ΝΟΜΟΣ ΙΩΑΝΝΙΝΩΝ	1,863	2,037	1,566	0,909
ΝΟΜΟΣ ΚΑΒΑΛΑΣ	1,881	2,102	1,582	0,894
ΝΟΜΟΣ ΚΑΡΔΙΤΣΗΣ	1,899	2,087	1,594	0,955
ΝΟΜΟΣ ΚΑΣΤΟΡΙΑΣ	1,882	2,077	1,593	0,946
ΝΟΜΟΣ ΚΕΡΚΥΡΑΣ	1,917	2,115	1,629	1,097
ΝΟΜΟΣ ΚΕΦΑΛΛΗΝΙΑΣ	1,986	2,178	1,694	1,090
ΝΟΜΟΣ ΚΙΛΚΙΣ	1,868	2,087	1,581	0,945
ΝΟΜΟΣ ΚΟΖΑΝΗΣ	1,857	2,044	1,571	0,879

Σχήμα 2.1: Παράδειγμα δελτίου με τις τιμές των υγρών καυσίμων ανά νομό της Ελληνικής επικράτειας

2.2 Προ-επεξεργασία δεδομένων

Στην παρούσα ενότητα αναφερόμαστε στην προ-επεξεργασία των δεδομένων, η οποία αποτελεί ένα ζωτικής σημασίας βήμα στην προετοιμασία τους για την ανάλυση και την μοντελοποίηση των τιμών των υγρών καυσίμων. Κατά την εξέταση των δεδομένων που λήφθηκαν από το Υπουργείο Ανάπτυξης και Επενδύσεων, διαπιστώθηκε ότι υπήρχαν σημαντικές ελλείψεις στις ημερομηνίες καταγραφής των τιμών, καθώς και περιπτώσεις όπου εμφανίζονταν μηδενικές τιμές για κάποια καύσιμα σε συγκεκριμένες ημερομηνίες.

Για την αντιμετώπιση αυτών των προβλημάτων και τη βελτίωση της ποιότητας των δεδομένων, επιλέχθηκε η μέθοδος της γραμμικής παρεμβολής (linear interpolation). Η γραμμική παρεμβολή είναι μια τεχνική εκτίμησης τιμών εντός ενός εύρους μεταξύ δύο γνωστών τιμών, χρησιμοποιώντας μια ευθεία γραμμή που ενώνει τις γνωστές τιμές. Χρησιμοποιείται ευρέως σε πολλούς τομείς, όπως την κλιματολογία, την οικονομετρία και την τεχνητή νοημοσύνη για την εξομάλυνση των δεδομένων. Μαθηματικά, η γραμμική παρεμβολή μπορεί να περιγραφεί ως:

$$y = y_1 + \frac{(x - x_1) \cdot (y_2 - y_1)}{x_2 - x_1}$$

Επιλέχθηκε η γραμμική παρεμβολή καθώς αποτελεί μια απλή, αλλά ταυτόχρονα αποτελεσματική μέθοδος για την αντιμετώπιση ελλείψεων και ασυνεχειών στα δεδομένα. Επιπλέον, αυτή η μέθοδος είναι κατάλληλη για την περίπτωση των τιμών των καυσίμων, καθώς μπορεί να χρησιμοποιηθεί για να εκτιμηθούν οι τιμές των καυσίμων μεταξύ γνωστών ημερομηνιών, λαμβάνοντας υπόψη την τάση των τιμών στις γνωστές ημερομηνίες.

Με την εφαρμογή της γραμμικής παρεμβολής στα δεδομένα, προκύπτει ένα ικανοποιητικό σύνολο, το οποίο είναι πλήρες και συνεχές. Τα δεδομένα αυτά μπορούν πλέον να χρησιμοποιηθούν για την περαιτέρω ανάλυση, καθώς και για τη δημιουργία του σχεσιακού μοντέλου που θα χρησιμεύσει για την πρόβλεψη και την ανάλυση των τιμών των υγρών καυσίμων στο μέλλον. Η προ-επεξεργασία των δεδομένων και η επιλογή της κατάλληλης μεθόδου γραμμικής παρεμβολής παίζει ίσως τον πιο βασικό ρόλο για την ακρίβεια και την αξιοπιστία των επόμενων βημάτων της ανάλυσης και της μοντελοποίησης.

Μετά την επιτυχή εφαρμογή της γραμμικής παρεμβολής και την αντιμετώπιση των ελλείψεων και των ασυνεχειών στα δεδομένα, οι επόμενοι στόχοι περιλαμβάνουν την εξερεύνηση των δεδομένων, την ανακάλυψη προτύπων και σχέσεων, καθώς και τη δημιουργία και την εκπαίδευση ενός σχεσιακού μοντέλου, το οποίο θα μπορούσε να προβλέπει τις τιμές των υγρών καυσίμων στο μέλλον με βάση τις τάσεις και τις παραμέτρους που παρουσιάζονται στα δεδομένα.

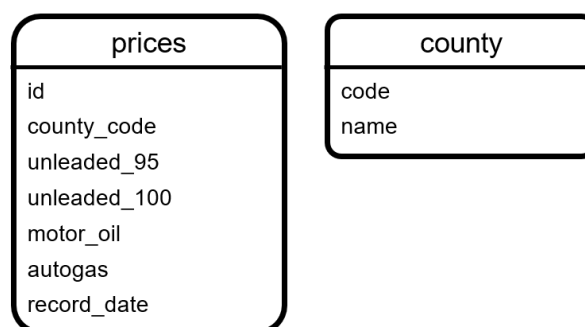
Συνοψίζοντας, η προ-επεξεργασία των δεδομένων αποτελεί ένα κρίσιμο στάδιο στη διαδικασία της ανάλυσης των τιμών των υγρών καυσίμων. Η εφαρμογή της γραμμικής παρεμβολής επέτρεψε τη βελτίωση της ποιότητας των δεδομένων, καθιστώντας τα ικανοποιητικά για την περαιτέρω ανάλυση και τη δημιουργία ενός αξιόπιστου σχεσιακού μοντέλου. Η αύξηση της ακρίβειας και της πληρότητας των δεδομένων βοηθά στην αποτελεσματικότερη εκτίμηση των παραμέτρων του μοντέλου, καθώς και στη βελτίωση των προβλέψεων για τις τιμές των υγρών καυσίμων στο μέλλον.

2.3 Δημιουργία σχεσιακής Βάσης Δεδομένων

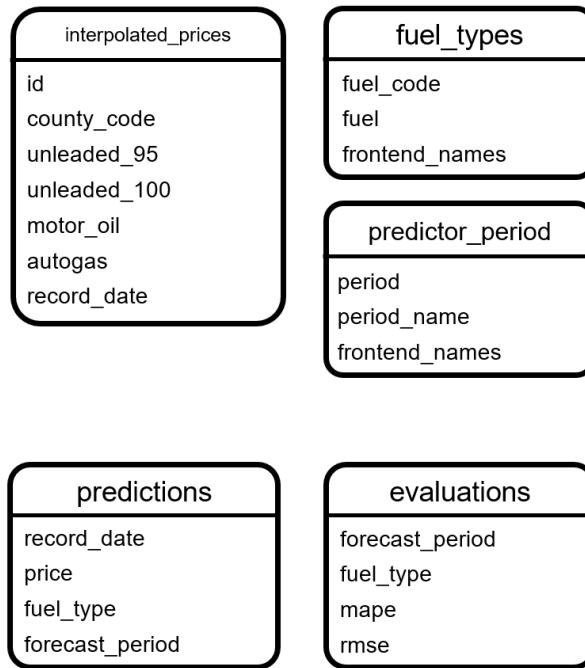
Στο πλαίσιο της δημιουργίας μιας σχεσιακής βάσης δεδομένων για την αποθήκευση και διαχείριση των δεδομένων που αφορούν τις τιμές των υγρών καυσίμων, εξετάζουμε τα σχεσιακά μοντέλα δεδομένων. Αυτά τα μοντέλα αποτελούν μια δημοφιλή προσέγγιση στην οργάνωση δεδομένων, βασιζόμενη στη χρήση πινάκων για την αποθήκευση των πληροφοριών.

Για την υλοποίηση της βάσης δεδομένων, χρησιμοποιήσαμε την MySQL, ένα δημοφιλές σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) ανοιχτού κώδικα. Η MySQL προσφέρει ευελιξία και αποτελεσματικότητα στη διαχείριση των δεδομένων, καθιστώντας την ιδανική επιλογή για την εφαρμογή μας.

Όπως έχει ήδη αναφερθεί, από άλλη διπλωματική εργασία έχουμε πρόσβαση σε δύο πίνακες ενός σχεσιακού μοντέλου, οι οποίοι είναι ο `prices` και ο `county`. Ο πίνακας `prices` διαθέτει όλα τα απαραίτητα δεδομένα που χρειαζόμαστε για την ανάλυση και πρόβλεψη, τα οποία είναι οι καθημερινές τιμές των υγρών καυσίμων ανά τύπο καυσίμου και ανά νομό της Ελλάδας. Ο πίνακας `county` περιέχει έναν μοναδικό κωδικό και την ονομασία για κάθε νομό της Ελλάδας. Για τις ανάγκες της διαδικτυακής εφαρμογής δημιουργήσαμε δύο επιπλέον πίνακες, τον `fuel_types` και `predictor_period`. Μετά την εφαρμογή της διαδικασίας γραμμικής παρεμβολής (linear interpolation) για τη συμπλήρωση των missing values, δημιουργήσαμε έναν νέο πίνακα, τον `interpolated_prices`, ο οποίος έχει ακριβώς τις ίδιες στήλες με τον πίνακα `prices` αλλά πλέον περιλαμβάνει τα ενημερωμένα δεδομένα των τιμών. Τέλος, μετά από την υλοποίηση του μοντέλου ARIMA, δημιουργήσαμε άλλους δύο πίνακες: τον `predictions` και τον `evaluations`. Ο πίνακας `predictions` αποθηκεύει τις προβλέψεις για τις τιμές των καυσίμων στην Ελλάδα ανά τύπο καυσίμου και ανά περίοδο πρόβλεψης, ενώ ο πίνακας `evaluations` αποθηκεύει τις αξιολογήσεις των προβλέψεων ανά τύπο καυσίμου και ανά περίοδο πρόβλεψης.



Σχήμα 2.2: Πίνακες σχεσιακού μοντέλου από άλλη διπλωματική εργασία



Σχήμα 2.3: Πίνακες σχεσιακού μοντέλου για την παρούσα πτυχιακή εργασία

Κεφάλαιο 3ο: Ανάλυση χρονοσειρών και το μοντέλο ARIMA

3.1 Χρονοσειρές

Η έννοια των χρονοσειρών εξετάζεται λεπτομερώς, καθώς αποτελεί έναν τύπο δεδομένων που περιγράφει την εξέλιξη μιας μεταβλητής κατά τη διάρκεια του χρόνου. Οι χρονοσειρές είναι ευρέως χρησιμοποιούμενες σε πολλούς κλάδους, όπως η οικονομία, η μετεωρολογία, η ιατρική, η κοινωνική επιστήμη και πολλούς άλλους.

Οι χρονοσειρές ορίζονται ως μια συλλογή παρατηρήσεων μιας μεταβλητής, που λαμβάνονται σε τακτά χρονικά διαστήματα. [5]. Αυτές οι παρατηρήσεις μπορούν να αφορούν τιμές, μετρήσεις ή άλλους δείκτες που εξετάζονται στο πλαίσιο του χρόνου. Η ανάλυση των χρονοσειρών επιτρέπει την ανίχνευση τάσεων, κυκλικών συμπεριφορών και άλλων δομικών στοιχείων που χαρακτηρίζουν τη συμπεριφορά των δεδομένων κατά τη διάρκεια του χρόνου.

Οι χρονοσειρές είναι καίριας σημασίας σε πολλές εφαρμογές, καθώς μας επιτρέπουν να κατανοήσουμε τη δυναμική εξέλιξη ενός φαινομένου και να λάβουμε αποφάσεις βασισμένες σε προβλέψεις. Για παράδειγμα, στον τομέα της οικονομίας, η ανάλυση χρονοσειρών χρησιμοποιείται για την πρόβλεψη των μελλοντικών τιμών των μετοχών, των ισοτιμιών των νομισμάτων, των επιτοκίων και των οικονομικών δεικτών. Στη μετεωρολογία, οι χρονοσειρές χρησιμοποιούνται για την πρόβλεψη των καιρικών συνθηκών και της κλιματικής εξέλιξης. Στην ιατρική, οι χρονοσειρές μπορούν να χρησιμοποιηθούν για την παρακολούθηση της προόδου μιας ασθένειας ή της αποτελεσματικότητας μιας θεραπείας κατά το χρόνο. Στις κοινωνικές επιστήμες, η ανάλυση χρονοσειρών επιτρέπει την παρακολούθηση της εξέλιξης κοινωνικών φαινομένων, όπως η ανεργία, η φτώχεια και η μετανάστευση.

Η ανάλυση χρονοσειρών δεν είναι απλή και απαιτεί κατανόηση των στατιστικών μεθόδων, των τεχνικών ανάλυσης και των υπολογιστικών εργαλείων. Επιπλέον, είναι σημαντικό να ληφθούν υπόψη οι ιδιαιτερότητες κάθε χρονοσειράς, όπως η σταθερότητα, η τάση, η εποχικότητα και οι αυτοσυσχετίσεις, καθώς αυτές μπορούν να επηρεάσουν την ακρίβεια και την αξιοπιστία των αποτελεσμάτων.

3.2 Μέθοδοι ανάλυσης χρονοσειρών

Η ανάλυση χρονοσειρών είναι μια καίρια στατιστική προσέγγιση που ασχολείται με την εξέταση και την πρόβλεψη δεδομένων που λαμβάνονται κατά τη διάρκεια του χρόνου. Κύριος στόχος αυτής της ανάλυσης είναι η κατανόηση των χαρακτηριστικών, των μοτίβων και των αλληλεπιδράσεων που παρουσιάζονται στα δεδομένα, με σκοπό την πρόβλεψη μελλοντικών τιμών ή την εξαγωγή κρυφής πληροφορίας. Η ανάλυση χρονοσειρών περιλαμβάνει μια πληθώρα μεθόδων που μπορούν να καταταχθούν σε διάφορες κατηγορίες ανάλογα με το είδος των δεδομένων, τη φύση του προβλήματος που αντιμετωπίζουν, τις υποθέσεις που γίνονται και τους στόχους της ανάλυσης. Μεθόδοι όπως η Εκθετική Εξομάλυνση, η Autoregressive Integrated Moving Average (ARIMA), και η Seasonal Decomposition of Time Series (STL) είναι μερικές από τις πιο δημοφιλείς τεχνικές που χρησιμοποιούνται στην πρακτική [6].

- *Εκθετική Εξομάλυνση*: Η εκθετική εξομάλυνση είναι μια τεχνική πρόβλεψης που βασίζεται στον υπολογισμό ενός μέσου ορόυ των προηγούμενων παρατηρήσεων, με μεγαλύτερο βάρος στις πιο πρόσφατες παρατηρήσεις. Η μέθοδος αυτή είναι κατάλληλη για την πρόβλεψη μιας χρονοσειράς χωρίς σαφή τάση ή κυκλικότητα [7].
- *Autoregressive Integrated Moving Average (ARIMA)*: Η μέθοδος ARIMA είναι μια γενικευμένη τεχνική πρόβλεψης που μπορεί να αντιμετωπίσει χρονοσειρές με τάσεις και κυκλικότητα, χρησιμοποιώντας μια συνδυασμένη προσέγγιση αυτοπαλίνδρομης (AR), κινούμενης μέσης τιμής (MA) και ολοκλήρωσης (I). Η μέθοδος αυτή επιτρέπει την προσαρμογή των παραμέτρων του μοντέλου για να ταιριάζουν καλύτερα στα χαρακτηριστικά της χρονοσειράς και να προβλέπει μελλοντικές τιμές με βάση τη δομή και τη συμπεριφορά των παρατηρήσεων [8].
- *Seasonal Decomposition of Time Series (STL)*: Η μέθοδος STL είναι μια τεχνική αποσύνθεσης που επιτρέπει την απομάκρυνση των κύριων συνιστωσών μιας χρονοσειράς, όπως η τάση, η εποχικότητα και το τυχαίο στοιχείο. Αυτό επιτυγχάνεται μέσω της ανάλυσης των συνιστωσών σε συνθετικές συναρτήσεις και της ανακατασκευής της χρονοσειράς από τις βασικές συνιστώσες. Η μέθοδος STL είναι κατάλληλη για χρονοσειρές με πολύπλοκα μοτίβα και εποχικές διακυμάνσεις, καθώς επιτρέπει την εξερεύνηση και την ανάλυση των διαφόρων στοιχείων που επηρεάζουν τη συμπεριφορά της σειράς [9].

3.3 Πρόβλεψη μελλοντικών τιμών χρονοσειρών

Κατά τη διερεύνηση της πρόβλεψης μελλοντικών τιμών χρονοσειρών, θα ασχοληθούμε με την κατανόηση των χαρακτηριστικών και των μοτίβων που παρουσιάζονται σε μια σειρά δεδομένων που λαμβάνονται κατά τη διάρκεια του χρόνου. Όπως αναφέρθηκε στις προηγούμενες ενότητες, η ανάλυση χρονοσειρών είναι μια στατιστική προσέγγιση που ασχολείται με την κατανόηση των χαρακτηριστικών και των μοτίβων που παρουσιάζονται σε μια σειρά δεδομένων που λαμβάνονται κατά τη διάρκεια του χρόνου. Με τη χρήση των διάφορων μεθόδων ανάλυσης χρονοσειρών που περιγράφηκαν στην ενότητα 3.2, όπως η εκθετική εξομάλυνση, το μοντέλο ARIMA και η ανάλυση STL, είμαστε σε θέση να προβλέψουμε τις μελλοντικές τιμές μιας χρονοσειράς [10].

Στην προσπάθεια να προβλέψουμε μελλοντικές τιμές χρονοσειρών, πρέπει να ακολουθήσουμε μια σειρά βημάτων που μπορούν να ενισχύσουν την ακρίβεια και την αξιοπιστία των προβλέψεων [11]:

- Προεπεξεργασία δεδομένων: Τα δεδομένα πρέπει να καθαριστούν από θόρυβο, απουσιάζουσες τιμές και απροσδόκητες αποκλίσεις. Η εποχική εξομάλυνση και η αποσύνθεση μπορεί να βοηθήσουν στην εξάλειψη των εποχικών και τυχαίων συνιστωσών.
- Επιλογή μοντέλου: Με βάση τα χαρακτηριστικά της σειράς, επιλέγεται το καταλληλότερο μοντέλο πρόβλεψης, όπως το μοντέλο ARIMA, η εκθετική εξομάλυνση ή η ανάλυση STL.
- Εκπαίδευση και βελτιστοποίηση: Το επιλεγμένο μοντέλο πρέπει να εκπαιδευτεί με τα διαθέσιμα δεδομένα. Ενδέχεται να χρειαστεί να προσαρμοστούν παράμετροι για τη βελτιστοποίηση της απόδοσης του μοντέλου.

- Αξιολόγηση και επικύρωση: Χρησιμοποιώντας τεχνικές όπως η διασταυρωμένη επικύρωση και τη ρίζα μέσης τετραγωνικής απόκλισης (Root Mean Squared Error - RMSE), μπορούμε να αξιολογήσουμε την απόδοση του μοντέλου προβλέψεων. Αυτό μπορεί να βοηθήσει στην επιλογή του βέλτιστου μοντέλου και στη βελτίωση της ακρίβειας των προβλέψεων.

Κατά τη διαδικασία πρόβλεψης μελλοντικών τιμών χρονοσειρών, μπορεί να προκύψουν διάφορες δυσκολίες, όπως [12]:

- Επιλογή κατάλληλου μοντέλου: Ενδέχεται να υπάρχουν πολλά μοντέλα που ταιριάζουν στα δεδομένα, και η επιλογή του καλύτερου μπορεί να είναι δύσκολη.
- Μη-σταθερότητα των δεδομένων: Οι χρονοσειρές μπορεί να παρουσιάζουν αλλαγές στο χρόνο λόγω εξωτερικών παραγόντων. Αυτό μπορεί να καταστήσει τις προβλέψεις λιγότερο αξιόπιστες.
- Πολυπλοκότητα των δεδομένων: Οι χρονοσειρές μπορεί να περιλαμβάνουν πολλές συνιστώσες, όπως εποχικότητα, τάση και κύκλο, που μπορεί να καθιστούν την ανάλυση και την πρόβλεψη πιο πολύπλοκες. Η αναγνώριση και η απομόνωση αυτών των συνιστωσών είναι απαραίτητη για τη βελτίωση της ακρίβειας των προβλέψεων.
- Παραμετροποίηση μοντέλων: Ορισμένα μοντέλα απαιτούν την εκτίμηση παραμέτρων, που μπορεί να είναι δύσκολη και χρονοβόρα. Η χρήση βέλτιστων τεχνικών βελτιστοποίησης και επικύρωσης είναι απαραίτητη για την αποφυγή υπερεκπαίδευσης ή υποεκπαίδευσης του μοντέλου.

Για να αντιμετωπίσουμε αυτές τις δυσκολίες, μπορούμε να ακολουθήσουμε τις παρακάτω στρατηγικές:

- Επιλογή μοντέλου: Μπορούμε να εφαρμόσουμε διάφορα μοντέλα στα δεδομένα μας και να συγκρίνουμε την απόδοσή τους με βάση κάποιο κριτήριο αξιολόγησης, όπως το RMSE (Root Mean Squared Error), για να επιλέξουμε το καλύτερο.
- Εποχική προσαρμογή: Αν οι χρονοσειρές παρουσιάζουν εποχικότητα, μπορούμε να απομονώσουμε και να αφαιρέσουμε την εποχική συνιστώσα πριν από την εφαρμογή των μοντέλων πρόβλεψης.
- Σταθεροποίηση των δεδομένων: Στις περιπτώσεις που η χρονοσειρά δεν είναι σταθερή, μπορούμε να εφαρμόσουμε τεχνικές, όπως η διαφοροποίηση, για να καταστήσουμε τη σειρά σταθερή και να βελτιώσουμε την ακρίβεια των προβλέψεων.
- Επικύρωση μοντέλου: Χρησιμοποιώντας τεχνικές όπως η διασταυρούμενη επικύρωση, μπορούμε να αξιολογήσουμε την απόδοση του μοντέλου σε διαφορετικά υποσύνολα των δεδομένων, βοηθώντας μας να αποφύγουμε την υπερεκπαίδευση και να επιλέξουμε το κατάλληλο μοντέλο.

Για παράδειγμα, ας υποθέσουμε ότι θέλουμε να προβλέψουμε τις μελλοντικές τιμές των πωλήσεων ενός προϊόντος. Οι χρονοσειρές των πωλήσεων μπορεί να περιέχουν τάση, εποχικότητα και τυχαίες διακυμάνσεις. Θα μπορούσαμε να χρησιμοποιήσουμε την ανάλυση STL για να απομονώσουμε τις διαφορές συνιστώσες και να εφαρμόσουμε το μοντέλο ARIMA για να προβλέψουμε τις μελλοντικές τιμές. Κατόπιν, θα χρησιμοποιήσουμε τεχνικές επικύρωσης, όπως η διασταυρούμενη επικύρωση, για να αξιολογήσουμε την απόδοση του μοντέλου μας και να βελτιώσουμε την ακρίβεια των προβλέψεων.

Κατά τη διάρκεια αυτής της διαδικασίας, μπορεί να αντιμετωπίσουμε δυσκολίες, όπως η αστάθεια των δεδομένων ή η αλλαγή των μοτίβων. Σε αυτές τις περιπτώσεις, θα μπορούσαμε να εξετάσουμε τη χρήση επιπλέον τεχνικών, όπως η προσαρμογή των παραμέτρων του μοντέλου, η εφαρμογή της διαφοροποίησης για την απομάκρυνση των τάσεων ή η ενσωμάτωση εξωτερικών παραγόντων που μπορεί να επηρεάζουν τις πωλήσεις, όπως οι διαφημίσεις ή οι προωθητικές εκστρατείες.

Σε γενικές γραμμές, η προβλεψη μελλοντικών τιμών χρονοσειρών απαιτεί μια προσεκτική και συστηματική προσέγγιση που συνδυάζει την ανάλυση των χαρακτηριστικών των δεδομένων, την επιλογή του κατάλληλου μοντέλου και την επικύρωση της απόδοσης του μοντέλου. Επίσης, πρέπει να λάβουμε υπόψη τις πιθανές δυσκολίες και να χρησιμοποιήσουμε κατάλληλες τεχνικές για την αντιμετώπισή τους. Η προβλεψη μελλοντικών τιμών είναι μια διαδικασία που απαιτεί συνεχή βελτίωση, καθώς τα δεδομένα και οι συνθήκες εξελίσσονται με τον καιρό. Είναι επίσης σημαντικό να αξιολογούμε τακτικά την απόδοση των μοντέλων πρόβλεψης και να προσαρμόζουμε τις παραμέτρους τους, εάν χρειάζεται [13].

3.4 Το μοντέλο ARIMA

Μια από τις πιο δημοφιλείς τεχνικές για την ανάλυση και την πρόβλεψη χρονοσειρών, είναι η μέθοδος του μοντέλου ARIMA. Το ARIMA, που σημαίνει AutoRegressive Integrated Moving Average, είναι ένα συνδυασμός τριών βασικών συστατικών: Auto-regression (AR), Integrated (I) και Moving average (MA).

Η μέθοδος ARIMA χρησιμοποιείται ευρέως σε διάφορες εφαρμογές, όπως η οικονομική πρόβλεψη, η πρόβλεψη κυκλοφορίας και η πρόβλεψη ζήτησης προϊόντων. Η δύναμη του ARIMA έγκειται στην ικανότητά του να μοντελοποιεί συσχετίσεις χρονοσειρών και να κάνει προβλέψεις με βάση αυτές τις συσχετίσεις. Είναι κατάλληλη όταν η χρονοσειρά παρουσιάζει σταθερότητα, γραμμική συσχέτιση και είναι σταθμισμένη. Δεν είναι κατάλληλη για χρονοσειρές με μη γραμμικές συσχετίσεις ή πολύπλοκα μοτίβα [14].

Η μέθοδος ARIMA περιλαμβάνει τρεις παραμέτρους: p (auto-regression), d (integrated) και q (moving average). Η επιλογή των βέλτιστων παραμέτρων p , d και q βασίζεται στην ελαχιστοποίηση ενός κριτηρίου, όπως το Akaike Information Criterion (AIC) ή το Bayesian Information Criterion (BIC). Η διαδικασία επιλογής των παραμέτρων περιλαμβάνει την εκτίμηση πολλαπλών μοντέλων και την επιλογή εκείνου που παρέχει την καλύτερη επίδοση πρόβλεψης.

Επιλέγουμε να χρησιμοποιήσουμε τη μέθοδο ARIMA στην πτυχιακή μας εργασία, καθώς είναι μια καλά κατασταλαγμένη και δοκιμασμένη τεχνική που έχει αποδειχθεί αποτελεσματική στην πρόβλεψη χρονοσειρών σε διάφορους τομείς.

Το μοντέλο της μεθόδου ARIMA εκφράζεται ως εξής:

ARIMA(p, d, q)

Όπου:

- p: Η τάξη του auto-regression (AR) του μοντέλου, που αντιπροσωπεύει τον αριθμό των παρελθόντων τιμών που χρησιμοποιούνται για την πρόβλεψη της τρέχουσας τιμής.
- d: Η τάξη του integration (I) του μοντέλου, που αντιπροσωπεύει τον αριθμό των διαφορίσεων που απαιτούνται για την επίτευξη σταθερότητας στη χρονοσειρά.
- q: Η τάξη του moving average (MA) του μοντέλου, που αντιπροσωπεύει τον αριθμό των όρων κινητού μέσου όρου που χρησιμοποιούνται για την πρόβλεψη της τρέχουσας τιμής.

Το μαθηματικό μοντέλο ARIMA μπορεί να γραφτεί ως:

$$\Delta^d y_t = \alpha + \varphi_1 \Delta^d y_{t-1} + \dots + \varphi_p \Delta^d y_{t-p} + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

Όπου:

- $\Delta^d y_t$ είναι η d-η διαφορά της χρονοσειράς y_t
- α είναι ο σταθερός όρος
- φ_i είναι οι συντελεστές auto-regression (AR)
- θ_j είναι οι συντελεστές moving average (MA)
- ε_t είναι ο όρος σφάλματος στο χρόνο t

Η εκπαίδευση ενός μοντέλου ARIMA περιλαμβάνει την εκτίμηση των παραμέτρων (α , φ_i , θ_j) με βάση τα δεδομένα της χρονοσειράς. Αυτό μπορεί να γίνει με τη χρήση μεθόδων βελτιστοποίησης, όπως η μέθοδος ελαχίστων τετραγώνων ή η μέθοδος μέγιστης πιθανοφάνειας.

Μόλις έχει εκπαιδευτεί το μοντέλο, μπορεί να χρησιμοποιηθεί για την πρόβλεψη μελλοντικών τιμών της χρονοσειράς. Οι προβλέψεις μπορούν να βελτιωθούν με τη χρήση τεχνικών, όπως cross-validation ή grid search, που επιτρέπουν την αξιολόγηση και την επιλογή του καλύτερου μοντέλου με βάση την επίδοσή του σε ένα σύνολο δεδομένων ελέγχου [15].

Για την πτυχιακή εργασία, επιλέχθηκε να χρησιμοποιηθεί η μέθοδος ARIMA, καθώς προσφέρει μια αποτελεσματική και εύκολα προσαρμόσιμη προσέγγιση για την πρόβλεψη χρονοσειρών. Επιπλέον, το ARIMA είναι κατανοητό και εύκολο να εξηγηθεί, καθιστώντας το ιδανικό για την παρουσίαση και τη συζήτηση των αποτελεσμάτων της πτυχιακής εργασίας.

Συνοψίζοντας, η μέθοδος ARIMA είναι μια δημοφιλής και αποτελεσματική τεχνική για την ανάλυση και την πρόβλεψη χρονοσειρών. Η κατανόηση των βασικών στοιχείων του μοντέλου και η εφαρμογή του σε πραγματικά δεδομένα είναι καίρια για την επίτευξη ακριβών και αξιόπιστων προβλέψεων.

Κεφάλαιο 4ο: Τεχνολογίες

4.1 Τεχνολογίες Front-End

4.1.1 HTML5 και CSS

Οι τεχνολογίες HTML5 και CSS είναι θεμελιώδεις για τη δημιουργία και τον σχεδιασμό ιστοσελίδων. Οι τεχνολογίες αυτές χρησιμοποιήθηκαν στην πτυχιακή εργασία, καθώς επιτρέπουν τη δημιουργία προσαρμοστικών και λειτουργικών ιστοσελίδων.

HTML5 :

Η HTML5 είναι η τελευταία έκδοση της γλώσσας HTML (HyperText Markup Language), που επιτρέπει τη δημιουργία και τη δομή του περιεχομένου μιας ιστοσελίδας. Η HTML είναι μια γλώσσα σήμανσης κειμένου που χρησιμοποιείται για τη δημιουργία και τη δομή του περιεχομένου μιας ιστοσελίδας. Η HTML δεν είναι αυστηρά μια γλώσσα προγραμματισμού, αλλά μια γλώσσα σήμανσης που διαμορφώνει τη δομή της ιστοσελίδας, ορίζοντας τα στοιχεία, όπως επικεφαλίδες, παραγράφους, λίστες και σύνδεσμους.

Η HTML χρησιμοποιεί τα λεγόμενα "tags" ή ετικέτες για να ορίζει τα διάφορα στοιχεία της ιστοσελίδας. Οι ετικέτες είναι εντολές που τοποθετούνται μέσα σε αγκύλες (<>) και εφαρμόζονται στο κείμενο για να ορίσουν τη δομή και την εμφάνιση του περιεχομένου. Για παράδειγμα, η ετικέτα <h1> ορίζει μια επικεφαλίδα πρώτου επιπέδου, ενώ η ετικέτα <p> ορίζει μια παράγραφο. Η HTML αποτελεί το θεμέλιο του Παγκόσμιου Ιστού και είναι απαραίτητη για τη δημιουργία ιστοσελίδων. Ωστόσο, για να προσφέρουν οι ιστοσελίδες πιο προηγμένη λειτουργικότητα και εμφάνιση, συχνά συνδυάζεται με άλλες τεχνολογίες, όπως το CSS (Cascading Style Sheets).

Η HTML5 προσφέρει πολλά νέα στοιχεία και χαρακτηριστικά που επιτρέπουν την ενσωμάτωση πολυμέσων, την αλληλεπίδραση με τον χρήστη και τη βελτίωση της προσβασιμότητας. Μερικά από τα νέα στοιχεία που προσφέρει η HTML5 είναι τα <video>, <audio>, <canvas> και <section>. Επιλέγοντας να χρησιμοποιήσουν την HTML5, οι προγραμματιστές μπορούν να δημιουργήσουν ολοκληρωμένες και σύγχρονες ιστοσελίδες που ανταποκρίνονται στις σύγχρονες ανάγκες των χρηστών [16].

CSS :

Η CSS (Cascading Style Sheets) είναι μια γλώσσα σχεδίασης που χρησιμοποιείται για τον ορισμό της εμφάνισης και της διάταξης των στοιχείων που βρίσκονται σε μια ιστοσελίδα. Η CSS επιτρέπει τον χωρισμό του περιεχομένου από την εμφάνιση, καθιστώντας τη διαδικασία σχεδίασης και συντήρησης των ιστοσελίδων πιο αποτελεσματική και ευέλικτη. Με τη χρήση της CSS, μπορούμε να ορίσουμε τα χρώματα, τις γραμματοσειρές, τα περιθώρια, τις αποστάσεις και άλλα στοιχεία σχεδίασης για την ιστοσελίδα μας.

Η CSS είναι ιδιαίτερα χρήσιμη για τη δημιουργία προσαρμοστικών και αποκρίσιμων ιστοσελίδων, καθώς μας επιτρέπει να ορίζουμε διαφορετικά στυλ για διαφορετικές συσκευές και αναλύσεις οθόνης. Επιλέξαμε να χρησιμοποιήσουμε την CSS στην πτυχιακή μας εργασία, γιατί μας επιτρέπει να δημιουργήσουμε μια καλά οργανωμένη και ευχάριστη εμπειρία χρήστη, προσαρμοζόμενη σε διάφορες συσκευές και περιβάλλοντα περιήγησης [17].

Συνοψίζοντας, οι τεχνολογίες HTML5 και CSS είναι βασικά εργαλεία για τη δημιουργία και τον σχεδιασμό ιστοσελίδων, προσφέροντας μεγάλη ευελιξία και προσαρμοστικότητα. Χρησιμοποιώντας την HTML5, μπορούμε να δημιουργήσουμε πλούσιο περιεχόμενο και να ενσωματώσουμε πολυμέσα στις ιστοσελίδες μας, ενώ η CSS μας επιτρέπει να ορίσουμε την εμφάνιση και τη διάταξη των στοιχείων, εξασφαλίζοντας μια καλή εμπειρία χρήστη ανεξαρτήτως συσκευής ή περιβάλλοντος περιήγησης. Η χρήση αυτών των τεχνολογιών στην πτυχιακή μας εργασία ήταν απαραίτητη για την ανάπτυξη μιας ολοκληρωμένης, σύγχρονης και προσαρμοστικής ιστοσελίδας που θα ανταποκρίνεται στις ανάγκες των χρηστών.

4.1.2 Javascript και βιβλιοθήκες της

JavaScript :

Η JavaScript είναι μια δυναμική, αντικειμενοστρεφής γλώσσα προγραμματισμού υψηλού επιπέδου, η οποία χρησιμοποιείται κυρίως για την ανάπτυξη διαδραστικών ιστοσελίδων και εφαρμογών. Αναπτύχθηκε για πρώτη φορά το 1995 από τον Brendan Eich, ενώ εργαζόταν στην Netscape Communications. Είναι μια από τις πιο δημοφιλείς γλώσσες προγραμματισμού στον κόσμο και υποστηρίζεται από όλους τους μείζονες περιηγητές.

Η JavaScript επιτρέπει στους προγραμματιστές να προσθέτουν διαδραστικά στοιχεία και λειτουργικότητα στις ιστοσελίδες, όπως αλληλεπίδραση με τον χρήστη, αναζήτηση και επεξεργασία δεδομένων, καθώς και ελέγχους φόρμας. Η JavaScript μπορεί να τρέξει σε όλα τα επίπεδα της προγραμματιστικής στοίβας, από τον περιηγητή του πελάτη ως τον server, χρησιμοποιώντας τεχνολογίες όπως η Node.js.

Η JavaScript είναι επίσης πολύ ευέλικτη, καθώς υποστηρίζει πολλά προγραμματιστικά παραδείγματα, όπως αντικειμενοστρεφής, λειτουργικός και συμβατικός προγραμματισμός. Αυτό επιτρέπει στους προγραμματιστές να επιλέγουν τον τρόπο που ταιριάζει καλύτερα στις ανάγκες τους και τους βοηθά να αναπτύξουν κώδικα πιο αποδοτικά και ευκολότερα [18].

Η JavaScript έχει αναπτυχθεί σημαντικά κατά τη διάρκεια των ετών, με την προσθήκη νέων χαρακτηριστικών και βελτιώσεων στην απόδοση. Οι τελευταίες εκδόσεις της γλώσσας περιλαμβάνουν νέα στοιχεία, όπως οι προδιαγραφές ECMAScript (ECMA-262), οι οποίες ορίζουν τη σύνταξη, τα στοιχεία και τις λειτουργίες της γλώσσας. Αυτές οι βελτιώσεις επιτρέπουν την ανάπτυξη πιο πολύπλοκων και υψηλής απόδοσης εφαρμογών [19].

Η JavaScript είναι κρίσιμη για τη δημιουργία δυναμικών, ανταποκριτικών και ευχάριστων εμπειριών χρήστη στον ιστό. Σε συνδυασμό με την HTML5 και τη CSS, η JavaScript παρέχει τη βάση για την ανάπτυξη σύγχρονων ιστοσελίδων και εφαρμογών, καθώς και την προσαρμογή και επέκτασή τους σύμφωνα με τις απαιτήσεις των χρηστών και της αγοράς.

Bootstrap :

Η Bootstrap είναι μια δημοφιλής βιβλιοθήκη πλαισίου (framework) για την ανάπτυξη αποκρίσιμων και κινητών ιστοσελίδων. Αναπτύχθηκε αρχικά από την ομάδα του Twitter και αποτελεί ένα ανοιχτού κώδικα εργαλείο, το οποίο βοηθά τους προγραμματιστές να δημιουργήσουν γρήγορα και εύκολα επαγγελματικής ποιότητας ιστοσελίδες.

Η Bootstrap παρέχει μια συλλογή από επαναχρησιμοποιήσιμα στοιχεία διασύνδεσης, όπως φόρμες, κουμπιά, πλοήγηση, κεφαλίδες, υποσέλιδα και πολλά άλλα. Ταυτόχρονα, περιλαμβάνει ένα πλήρες σύστημα πλαισίων (grid system) που επιτρέπει τη δημιουργία ευέλικτων και αποκρίσιμων διατάξεων. Η Bootstrap χρησιμοποιεί HTML, CSS και JavaScript για να προσφέρει αυτά τα στοιχεία και τις λειτουργίες [20].

Η χρήση της Bootstrap στην πτυχιακή εργασία επιλέχθηκε λόγω της ταχύτητας ανάπτυξης και της επαγγελματικής εμφάνισης που παρέχει. Επιπλέον, η Bootstrap εξασφαλίζει ότι η ιστοσελίδα θα είναι συμβατή με διάφορες συσκευές και προγράμματα περιήγησης, εξασφαλίζοντας έτσι μια καλύτερη εμπειρία για τους χρήστες. Η Bootstrap διαθέτει μια μεγάλη κοινότητα υποστήριξης και πολλούς οδηγούς, που καθιστούν εύκολη την εκμάθηση και την εφαρμογή του πλαισίου σε έργα ανάπτυξης ιστοσελίδων.

ApexCharts.js :

Η ApexCharts.js είναι μια εύχρηστη και ισχυρή βιβλιοθήκη της JavaScript για τη δημιουργία διαδραστικών γραφημάτων και διαγραμμάτων που μπορούν να ενσωματωθούν σε ιστοσελίδες και εφαρμογές. Η ApexCharts.js υποστηρίζει μια πληθώρα τύπων γραφημάτων, όπως γραμμικά, ιστογράμματα, θερμοκρασίας, ραβδωτά, μονάδων, και κυκλικά διαγράμματα.

Η βιβλιοθήκη ApexCharts.js προσφέρει επίσης ευελιξία στην προσαρμογή των γραφημάτων, με πολλές επιλογές διαμόρφωσης και θεμάτων. Επιπλέον, τα γραφήματα είναι αποκρίσιμα και προσαρμόζονται αυτόματα στις αλλαγές του μεγέθους της οθόνης [21].

Στην παρούσα πτυχιακή εργασία, επιλέχθηκε η βιβλιοθήκη ApexCharts.js για την παρουσίαση των γραφημάτων των χρονοσειρών, καθώς προσφέρει πολλά εργαλεία και δυνατότητες που καθιστούν την ανάλυση και την κατανόηση των δεδομένων ευκολότερη και πιο ευχάριστη για τον χρήστη.

4.2 Τεχνολογίες Back-End

Η επιλογή των τεχνολογιών έγινε με βάση την αποτελεσματικότητα, την ευκολία χρήσης και την ευελιξία τους. Η Python χρησιμοποιήθηκε τόσο για την ανάλυση και την πρόβλεψη των χρονοσειρών όσο και για την υλοποίηση του server. Το FastAPI, ένα μοντέρνο και υψηλής απόδοσης web framework, επιλέχθηκε για την κατασκευή του server, καθώς προσφέρει γρήγορη ανάπτυξη και ευκολία συντήρησης. Τέλος, η MySQL χρησιμοποιήθηκε για τη διαχείριση των δεδομένων και την επικοινωνία με τη βάση δεδομένων. Στη συνέχεια, θα αναλύσουμε λεπτομερώς κάθε τεχνολογία και τον τρόπο που συμβάλλουν στην υλοποίηση της πτυχιακής εργασίας.

4.2.1 Python

Η Python είναι μια δυναμική, υψηλού επιπέδου, αντικειμενοστραφής γλώσσα προγραμματισμού, η οποία ξεχωρίζει για την ευκολία χρήσης, την καθαρή σύνταξη και την ευαναγνωσία του κώδικα. Αναπτύχθηκε από τον Guido van Rossum το 1991. Χαρακτηρίζεται από την ευκολία χρήσης και την καθαρή, κατανοητή σύνταξή της, που επιτρέπει στους προγραμματιστές να εκφράζουν τις ιδέες τους με λιγότερο κώδικα σε σύγκριση με άλλες γλώσσες προγραμματισμού. Αυτά τα χαρακτηριστικά καθιστούν την Python εξαιρετική επιλογή για την ανάπτυξη εφαρμογών σε διάφορους τομείς, όπως web development, ανάλυση δεδομένων, μηχανική μάθηση και τεχνητή νοημοσύνη [22].

Η Python διαθέτει μια εκτεταμένη βιβλιοθήκη τυποποιημένων modules, τα οποία καλύπτουν πολλές λειτουργίες, καθιστώντας τη γλώσσα ακόμα πιο ισχυρή και ευέλικτη. Επιπλέον, η Python υποστηρίζει πολλά εξωτερικά πακέτα, τα οποία προσφέρουν πρόσθετες δυνατότητες και επεκτείνουν τις εφαρμογές της γλώσσας. Κάποια από τα δημοφιλή πακέτα που χρησιμοποιούνται στην Python είναι τα NumPy, pandas, scikit-learn και TensorFlow [23].

4.2.2 FastAPI

Το FastAPI είναι ένα σύγχρονο, υψηλής απόδοσης web framework για τη δημιουργία APIs με τη γλώσσα προγραμματισμού Python, βασισμένο στο standard Python type hints. Η κύρια ιδέα πίσω από το FastAPI είναι η δυνατότητα να παρέχει υψηλής ταχύτητας, εύκολη στη χρήση και ευέλικτη δημιουργία API, χρησιμοποιώντας κώδικα που είναι εύκολο να γραφτεί και να διαβαστεί.

Το FastAPI βασίζεται στο Starlette για το web framework και στο Pydantic για τη δήλωση των αντικειμένων που χρησιμοποιούνται από τα endpoints του API. Αυτό επιτρέπει στο FastAPI να παρέχει υψηλή απόδοση χάρη στην ασύγχρονη υποστήριξη και την αυτόματη βελτιστοποίηση του κώδικα.

Κάποια από τα πλεονεκτήματα του FastAPI είναι η αυτόματη δημιουργία τεκμηρίωσης για το API, η ευκολία επέκτασης και η υποστήριξη για την ανάπτυξη των API-driven προγραμμάτων. Επιπλέον, το FastAPI διευκολύνει τη διαχείριση εξαρτήσεων και την ανάπτυξη modular κώδικα [24].

4.2.3 MySQL

Η MySQL είναι ένα από τα πιο δημοφιλή συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) που χρησιμοποιείται ευρέως στην ανάπτυξη ιστοσελίδων και εφαρμογών. Η MySQL δημιουργήθηκε από την εταιρεία MySQL AB, που αποκτήθηκε αργότερα από την Oracle Corporation. Χρησιμοποιεί τη γλώσσα SQL (Structured Query Language) για τη διαχείριση και την αναζήτηση δεδομένων.

Η MySQL προσφέρει πολλά πλεονεκτήματα, όπως η ευελιξία, η αξιοπιστία και η απόδοση. Με την MySQL, οι προγραμματιστές μπορούν να δημιουργήσουν και να διαχειριστούν περίπλοκες βάσεις δεδομένων, που αποθηκεύουν και ανακτούν δεδομένα από διαφορετικές πηγές. Η MySQL υποστηρίζει τις βασικές λειτουργίες των σχεσιακών βάσεων δεδομένων, όπως οι πίνακες, τα ερωτήματα, τα ευρετήρια, τα πεδία και οι σχέσεις μεταξύ των πινάκων. Επίσης, υποστηρίζει διάφορους τύπους δεδομένων, όπως κείμενο, αριθμούς, ημερομηνίες, και δυαδικά δεδομένα [25].

Στην πτυχιακή εργασία, επιλέξαμε την MySQL ως το σύστημα διαχείρισης βάσεων δεδομένων (DBMS) για την αποθήκευση και ανάκτηση των δεδομένων. Η MySQL επιτρέπει την αποτελεσματική και ασφαλή αποθήκευση των δεδομένων, καθιστώντας την εύκολη πρόσβαση σε αυτά μέσω του backend. Η χρήση της MySQL επίσης εξασφαλίζει την ανεξαρτησία των δεδομένων από την εφαρμογή, καθώς τα δεδομένα είναι αποθηκευμένα σε ένα κεντρικό σημείο και μπορούν να χρησιμοποιηθούν από διαφορετικές εφαρμογές και συστήματα.

Κεφάλαιο 5ο: Πρόβλεψη τιμών υγρών καυσίμων με το μοντέλο ARIMA

Στα πλαίσια αυτού του κεφαλαίου θα εστιάσουμε το ενδιαφέρον μας στα πιο πρακτικά σημεία της πτυχιακής εργασίας όπως η υλοποίηση της προ-επεξεργασίας των πρωτογενή δεδομένων με σκοπό την κατάλληλη προετοιμασία τους για την περαιτέρω ανάλυση. Έπειτα θα αναφερθεί με παραδείγματα κώδικα η υλοποίηση της μεθόδου ARIMA και το πως θα χρησιμοποιηθεί για την πρόβλεψη των μελλοντικών τιμών των υγρών καυσίμων στην Ελλάδα. Τέλος, θα παρουσιαστούν πειράματα που γίναν ώστε να επιλεγεί το καταλληλότερο και βέλτιστο μοντέλο παραμέτρων για το σύνολο των δεδομένων μας καθώς και οπτική απεικόνιση των αποτελεσμάτων - προβλέψεων σε μορφή γραφημάτων.

5.1 Προ-επεξεργασία των δεδομένων στην Python

Όπως έχει ήδη αναφερθεί στο κεφάλαιο 2, στα πλαίσια άλλης διπλωματικής εργασίας αναπτύχθηκε ένας μηχανισμός για την αυτόματη λήψη και αποθήκευση των τιμών των υγρών καυσίμων σε μια σχεσιακή βάση δεδομένων και πιο συγκεκριμένα, στον πίνακα `prices`.

Για την υλοποίηση της προ-επεξεργασίας των δεδομένων καθώς και για την οπτικοποίηση τους, χρησιμοποιήθηκαν κάποιες πολύ βασικές βιβλιοθήκες της python όπως η `pandas`, η `numpy` και η `matplotlib`.

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

Για να έχουμε μια πρώτη εικόνα και άποψη για το σύνολο των δεδομένων αυτών, αφού πήραμε πρόσβαση στην σχεσιακή βάση δεδομένων, καταχωρήσαμε στην Python τον πίνακα `prices` σε μορφή `dataframe` ως εξής :

```
connection = get_connection()
sql = f"""SELECT unleaded_95, unleaded_100, motor_oil, autogas,
              record_date, county_code FROM `{table_name}`"""
df = pd.read_sql(sql, connection)
connection.close()
```

Όπου :

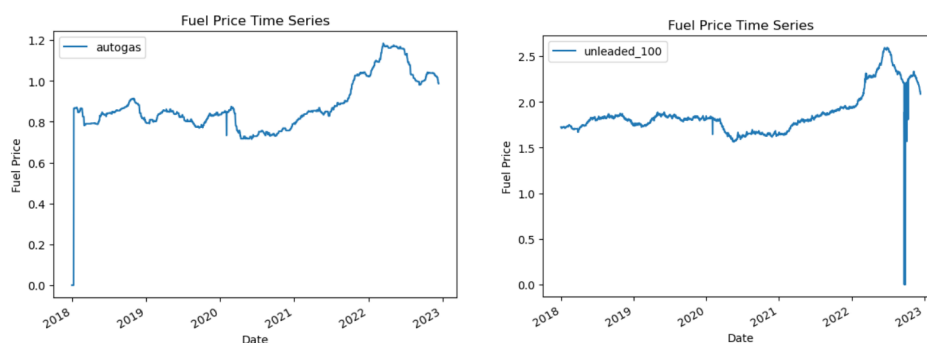
- `table_name`: το όνομα του πίνακα `prices`

Το dataframe που έχουμε μοιάζει κάπως έτσι :

county_code	unleaded_95	unleaded_100	motor_oil	autogas	record_date
1	1.706	1.883	1.447	0.896	09.10.2021
2	1.762	1.933	1.495	0.997	09.10.2021
3	1.727	1.895	1.462	0.957	09.10.2021
4	1.748	1.914	1.478	0.990	09.10.2021
5	1.727	1.898	1.456	0.960	09.10.2021
...
47	1.873	2.162	1.833	0.999	13.12.2022
48	1.942	2.204	1.897	0.998	13.12.2022
49	1.911	2.146	1.847	0.996	13.12.2022
50	1.894	2.108	1.846	1.124	13.12.2022
51	1.865	2.130	1.792	1.258	13.12.2022

Σχήμα 5.1: Dataframe του πίνακα prices

Ως επόμενο βήμα κάναμε οπτικοποίηση των δεδομένων σε γράφημα ξεχωριστά για κάθε νομό και για κάθε τύπο καυσίμου. Παρακάτω παρουσιάζονται δύο τυχαία γραφήματα για τον νομό Άρτης, για τα καύσιμα τύπου "αερίου" και "αμόλυβδης 100" :



Σχήμα 5.2: Γραφήματα των καυσίμων "αέριο" και "αμόλυβδη 100" για τον νομό Άρτης του πίνακα prices

Από τα γραφήματα που προκύπτουν είναι εύκολα αντιληπτό ότι υπάρχουν μηδενικές τιμές καθώς και κάποια ελλιπή δεδομένα σε κάποιες ημερομηνίες, πράγμα το οποίο είναι αδύνατο να υφίσταται στην πραγματική ζωή καθώς η τιμές των υγρών καυσίμων δεν είναι ποτέ μηδενικές.

Για την αντιμετώπιση του προβλήματος αυτού, αποφασίστηκε να γίνει η γραμμική παρεμβολή σε όλο το dataframe. Η υλοποίηση της στην python μέσω της ολοκληρωμένης μεθόδου `interpolate_missing_values()`:

```
def interpolate_missing_values(table_name, nomos_code):
    connection = get_connection()
    sql = f"""SELECT unleaded_95, unleaded_100, motor_oil, autogas,
                    record_date, county_code
                FROM `{table_name}` where county_code='{nomos_code}'"""
    df = pd.read_sql(sql, connection)
    connection.close()

    df = df.sort_values(by=['record_date']).set_index('record_date')
    idx = pd.date_range(df.index.min(), df.index.max())
    df = df.reindex(idx)
    df.replace(0, np.nan, inplace=True)
    df = df.interpolate(method='linear', limit_direction='both')

    df.reset_index(inplace=True)
    df = df.rename(columns={"index": "record_date"})

    return df
```

Αμέσως μετά την γραμμική παρεμβολή υπολογίστηκε και η μέση τιμή ανά κάθε τύπο καυσίμου για όλη την Ελλάδα. Αυτό το βήμα είναι απαραίτητο ώστε να μπορέσουμε στην συνέχεια να εφαρμόσουμε το μοντέλο ARIMA για να προβλέψουμε τις τιμές των υγρών καυσίμων σε ολόκληρη την επικράτεια. Ως "county_code" για την Ελλάδα ανατέθηκε το 0 (μηδέν) . :

```
def calculate_avg_of_country(df):
    df_avg = df.groupby('record_date').agg({
        'unleaded_95': 'mean', 'unleaded_100': 'mean',
        'motor_oil': 'mean',
        'autogas': 'mean'
    }).reset_index()

    df_avg[['unleaded_95',
             'unleaded_100',
             'motor_oil',
             'autogas']] = df_avg[['unleaded_95', 'unleaded_100',
                                   'motor_oil',
                                   'autogas']].round(3)

    df_avg['county_code'] = 0

    return df_avg
```

Ύστερα από την επιτυχή δημιουργία όλων των παραπάνω μεθόδων, δημιουργούμε δυναμικά μέσω της Python τον πίνακα `interpolated_prices` με τον παρακάτω μέθοδο :

```
def create_interpolated_table(table_name):
    connection = get_connection()
    regions = retrieve_data(connection, "SELECT code, name FROM county")
    connection.close()

    county_codes = [d['code'] for d in regions]

    df = pd.DataFrame(columns=['record_date',
                              'unleaded_95',
                              'unleaded_100',
                              'motor_oil',
                              'autogas',
                              'county_code'])

    for region_code in county_codes:
        df = pd.concat(
            [df, interpolate_missing_values(table_name, region_code)])

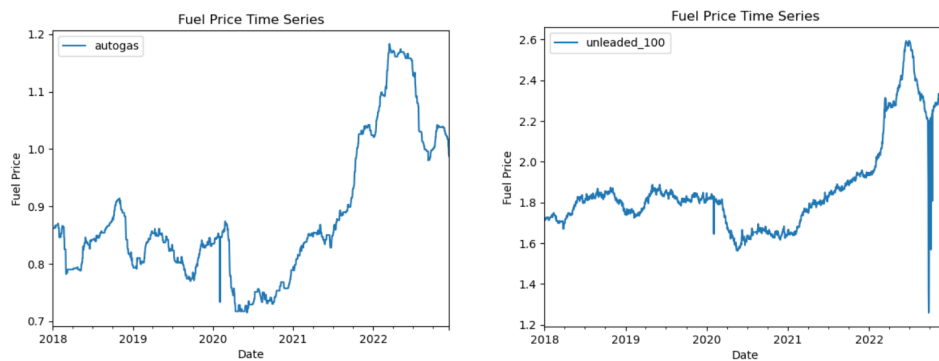
    df_avg = calculate_avg_of_country(df)

    df = pd.concat([df_avg, df])

    engine = create_engine(
        f"mysql+mysqlconnector://{conn.USER}:{conn.PASSWORD}@{conn.HOST}/{conn.DATABASE}")

    connection = get_connection()
    df.to_sql(name="interpolated_prices", con=engine,
              if_exists='replace', index=False)
    connection.commit()
    connection.close()
```

Ακολουθώντας την διαδικασία που περιγράφηκε παραπάνω και υλοποιώντας όλες τις μεθόδους που ανφέρθηκαν για την προ-επεξεργασία των δεδομένων, τα ζητήματα που προέκυψαν από τις ελλείψεις των δεδομένων αντιμετωπίστηκαν επαρκώς, όπως φαίνεται και απο τα παρακάτω γραφήματα :



Σχήμα 5.3: Γραφήματα των καυσίμων "αέριο" και "αμόλυβδη 100" για τον νομό Άρτης του πίνακα `interpolated_prices`

Τέλος, έχουμε πλέον έναν ικανοποιητικό και ολοκληρωμένο πίνακα δεδομένων για τις τιμές των υγρών καυσίμων ανά τύπο καυσίμου και νομό της Ελλάδας καθώς και την μέση τιμή κάθε καυσίμου για όλη τη χώρα. Ο πίνακας αυτός είναι όπως αναφέραμε ο `interpolated_prices` και μοιάζει κάπως έτσι σε μορφή `dataframe` :

<code>record_date</code>	<code>unleaded_95</code>	<code>unleaded_100</code>	<code>motor_oil</code>	<code>autogas</code>	<code>county_code</code>
01.01.2018 0:00	1.561	1.752	1.264	0.860	0
02.01.2018 0:00	1.562	1.753	1.265	0.861	0
03.01.2018 0:00	1.563	1.755	1.266	0.861	0
04.01.2018 0:00	1.564	1.754	1.268	0.862	0
05.01.2018 0:00	1.565	1.750	1.269	0.863	0
...
09.12.2022 0:00	1.896	2.172	1.835	1.258	51
10.12.2022 0:00	1.885	2.172	1.820	1.258	51
11.12.2022 0:00	1.885	2.172	1.819	1.258	51
12.12.2022 0:00	1.877	2.167	1.807	1.258	51
13.12.2022 0:00	1.865	2.130	1.792	1.258	51

Σχήμα 5.4: `Dataframe` του πίνακα `interpolated_prices`

5.2 Χρήση του μοντέλου ARIMA στην Python

Για την υλοποίηση του μοντέλου ARIMA στην Python ως αρχικό βήμα πρέπει να γίνει εισαγωγή του ARIMA module απο την βιβλιοθήκη `statsmodels` ως εξής :

```
from statsmodels.tsa.arima.model import ARIMA
```

Έπειτα, δημιουργήθηκαν δύο βοηθητικές μέθοδοι :

```
def build_arima_model(data, order):
    return ARIMA(data, order=order).fit()
```

```
def get_predictions(results, start, end, dynamic=False):
    return results.get_prediction(start=start, end=end, dynamic=dynamic)
```

Όπου η `build_arima_model` δέχεται δύο παραμέτρους :

- *data*: ένα σύνολο δεδομένων τύπου dataframe με τις ημερομηνίες και τις τιμές ενός τύπου καυσίμου
- *order*: τις τιμές για τις παραμέτρους p , d και q της ARIMA

και επιστρέφει το εκπαιδευμένο μοντέλο της ARIMA πάνω στα δεδομένα που δέχθηκε ως input. Από την άλλη, η `get_predictions` δέχεται τέσσερις παραμέτρους :

- *results*: το εκπαιδευμένο μοντέλο της ARIMA
- *start*: η ημερομηνία έναρξης των προβλέψεων
- *end*: η ημερομηνία λήξης των προβλέψεων
- *dynamic = False*: οι προβλέψεις δημιουργούνται χρησιμοποιώντας μόνο ιστορικά δεδομένα και τις εκτιμώμενες παραμέτρους του μοντέλου

και επιστρέφει τις προβλέψεις των τιμών των υγρών καυσίμων σε μορφή λίστας.

Για την συνέχεια της πτυχιακής αποφασίστηκε ότι θα γίνουν δύο προβλέψεις για κάθε τύπο καυσίμου στην χώρα. Η μία θα αφορά προβλέψεις μελλοντικές, δηλαδή προβλέψεις για ένα μήνα ή για τρεις μήνες μπροστά από την τελευταία ημερομηνία που έχουμε δεδομένα. Η δεύτερη θα αφορά παρελθοντικές ημερομηνίες. Αυτό σημαίνει ότι θα χωρίζουμε το dataframe μας σε `train_df` και `test_df`. Το `train_df` θα περιλαμβάνει το κομμάτι των δεδομένων με το οποίο θα εκπαιδευτεί το μοντέλο της ARIMA, ενώ το `test_df` θα περιλαμβάνει το κομμάτι εκείνο με το οποίο θα γίνουν οι αξιολογήσεις των αποτελεσμάτων των προβλέψεων του μοντέλου της ARIMA. Ο διαχωρισμός το αρχικού dataframe θα γίνεται είτε ένα μήνα είτε τρεις μήνες πίσω από την τελευταία καταγεγραμμένη ημερομηνία του συνόλου των δεδομένων.

Για την αξιολόγηση των αποτελεσμάτων επιλέχθηκαν δύο δείκτες, το Root Mean Squared Error (RMSE) και το Mean Absolute Percentage Error (MAPE).

- *Root Mean Squared Error (RMSE)*: Το RMSE είναι μια μετρική που χρησιμοποιείται για την εκτίμηση της διαφοράς μεταξύ τιμών που προβλέπονται από ένα μοντέλο ή εκτιμητή και τις πραγματικές τιμές. Είναι μια δημοφιλής μετρική όπως φαίνεται από την ευρεία της χρήση σε διάφορα πεδία, συμπεριλαμβανομένης της πρόβλεψης, της ανάλυσης παρατηρήσεων και της μηχανικής μάθησης. Ο RMSE υπολογίζεται ως η τετραγωνική ρίζα του μέσου τετραγωνικού σφάλματος, που είναι ο μέσος όρος των τετραγώνων των διαφορών μεταξύ των προβλεπόμενων και των πραγματικών τιμών. Αυτός ο υπολογισμός επιβραβεύει τα μεγαλύτερα σφάλματα, κάνοντας τον RMSE ιδιαίτερα χρήσιμο όταν τα μεγάλα σφάλματα είναι πιο ανεπιθύμητα [26]. Το μαθηματικό του μοντέλο είναι:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2}$$

- *Mean Absolute Percentage Error (MAPE)*: Το MAPE είναι επίσης μία πολύ χρήσιμη μετρική για την αξιολόγηση των προβλέψεων ενός μοντέλου. Αντί να υπολογίζει την απόλυτη διαφορά μεταξύ των πραγματικών και των προβλεπόμενων τιμών, όπως στην περίπτωση του RMSE, το MAPE υπολογίζει την απόλυτη διαφορά ως ποσοστό της πραγματικής τιμής. Αυτό το καθιστά ιδιαίτερα χρήσιμο σε περιπτώσεις όπου επιθυμούμε να εκφράσουμε τα λάθη πρόβλεψης σε ποσοστιαίους όρους. Ένα μεγάλο πλεονέκτημα του MAPE είναι ότι είναι εύκολο να ερμηνευθεί, επειδή παρέχει έναν άμεσο δείκτη της απόδοσης του μοντέλου σε ποσοστιαίους όρους [26]. Το μαθηματικό του μοντέλο είναι:

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - y'_i}{y_i} \right|$$

Για την υλοποίηση τους στην Python χρησιμοποιήθηκαν οι παρακάτω βιβλιοθήκες, καθώς δημιουργήθηκε και μία βοηθητική μέθοδος για το MAPE :

```
from sklearn.metrics import mean_squared_error, mean_absolute_error
import numpy as np

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

Η παρακάτω μέθοδος είναι η υλοποίηση των παρελθοντικών προβλέψεων καθώς και οι αξιολογήσεις τους:

```

def existing_predictor(fuel, forecastPeriod, df):

    train_df = df[:len(df)-forecastPeriod]
    test_df = df[len(df)-forecastPeriod:]

    train_df.index.freq = pd.infer_freq(train_df.index)

    testing_results = build_arima_model(train_df[fuel], (p, d, q))
    testing_predictions = get_predictions(testing_results, pd.to_datetime(
        test_df.index[0]), pd.to_datetime(test_df.index[-1]))

    mape = round(mean_absolute_percentage_error(
        test_df, testing_predictions.predicted_mean), 2)
    rmse = round(np.sqrt(mean_squared_error(
        test_df, testing_predictions.predicted_mean)), 2)

    return {
        'existing_predictions': testing_predictions.predicted_mean,
        'mape': mape,
        'rmse': rmse
    }

```

Οι παρελθοντικές προβλέψεις υπολογίζονται, τόσο για να κάνουμε αξιολόγηση των αποτελεσμάτων με τους δείκτες που αναφέρθηκαν παραπάνω, όσο και για να γίνει οπτικοποίηση των τιμών αυτών στο γράφημα της διαδουκτιακής εφαρμογής που θα υλοποιηθεί στην συνέχεια, ώστε ο χρήστης να έχει και μια οπτική απεικόνιση των αξιολογήσεων των προβλέψεων.

Η παρακάτω μέθοδος είναι η υλοποίηση των μελλοντικών προβλέψεων :

```

def future_predictor(fuel, forecastPeriod, df):

    df.index.freq = pd.infer_freq(df.index)

    future_results = build_arima_model(df[fuel], (p, d, q))

    prediction_dates = [df.index[-1] +
        DateOffset(days=x) for x in range(0, forecastPeriod)]
    prediction_dates = pd.DataFrame(
        index=prediction_dates[1:], columns=df.columns)

    p_start = len(df[fuel])
    p_end = len(df[fuel]) + forecastPeriod - 1
    future_predictions = get_predictions(future_results, p_start, p_end)

    return future_predictions.predicted_mean

```

Στην επόμενη ενότητα θα αναλυθεί πλήρως η μελέτη στο σύνολο των δεδομένων και πως καταλήξαμε στον καθορισμό των παραμέτρων (p, d, q) για την εκπαίδευση του μοντέλου ARIMA.

5.3 Καθορισμός των παραμέτρων

Όπως έχει γίνει ξεκάθαρο και από το θεωρητικό κεφάλαιο 3, η μέθοδος του μοντέλου ARIMA πρόκειται για ένα στατιστικό εργαλείο. Είναι ιδιαίτερα σημαντικό να σημειωθεί ότι η χρονοσειρά που χρησιμοποιείται για την υλοποίηση αυτής της μεθόδου πρέπει να περιλαμβάνει μόνο ένα χαρακτηριστικό. Διατυπώνοντας το πιο απλά, η πρόγνωση ενός χαρακτηριστικού εξαρτάται απολύτως από τις προηγούμενες τιμές του. Είναι προφανές ότι το dataframe σε αυτό το σενάριο περιλαμβάνει μόνο τον παράγοντα χρόνο και τις τιμές ενός είδους υγρού καυσίμου.

Η διαδικασία επιλογής των παραμέτρων για το μοντέλο ARIMA πραγματοποιήθηκε αυτοματοποιημένα μέσω της βιβλιοθήκης `pmdata.ima`, χρησιμοποιώντας τη συνάρτηση `auto_arima`. Η `auto_arima` είναι μια σημαντική λειτουργία που παρέχεται από τη βιβλιοθήκη `pmdata.ima` στην Python. Η `auto_arima` αυτοματοποιεί τη διαδικασία επιλογής του καλύτερου μοντέλου ARIMA για μια σειρά χρόνου, κάτι που μπορεί να είναι ιδιαίτερα δύσκολο και χρονοβόρο αν γίνει χειροκίνητα.

Η `auto_arima` λειτουργεί εκτελώντας μία εξαντλητική (ή ενδεχομένως τυχαία) αναζήτηση σε ένα πλέγμα παραμέτρων και επιλέγοντας το μοντέλο που μεγιστοποιεί μια δεδομένη μετρική ποιότητας (π.χ., το AIC). Τα παραμετρικά όρια είναι τα εξής:

- p : Ο αριθμός των καθυστερημένων εγγραφών που συμπεριλήφθηκαν στο μοντέλο
- d : Ο αριθμός των περιπτώσεων που διαφοροποιούνται οι ακατέργαστες παρατηρήσεις
- q : Το μέγεθος παραθύρου του κινούμενου μέσου

Η `auto_arima` επιστρέφει το καλύτερο μοντέλο ARIMA με βάση το επιλεγμένο κριτήριο πληροφορίας. Στην περίπτωση μας, το επιλεγμένο κριτήριο είναι το AIC, ή Κριτήριο Πληροφορίας Akaike, το οποίο είναι ένα μέτρο που χρησιμοποιείται στη στατιστική για τη σύγκριση και την επιλογή μοντέλων. Το AIC εκτιμά την ποιότητα ενός μοντέλου σχετικά με άλλα μοντέλα. Το μοντέλο με το χαμηλότερο AIC προτιμάται, καθώς αυτό υποδεικνύει μια καλή εκτίμηση των δεδομένων με την ελάχιστη πολυπλοκότητα. Η υλοποίηση του `auto_arima` στην Python είναι η εξής :

```
model = pm.auto_arima(df, start_p=1, start_q=1,
                      test='adf',          # use adftest to find optimal 'd'
                      max_p=15, max_q=15, # maximum p and q
                      m=1,                 # frequency of series
                      d=None,              # let model determine 'd'
                      seasonal=False,     # No Seasonality
                      start_P=0,
                      D=0,
                      trace=True,
                      error_action='ignore',
                      suppress_warnings=True,
                      stepwise=True)
```

Ένα παράδειγμα εφαρμογής της μεθόδου `auto_arima` στο σύνολο των δεδομένων που έχουμε είναι το εξής : Το μοντέλο μας θέλουμε να προβλέψει τις τιμές του αερίου για τον επόμενο μήνα. Τα βήματα που θα ακολουθηθούν είναι τα παρακάτω :

- Αποθηκεύουμε σε μορφή *dataframe* την μέση τιμή του αερίου στην Ελλάδα καθημερινά

```
fuel = 'autogas'
county_code = 0

df = df.loc[df['county_code'] == county_code]
df = df[["record_date", fuel]]
df["record_date"] = pd.to_datetime(df["record_date"], dayfirst=True)
df.set_index('record_date', inplace=True)
```

Το dataframe :

autogas	
record_date	
2018-01-01	0.860
2018-01-02	0.861
2018-01-03	0.861
2018-01-04	0.862
2018-01-05	0.863
...	...
2022-12-09	1.029
2022-12-10	1.025
2022-12-11	1.023
2022-12-12	1.020
2022-12-13	1.017

Σχήμα 5.5: Dataframe της καθημερινής μέσης τιμής του αερίου στην Ελλάδα

- Τρέχουμε την `auto_arima` για το dataframe του αερίου ώστε να μας δώσει τις βέλτιστες παραμέτρους για το τελικό μοντέλο που θα επιλεγεί για την εφαρμογή της ARIMA στο σύνολο των δεδομένων
- Ελέγχουμε τα αποτελέσματα και καταλήγουμε στο μοντέλο

```

Performing stepwise search to minimize aic
ARIMA(1,1,1)(0,0,0)[0] intercept : AIC=-15946.952, Time=0.38 sec
ARIMA(0,1,0)(0,0,0)[0] intercept : AIC=-15825.642, Time=0.37 sec
ARIMA(1,1,0)(0,0,0)[0] intercept : AIC=-15945.386, Time=0.92 sec
ARIMA(0,1,1)(0,0,0)[0] intercept : AIC=-15912.315, Time=0.84 sec
ARIMA(0,1,0)(0,0,0)[0] intercept : AIC=-15826.157, Time=0.14 sec
ARIMA(2,1,1)(0,0,0)[0] intercept : AIC=-15964.899, Time=1.99 sec
ARIMA(2,1,0)(0,0,0)[0] intercept : AIC=-15956.938, Time=0.51 sec
ARIMA(3,1,1)(0,0,0)[0] intercept : AIC=-16047.566, Time=1.19 sec
ARIMA(3,1,0)(0,0,0)[0] intercept : AIC=-16031.464, Time=0.47 sec
ARIMA(4,1,1)(0,0,0)[0] intercept : AIC=-16125.140, Time=1.18 sec
ARIMA(4,1,0)(0,0,0)[0] intercept : AIC=-16105.862, Time=1.96 sec
ARIMA(5,1,1)(0,0,0)[0] intercept : AIC=-16196.041, Time=3.04 sec
ARIMA(5,1,0)(0,0,0)[0] intercept : AIC=-16169.675, Time=2.30 sec
ARIMA(6,1,1)(0,0,0)[0] intercept : AIC=-16218.338, Time=1.39 sec
ARIMA(6,1,0)(0,0,0)[0] intercept : AIC=-16213.008, Time=1.65 sec
ARIMA(7,1,1)(0,0,0)[0] intercept : AIC=-16237.998, Time=1.15 sec
ARIMA(7,1,0)(0,0,0)[0] intercept : AIC=-16244.957, Time=1.81 sec
ARIMA(8,1,0)(0,0,0)[0] intercept : AIC=-16257.925, Time=3.33 sec
ARIMA(9,1,0)(0,0,0)[0] intercept : AIC=-16259.356, Time=1.89 sec
ARIMA(10,1,0)(0,0,0)[0] intercept : AIC=-16257.886, Time=2.05 sec
ARIMA(9,1,1)(0,0,0)[0] intercept : AIC=-16237.329, Time=1.76 sec
ARIMA(8,1,1)(0,0,0)[0] intercept : AIC=-16248.005, Time=3.56 sec
ARIMA(10,1,1)(0,0,0)[0] intercept : AIC=-16234.848, Time=1.90 sec
ARIMA(9,1,0)(0,0,0)[0] intercept : AIC=-16261.207, Time=0.96 sec
ARIMA(8,1,0)(0,0,0)[0] intercept : AIC=-16259.752, Time=0.63 sec
ARIMA(10,1,0)(0,0,0)[0] intercept : AIC=-16259.745, Time=1.05 sec
ARIMA(9,1,1)(0,0,0)[0] intercept : AIC=-16246.998, Time=1.11 sec
ARIMA(8,1,1)(0,0,0)[0] intercept : AIC=-16241.987, Time=0.90 sec
ARIMA(10,1,1)(0,0,0)[0] intercept : AIC=-16244.554, Time=2.84 sec

Best model: ARIMA(9,1,0)(0,0,0)[0]
Total fit time: 43.298 seconds

```

Σχήμα 5.6: Αποτελέσματα βέλτιστων παραμέτρων για το μοντέλο ARIMA

- Επιλογή του μοντέλου $(9,1,0)$ όπου $p = 9$, $d = 1$ και $q = 0$
- Διαχωρισμός του συνολικού `dataframe` σε δεδομένα προς εκπαίδευση (`train_df`) και σε δεδομένα για `test` (`test_df`) με σκοπό την πρόβλεψη τιμών σε ήδη υπάρχουσες ημερομηνίες για την αξιολόγηση των αποτελεσμάτων. Ο διαχωρισμός θα γίνει βάση της μελλοντικής πρόβλεψης που θέλουμε να κάνουμε, δηλαδή ένα μήνα

```
forecastPeriod = 30
```

```
train_df = df[:len(df)-forecastPeriod]
test_df = df[len(df)-forecastPeriod:]
```

```
train_df.index.freq = pd.infer_freq(train_df.index)
```

- Εισαγωγή της μεθόδου *ARIMA* από την βιβλιοθήκη *statsmodels* και εκπαίδευση του μοντέλου

```
from statsmodels.tsa.arima.model import ARIMA
```

```
model = ARIMA(train_df, order=(9,1,0))
results = model.fit()
```

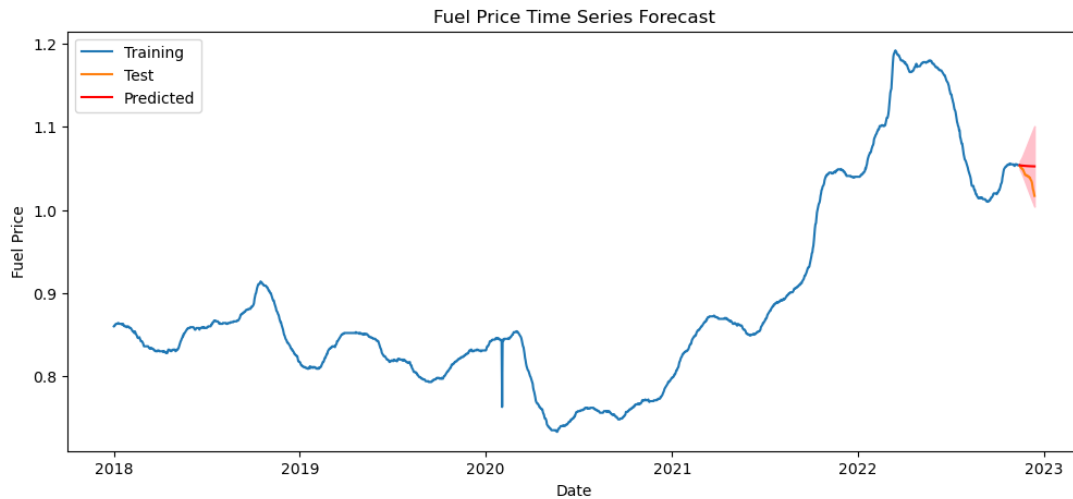
- Καταγραφή των αποτελεσμάτων - προβλέψεων

```
predictions = results.get_prediction(start=pd.to_datetime(test_df.index[0]),
                                     end=pd.to_datetime(test_df.index[-1]),
                                     dynamic=False)
predictions_conf_int = predictions.conf_int()
```

- Οπτικοποίηση των αποτελεσμάτων - προβλέψεων

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

```
plt.figure(figsize=(12, 5))
plt.plot(train_df, label="Training")
plt.plot(test_df, label="Test")
plt.plot(predictions.predicted_mean, label="Predicted", color="r")
plt.fill_between(predictions_conf_int.index,
                 predictions_conf_int.iloc[:, 0],
                 predictions_conf_int.iloc[:, 1], color="pink")
plt.xlabel("Date")
plt.ylabel("Fuel Price")
plt.title("Fuel Price Time Series Forecast")
plt.legend()
plt.show()
```



Σχήμα 5.7: Οπτικοποίηση των αποτελεσμάτων - προβλέψεων

- Αξιολόγηση των αποτελεσμάτων - προβλέψεων με τους δείκτες *RMSE* και *MAPE*

```

from sklearn.metrics import mean_squared_error

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

mape = mean_absolute_percentage_error(test_df, predictions.predicted_mean)
rmse = np.sqrt(mean_squared_error(test_df, predictions.predicted_mean))

print("Root Mean Squared Error: {:.5f}€".format(rmse))
print("Mean Absolute Percentage Error (MAPE): {:.5f}%".format(mape))

```

Root Mean Squared Error: 0.01586€
 Mean Absolute Percentage Error (MAPE): 1.26064%

Σχήμα 5.8: Αξιολόγηση των αποτελεσμάτων - προβλέψεων

- Δημιουργία μελλοντικών ημερομηνιών έως ένα μήνα

```
from pandas.tseries.offsets import DateOffset

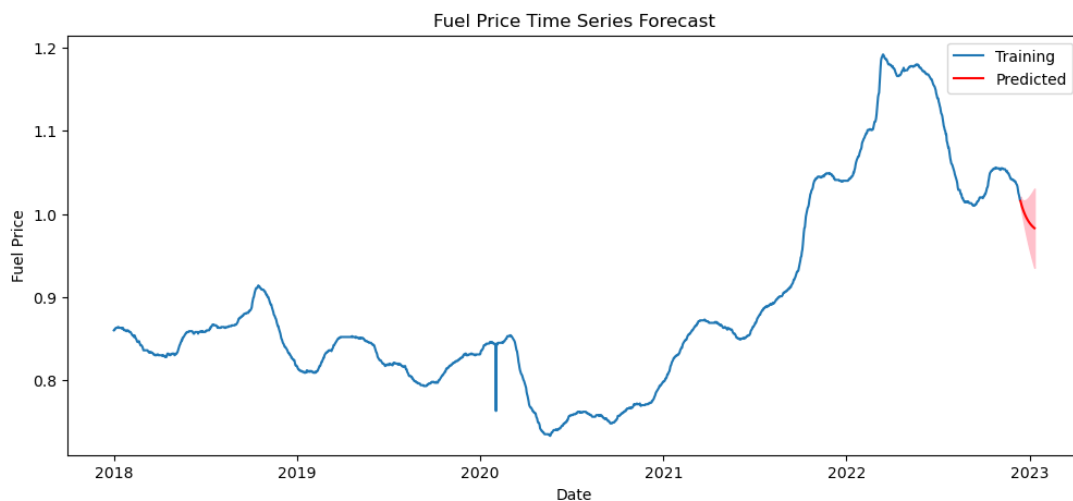
prediction_dates = [df.index[-1] +
                    DateOffset(days=x) for x in range(0, forecastPeriod)]
prediction_dates = pd.DataFrame(
    index=prediction_dates[1:], columns=df.columns)
```

- Εκπαίδευση του μοντέλου, καταγραφή και οπτικοποίηση των αποτελεσμάτων

```
fmodel = ARIMA(df, order=(9,1,0))
results = fmodel.fit()

fpredictions = results.get_prediction(start=pd.to_datetime(prediction_dates.index[0]),
                                     end=pd.to_datetime(prediction_dates.index[-1]),
                                     dynamic=False)
fpredictions_conf_int = fpredictions.conf_int()

plt.figure(figsize=(12, 5))
plt.plot(df, label="Training")
plt.plot(fpredictions.predicted_mean, label="Predicted", color="r")
plt.fill_between(fpredictions_conf_int.index,
                 fpredictions_conf_int.iloc[:, 0],
                 fpredictions_conf_int.iloc[:, 1], color="pink")
plt.xlabel("Date")
plt.ylabel("Fuel Price")
plt.title("Fuel Price Time Series Forecast")
plt.legend()
plt.show()
```



Σχήμα 5.9: Οπτικοποίηση των αποτελεσμάτων - προβλέψεων

5.4 Εκτίμηση των προβλέψεων

Μετά την υλοποίηση του μοντέλου ARIMA στη Python και την παρουσίαση των προβλέψεων μέσω γραφημάτων, καθώς και την ανάλυση των αποτελεσμάτων MAPE και RMSE για 30 μέρες πρόβλεψης για το καύσιμο του αερίου, η επεξήγηση και ανασκόπηση των αποτελεσμάτων είναι απαραίτητη.

Πρώτον, οι τιμές RMSE και MAPE, που είναι αντίστοιχα 0.01586€ και 1.26064%, δείχνουν ότι το μοντέλο πέτυχε αρκετά καλή πρόβλεψη. Το RMSE, ένας δείκτης που μετράει την τυπική απόκλιση των επιφυλάξεων, ήταν σχετικά χαμηλό, δείχνοντας ότι το μοντέλο ήταν ακριβές στις προβλέψεις του. Το MAPE, από την άλλη πλευρά, που εκφράζει το μέσο ποσοστό σφάλματος, είναι επίσης χαμηλό, γεγονός που δηλώνει ότι το μοντέλο κατάφερε να εκτιμήσει τις πραγματικές τιμές με μικρό ποσοστό σφάλματος.

Ωστόσο, παρά το φαινομενικά χαμηλό RMSE (0.01586) και το MAPE (1.26064), τα αποτελέσματα δεν ήταν εντελώς ικανοποιητικά, καθώς η πραγματική δυσκολία στην πρόβλεψη των τιμών των υγρών καυσίμων αποδεικνύεται από τη διακύμανση των προβλέψεων στο γράφημα που παρουσιάστηκε.

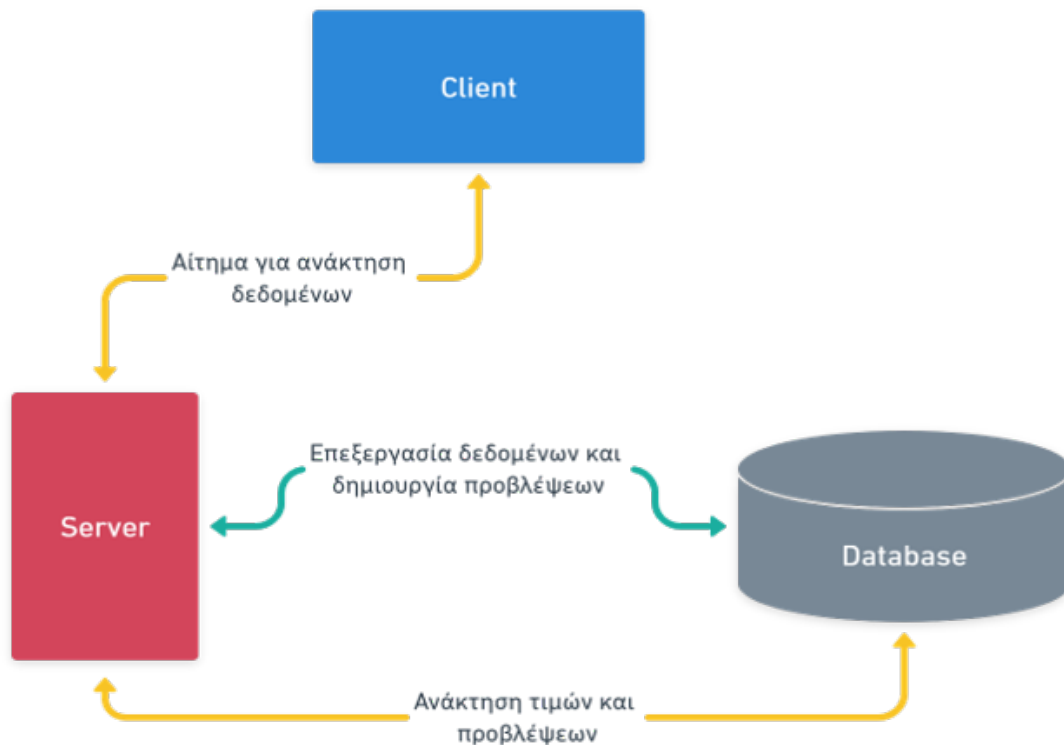
Το μοντέλο ARIMA, παρότι είναι ένα παραδοσιακό και ευρέως διαδεδομένο μοντέλο για την ανάλυση χρονοσειρών, παρουσιάζει περιορισμούς όταν αντιμετωπίζει δεδομένα με πολύπλοκες δυναμικές, όπως είναι οι τιμές των υγρών καυσίμων. Ενώ το μοντέλο μπορεί να είναι σε θέση να εκμεταλλευτεί στατιστικές ιδιότητες των δεδομένων όπως ο μέσος όρος και η τυπική απόκλιση, ενδέχεται να μην είναι εξίσου επιδέξιο στην εντοπισμό και την πρόβλεψη των πιο εξελικτικών δυναμικών, που προκαλούνται από πολλές αλληλεπιδρώντες μεταβλητές, όπως η πολιτική, η οικονομική κατάσταση, οι καιρικές συνθήκες, και άλλοι παγκόσμιοι παράγοντες που μπορούν να επηρεάσουν την τιμή των καυσίμων.

Έτσι, παρότι τα αριθμητικά αποτελέσματα φαίνονται αρκετά καλά, η οπτική επιβεβαίωση των προβλέψεων που παρουσιάζεται στο γράφημα αποδεικνύει ότι υπάρχει σαφώς ακόμη περιθώριο για βελτίωση.

Κεφάλαιο 6ο: Υλοποίηση του FuelPricesApp

Στο παρόν κεφάλαιο, θα διερευνήσουμε τη διαδικασία υλοποίησης και ανάπτυξης της εφαρμογής μαζί με παραδείγματα κώδικα. Η ανάπτυξη μιας τέτοιας εφαρμογής είναι μια σύνθετη διαδικασία που απαιτεί προσεκτικό σχεδιασμό, ανάπτυξη κώδικα, επιλογή κατάλληλων τεχνολογιών και αρκετή δοκιμή.

6.1 Αρχιτεκτονική του FuelPricesApp



Σχήμα 6.1: Διάγραμμα αρχιτεκτονικής του FuelPricesApp

6.2 Back-End

Στην παρούσα ενότητα, θα εστιάσουμε στο backend τμήμα της εφαρμογής FuelPricesApp. Η τεχνολογία που χρησιμοποιήθηκε για την υλοποίηση του backend είναι κυρίως τη γλώσσα προγραμματισμού Python, καθώς και το FastAPI, ένα σύγχρονο framework για τη δημιουργία APIs. Η επιλογή αυτής της τεχνολογίας βασίστηκε στην αποδοτικότητα, την ευελιξία και την ευκολία χρήσης που παρέχει. Το θεωρητικό κομμάτι των τεχνολογιών αυτών καλύφθηκε στο κεφάλαιο 4.

Το backend της εφαρμογής διακρίνεται σε δύο βασικά μέρη. Το πρώτο μέρος αφορά την προεπεξεργασία των δεδομένων και την υλοποίηση των προβλέψεων. Σε αυτό το μέρος, γίνεται η συλλογή και η εκκαθάριση των δεδομένων, η προετοιμασία τους για την ανάλυση και την εφαρμογή των μοντέλων πρόβλεψης, όπως το ARIMA που περιγράφηκε στο κεφάλαιο 5.

Το δεύτερο μέρος αφορά την υλοποίηση του API της εφαρμογής με την βοήθεια του FastAPI. Αυτό περιλαμβάνει τη δημιουργία των endpoints του API, τη διαχείριση των αιτημάτων και των απαντήσεων, καθώς και την επικοινωνία με τη βάση δεδομένων. Μέσω του API, το frontend της εφαρμογής είναι σε θέση να αντλεί τα απαραίτητα δεδομένα, να εκτελεί διάφορες λειτουργίες και να επιστρέφει αποτελέσματα στον χρήστη.

Συνολικά, το backend της εφαρμογής είναι η “καρδιά” του FuelPricesApp, καθώς είναι υπεύθυνο για την εκτέλεση των πιο σημαντικών λειτουργιών και διεργασιών, από την προεπεξεργασία των δεδομένων και την υλοποίηση των προβλέψεων, μέχρι την παροχή των δεδομένων στο frontend μέσω του API.

Καταρχάς, θα εξετάσουμε την υλοποίηση του API της εφαρμογής μας στην Python. Το API, ή Application Programming Interface, είναι ένα σύνολο κανόνων και πρωτοκόλλων που διευκολύνουν την επικοινωνία μεταξύ διαφορετικών συστημάτων λογισμικού. Στην περίπτωση της FuelPricesApp, το API μας επιτρέπει να συνδέσουμε το backend με το frontend, επιτρέποντας την ανταλλαγή δεδομένων μεταξύ των δύο.

Το API μας υλοποιήθηκε με τη βοήθεια του FastAPI, ένα πανίσχυρο και ευέλικτο web framework για τη δημιουργία APIs. Αυτό το framework μας επιτρέπει να δημιουργήσουμε διάφορα “endpoints”, τα οποία είναι σημεία πρόσβασης στο API μας όπου οι χρήστες (ή στην πραγματικότητα, το frontend της εφαρμογής μας) μπορούν να στέλνουν αιτήματα και να λαμβάνουν απαντήσεις.

Τα endpoints που έχουμε υλοποιήσει είναι τα εξής:

- *get('/')*: Αυτό το endpoint είναι το βασικό σημείο πρόσβασης στο API μας και επιστρέφει μια βασική απάντηση για να επιβεβαιώσει ότι το API λειτουργεί σωστά καθώς και την αρχική σελίδα της εφαρμογής
- *get('/predictions')*: Αυτό το endpoint επιστρέφει την δεύτερη σελίδα της εφαρμογής καθώς και τις περιοχές, τα καύσιμα και την περίοδο πρόβλεψης
- *post('/currentValues')*: Αυτό το endpoint δέχεται δεδομένα για την ανάλυση των τρεχόντων τιμών των καυσίμων
- *post('/predictedValues')*: Αυτό το endpoint δέχεται δεδομένα για την εκτέλεση προβλέψεων

Κάθε ένα από αυτά τα endpoints παίζει σημαντικό ρόλο στη λειτουργία της εφαρμογής μας και θα τα αναλύσουμε ξεχωριστά, αναφέροντας τη λειτουργικότητά τους, τα δεδομένα που ανταλλάσσουν και το πώς συμβάλλουν στη συνολική διεπαφή της εφαρμογής. Αυτή η επεξήγηση είναι κρίσιμη για την κατανόηση του πώς τα διάφορα στοιχεία της εφαρμογής μας διασυνδέονται και συνεργάζονται για να παράγουν την τελική λειτουργικότητα. Ας δούμε τώρα πιο αναλυτικά την υλοποίηση του server στην Python.

Αρχικά, φτιάξαμε το αρχείο `retrieveDataFromDatabase.py` το οποίο θα περιέχει όλες τις απαραίτητες μεθόδους για την επικοινωνία και την ανταλλαγή δεδομένων με την βάση. Ως παράδειγμα θα χρησιμοποιηθούν μερικές από τις μεθόδους του αρχείου:

```
import pandas as pd
import numpy as np
from sqlalchemy import create_engine
import mysql.connector as mariadb
import sys
import os
sys.path.append(os.path.abspath(os.path.join(
    os.path.dirname(__file__), '..', 'database')))
import connection_properties as conn

def get_connection():
    return mariadb.connect(
        user=conn.USER,
        password=conn.PASSWORD,
        host=conn.HOST,
        database=conn.DATABASE,
        auth_plugin='mysql_native_password'
    )

def retrieve_data(connection, sql):
    data = pd.read_sql(sql, connection).to_dict('records')
    return data

def retrieve_specific_region_prices(fuel, region_code):
    connection = get_connection()
    sql = f"SELECT record_date, `{fuel}`
           FROM interpolated_prices
           WHERE county_code='{region_code}'"
    data = retrieve_data(connection, sql)
    connection.close()
    return data
```

Όπου :

- `get_connection()`: Αυτή η μέθοδος επιστρέφει μια σύνδεση με τη βάση δεδομένων. Χρησιμοποιεί τις παραμέτρους σύνδεσης (όπως το όνομα χρήστη, τον κωδικό πρόσβασης, τον ξενιστή και το όνομα της βάσης δεδομένων) που ορίζονται στο αρχείο "connection_properties.py" για να δημιουργήσει τη σύνδεση

- *retrieve_data(connection, sql)*: Αυτή η μέθοδος δέχεται μια σύνδεση βάσης δεδομένων και μια ερώτηση SQL ως είσοδο. Χρησιμοποιεί τη βιβλιοθήκη pandas για να εκτελέσει την ερώτηση SQL και να επιστρέψει τα αποτελέσματα ως μια λίστα λεξικών, όπου κάθε λεξικό αντιπροσωπεύει μια εγγραφή στα αποτελέσματα
- *retrieve_specific_region_prices(fuel, region_code)*: Αυτή η μέθοδος δέχεται έναν κωδικό καυσίμου και έναν κωδικό περιοχής ως είσοδο. Χρησιμοποιεί τη μέθοδο get_connection() για να δημιουργήσει μια σύνδεση με τη βάση δεδομένων και στη συνέχεια χρησιμοποιεί τη μέθοδο retrieve_data() για να ανακτήσει τις τιμές του καυσίμου για την εν λόγω περιοχή. Τέλος, η μέθοδος κλείνει τη σύνδεση με τη βάση δεδομένων και επιστρέφει τα δεδομένα

Στην συνέχεια δημιουργήσαμε το αρχείο `server.py` το οποίο περιέχει το API της εφαρμογής. Ως πρώτο βήμα έγινε εισαγωγή των απαραίτητων βιβλιοθηκών.

```
from fastapi import FastAPI, Request
from fastapi.responses import HTMLResponse
from fastapi.templating import Jinja2Templates
from fastapi.staticfiles import StaticFiles
```

Όπου :

- *FastAPI*: το framework για την υλοποίηση του server
- *Request*: στο FastAPI είναι μια απεικόνιση του HTTP request που κάνει ο client στον server. Περιέχει πληροφορίες όπως headers, cookies, query parameters, και άλλα
- *HTMLResponse*: στο FastAPI είναι ένας τύπος απόκρισης που επιστρέφει HTML στον client. Χρησιμοποιείται κυρίως για την απευθείας επιστροφή HTML σελίδων
- *Jinja2Templates*: είναι ένα χαρακτηριστικό του FastAPI που επιτρέπει τη χρήση του Jinja2, ενός δημοφιλούς συστήματος για τη δημιουργία templates σε Python, το οποίο επιτρέπει την ενσωμάτωση δεδομένων Python μέσα σε HTML
- *StaticFiles*: είναι μια δυνατότητα του FastAPI που επιτρέπει την εξυπηρέτηση στατικών αρχείων (π.χ., CSS, JavaScript, εικόνες) από τον server στον client

Αφού φορτωθούν με επιτυχία οι παραπάνω βιβλιοθήκες, δημιουργούμε ένα αντικείμενο τύπου FastAPI, που αποθηκεύεται στη μεταβλητή app. Αυτό το αντικείμενο είναι το κεντρικό σημείο της εφαρμογής και είναι υπεύθυνο για τη διαχείριση των HTTP αιτημάτων και την παροχή των κατάλληλων απαντήσεων.

```
app = FastAPI()
```

Στη συνέχεια, δημιουργείται ένα αντικείμενο Jinja2Templates και αποθηκεύεται στη μεταβλητή templates. Αυτό το αντικείμενο είναι ουσιαστικά ένας διαχειριστής προτύπων που μπορεί να φορτώσει τα αρχεία HTML από τον κατάλογο "htmlPages" και να τα χρησιμοποιήσει για τη δημιουργία δυναμικών HTML σελίδων.

```
templates = Jinja2Templates(directory="htmlPages")
```

Έπειτα καλούμε τη μέθοδο app.mount που χρησιμοποιείται για να "προσαρτήσει" έναν κατάλογο στατικών αρχείων σε ένα συγκεκριμένο μονοπάτι της εφαρμογής. Σε αυτήν την περίπτωση, τα στατικά αρχεία είναι διαθέσιμα στον κατάλογο "htmlPages/static".

```
app.mount(
    "/htmlPages/static",
    StaticFiles(directory=Path(__file__).parent.absolute() /
                 "htmlPages" / "static"),
    name="static",
)
```

Τέλος, δημιουργείται το πρώτο endpoint για την αρχική σελίδα της εφαρμογής με τον διακομιστή. Αυτή η διαδρομή εξυπηρετείται με το HTTP GET στο μονοπάτι "/" (το οποίο αντιστοιχεί στην αρχική σελίδα της εφαρμογής). Ο server ανταποκρίνεται σε αιτήματα σε αυτό το μονοπάτι εκτελώντας τη συνάρτηση home_page. Η συνάρτηση δέχεται ένα αίτημα ως παράμετρο (το οποίο παρέχεται αυτόματα από το FastAPI) και επιστρέφει μια δυναμικά δημιουργημένη HTML σελίδα. Η συνάρτηση χρησιμοποιεί τον διαχειριστή προτύπων templates που δημιουργήθηκε νωρίτερα για να φορτώσει το αρχείο προτύπου "homePage.html", να το συμπληρώσει με τα δεδομένα που παρέχονται (σε αυτήν την περίπτωση, μόνο το αντικείμενο αίτησης) και να επιστρέψει το αποτέλεσμα ως μια απάντηση HTML.

```
@app.get("/")
async def home_page(req: Request):
    return templates.TemplateResponse("homePage.html", {"request": req})
```

Ως δεύτερο endpoint είναι ένα HTTP GET που ανταποκρίνεται στη διεύθυνση "/predictions" του διακομιστή. Όπως και το πρώτο endpoint, αντί για JSON δεδομένα, το "/predictions" επιστρέφει μια HTML σελίδα ως απάντηση. Αυτή η σελίδα παράγεται με τη χρήση του Jinja2, το οποίο είναι ένα σύστημα προτύπων για τη γλώσσα Python.

Πιο συγκεκριμένα, το endpoint κάνει τα εξής:

- Ανακτά τα δεδομένα που χρειάζεται για την προβλεπόμενη σελίδα: τις περιοχές (regions), τα καύσιμα (fuels) και την περίοδο πρόβλεψης (forecast_period). Αυτά τα δεδομένα ανακτώνται μέσω των αντίστοιχων συναρτήσεων που βρίσκονται στο αρχείο "retrieveDataFromDatabase.py" που αναφέρθηκε παραπάνω
- Χρησιμοποιεί το Jinja2 για να δημιουργήσει την HTML σελίδα "predictions.html", περνώντας τα απαραίτητα δεδομένα (το request, τις περιοχές, τα καύσιμα και την περίοδο πρόβλεψης) ως παραμέτρους στο template

Τα δεδομένα που περνιούνται στο template χρησιμοποιούνται για να γεμίσουν τα αντίστοιχα μέρη της σελίδας HTML. Με αυτόν τον τρόπο, η σελίδα μπορεί να προσαρμοστεί δυναμικά ανάλογα με τα δεδομένα που ανακτώνται από τη βάση δεδομένων.

Για παράδειγμα, το πεδίο "regions" της σελίδας HTML περιλαμβάνει επιλογές για κάθε περιοχή που επιστρέφεται από τη βάση δεδομένων. Ομοίως, το πεδίο "fuels" περιλαμβάνει επιλογές για κάθε τύπο καυσίμου που επιστρέφεται από τη βάση δεδομένων.

Τέλος, το endpoint επιστρέφει την HTML σελίδα ως απάντηση στο αίτημα του πελάτη. Αυτό σημαίνει ότι όταν ο χρήστης πλοηγείται στη διεύθυνση "/predictions" του διακομιστή, θα δει τη σελίδα προβλέψεων που περιέχει δυναμικά παραγόμενες επιλογές βάσει των δεδομένων που έχουν ανακτηθεί από τη βάση δεδομένων.

```

@app.get("/predictions", response_class=HTMLResponse)
async def predictions(request: Request):
    regions = actualData.retrieve_regions()
    fuels = actualData.retrieve_fuels()
    forecast_period = actualData.retrieve_forecastPeriod()

    return templates.TemplateResponse("predictions.html",
                                     {"request": request,
                                      "regions": regions,
                                      "fuels": fuels,
                                      "predictor_period": forecast_period})

```

Το επόμενο βήμα, είναι ένα HTTP POST endpoint στον server με το μονοπάτι `"/currentValues"`. Αυτό το endpoint αναμένει να λάβει JSON δεδομένα στο σώμα του αιτήματος, τα οποία μετατρέπονται αυτόματα σε λεξικό Python (dict) από το FastAPI.

Τα JSON δεδομένα που αναμένεται να λάβει αυτό το endpoint πρέπει να περιέχουν δύο κλειδιά: `'fuel'` και `'region'`. Το `'fuel'` θα πρέπει να αντιστοιχεί σε έναν τύπο καυσίμου (για παράδειγμα, `"unleaded_95"` ή `"motor_oil"`), ενώ το `'region'` πρέπει να αντιστοιχεί σε μια λίστα με κωδικούς περιοχής.

Για κάθε κωδικό περιοχής στη λίστα `'region'`, η συνάρτηση εκτελεί τις εξής εργασίες:

- Καλεί τη συνάρτηση `retrieve_specific_region_prices` για να λάβει τις τρέχουσες τιμές των καυσίμων για τον συγκεκριμένο τύπο καυσίμου στη συγκεκριμένη περιοχή
- Καλεί τη συνάρτηση `retrieve_region_name` για να λάβει το όνομα της περιοχής που αντιστοιχεί στον κωδικό περιοχής
- Προσθέτει το όνομα της περιοχής στη λίστα `region_names_list`
- Δημιουργεί μια νέα λίστα με τις τρέχουσες τιμές των καυσίμων για την περιοχή και την προσθέτει στη λίστα `current_fuel_prices_list`
- Δημιουργεί μια νέα λίστα με τις ημερομηνίες καταγραφής για κάθε τιμή καυσίμου, μετατρέπει κάθε ημερομηνία σε μορφή string ('Y-M-D') και την προσθέτει στη λίστα `current_dates_list`

Τέλος, η συνάρτηση επιστρέφει ένα λεξικό με τρεις λίστες: τη λίστα με τα ονόματα των περιοχών (`region_names_list`), τη λίστα με τις τρέχουσες τιμές των καυσίμων για κάθε περιοχή (`current_fuel_prices_list`), και τη λίστα με τις ημερομηνίες καταγραφής για κάθε τιμή καυσίμου (`current_dates_list`).

Εδώ πρέπει να σημειωθεί ότι η σειρά των στοιχείων σε κάθε λίστα αντιστοιχεί στη σειρά των περιοχών στη λίστα 'region' που παρέχεται στο αίτημα. Δηλαδή, το πρώτο στοιχείο στη region_names_list αντιστοιχεί στην πρώτη περιοχή στη λίστα 'region', το πρώτο στοιχείο στη current_fuel_prices_list αντιστοιχεί στις τρέχουσες τιμές καυσίμου για την πρώτη περιοχή, και ούτω καθεξής.

```
@app.post('/currentValues')
async def display_current_prices(data: dict):

    fuel = data['fuel']
    regions = data['region']

    current_fuel_prices_list = []
    current_dates_list = []
    region_names_list = []

    for region_code in regions:

        current_data = actualData.retrieve_specific_region_prices(
            fuel, region_code)

        region_dict = actualData.retrieve_region_name(region_code)
        region_name = region_dict[0]['name']

        region_names_list.append(region_name)

        current_fuel_prices = [d[fuel] for d in current_data]
        current_fuel_prices_list.append(current_fuel_prices)

        current_dates = [d['record_date'] for d in current_data]
        current_dates = [datetime.strptime(
            date, '%Y-%m-%d') for date in current_dates]
        current_dates_list.append(current_dates)

    return {
        'region_names_list': region_names_list,
        'current_fuel_prices_list': current_fuel_prices_list,
        'current_dates_list': current_dates_list,
    }
```

Το τελευταίο endpoint που δημιουργήσαμε είναι το `"/predictedValues"`, επίσης μια μέθοδος HTTP POST. Αυτό το endpoint δέχεται ένα λεξικό με τον τύπο του καυσίμου (fuel) και την περίοδο πρόβλεψης (forecastPeriod), και στη συνέχεια επιστρέφει τις τρέχουσες τιμές του καυσίμου, τις προβλεπόμενες τιμές και τις μετρήσεις αξιολόγησης της πρόβλεψης. Ας εξετάσουμε λίγο πιο αναλυτικά τι κάνει αυτό το endpoint:

- Διαβάζει την τιμή fuel και την forecastPeriod από τα εισερχόμενα δεδομένα του αιτήματος. Εδώ, το fuel αναφέρεται στον τύπο του καυσίμου (π.χ. "autogas"), ενώ το forecastPeriod αναφέρεται στην περίοδο πρόβλεψης (π.χ. 30, 90)
- Ορίζει το region_code ως 0 και το region_name ως 'Ελλάδα'. Αυτό σημαίνει ότι οι προβλέψεις και οι τρέχουσες τιμές που θα ανακτηθούν θα αφορούν την εθνική αγορά καυσίμων
- Ανακτά τις προβλέψεις για τις τιμές των καυσίμων για την επιλεγμένη περίοδο πρόβλεψης, χρησιμοποιώντας τη μέθοδο `retrieve_predictions()`. Οι προβλέψεις αντιστοιχούν στις εκτιμώμενες τιμές των καυσίμων για την επόμενη περίοδο
- Συλλέγει τις ημερομηνίες και τις τιμές των καυσίμων από τα προβλεπόμενα δεδομένα και τα μετατρέπει σε σειρές, ετοιμάζοντάς τα για να επιστραφούν στον client
- Ανακτά τα δεδομένα αξιολόγησης των προβλέψεων για την επιλεγμένη περίοδο πρόβλεψης, χρησιμοποιώντας τη συνάρτηση `retrieve_evaluations()`. Αυτά τα δεδομένα περιλαμβάνουν μετρήσεις όπως το Mean Absolute Percentage Error (MAPE) και το Root Mean Squared Error (RMSE), που είναι δύο μετρήσεις που επιλέχθηκαν για την αξιολόγηση της ακρίβειας των προβλέψεων
- Επιστρέφει ένα λεξικό που περιλαμβάνει το όνομα της περιοχής, τις τρέχουσες τιμές των καυσίμων, τις τρέχουσες ημερομηνίες, την περίοδο πρόβλεψης, τις προβλεπόμενες τιμές των καυσίμων, τις προβλεπόμενες ημερομηνίες, καθώς και τις μετρήσεις αξιολόγησης των προβλέψεων (MAPE και RMSE). Αυτά τα δεδομένα μπορούν να χρησιμοποιηθούν από τον client για να εμφανίσει γραφήματα των τρέχουσων και προβλεπόμενων τιμών των καυσίμων, καθώς και για να παρέχει ενημέρωση σχετικά με την ακρίβεια των προβλέψεων

```
@app.post('/predictedValues')
async def make_predictions(data: dict):

    fuel = data['fuel']
    forecastPeriod = data['forecastPeriod']
    region_code = 0
    region_name = 'Ελλάδα'

    current_data = actualData.retrieve_specific_region_prices(
        fuel, region_code)
    current_fuel_prices = [d[fuel] for d in current_data]
    current_dates = [d['record_date'] for d in current_data]
    current_dates = [datetime.strptime(
        date, '%Y-%m-%d') for date in current_dates]
```

```

predicted_data = actualData.retrieve_predictions(
    fuel, region_code, forecastPeriod)

predicted_fuel_prices = [d['price'] for d in predicted_data]

predicted_dates = [d['record_date'] for d in predicted_data]
predicted_dates = [datetime.strptime(
    date, '%Y-%m-%d') for date in predicted_dates]

evaluations_data = actualData.retrieve_evaluations(
    fuel, region_code, forecastPeriod)

mape = evaluations_data[0]['mape']
rmse = evaluations_data[0]['rmse']

return {
    'region_name': region_name,
    'current_fuel_prices': current_fuel_prices,
    'current_dates': current_dates,
    'forecastPeriod': forecastPeriod,
    'predicted_fuel_prices': predicted_fuel_prices,
    'predicted_dates': predicted_dates,
    'mape': mape,
    'rmse': rmse
}

```

6.3 Front-End

Γραφική διεπαφή :

Στον σχεδιασμό του front-end της εφαρμογής, χρησιμοποιήθηκαν διάφορες τεχνολογίες για να καταστεί δυνατή η επίτευξη των στόχων της. Η HTML και η CSS ήταν οι βασικές γλώσσες που χρησιμοποιήθηκαν για τη δημιουργία και τον σχεδιασμό των ιστοσελίδων, ενώ το JavaScript προσέθεσε δυναμική λειτουργικότητα. Το πλαίσιο Bootstrap επέτρεψε τη δημιουργία ενός αισθητικά ευχάριστου και αποκρίσιμου σχεδιασμού. Τέλος, για την παρουσίαση των δεδομένων σε γραφήματα, χρησιμοποιήθηκε η βιβλιοθήκη ApexCharts.js.

Όλες αυτές οι τεχνολογίες καλύφθηκαν θεωρητικά στο κεφάλαιο 4 της εργασίας. Στην συνέχεια, θα δούμε πως αυτές εφαρμόστηκαν στην πράξη μέσω παραδειγμάτων κώδικα και συγκεκριμένων εξηγήσεων της υλοποίησης.

Για την γραφική διεπαφή της εφαρμογής δημιουργήθηκαν δύο ιστοσελίδες, η `homePage.html` και η `predictions.html`, οι οποίες χρησιμοποιούνται για την αρχική σελίδα του FuelPricesApp καθώς και για την παρουσίαση και την πρόβλεψη τιμών καυσίμων αντίστοιχα. Οι σελίδες περιλαμβάνουν στοιχεία της Bootstrap για τη δημιουργία της διάταξης, ενώ τα απαραίτητα CSS και JavaScript αρχεία φορτώνονται μέσω CDN (Content Delivery Network) και τοπικών αρχείων.

Ένα αξιοσημείωτο κομμάτι στην σελίδα `predictions.html` είναι τα στοιχεία `<select>` στην ιστοσελίδα όπου δίνουν στον χρήστη τη δυνατότητα να επιλέξει διάφορες επιλογές από μια λίστα. Υπάρχουν τρία τέτοια στοιχεία, τα οποία αφορούν την επιλογή του καυσίμου, την επιλογή της περιοχής και την επιλογή της περιόδου πρόβλεψης. Οι επιλογές στα `<select>` στοιχεία γεμίζονται μέσω του FastApi server, όπως φαίνεται από τον κώδικα Jinja2 (`% for ... %` και `% endfor %`). Αυτό σημαίνει ότι οι επιλογές που εμφανίζονται στον χρήστη εξαρτώνται από τα δεδομένα που λαμβάνονται από τον διακομιστή, επιτρέποντας μια δυναμική αλλαγή των επιλογών με βάση τις διαθέσιμες πληροφορίες.

```
<select id="fuel" class="form-select" aria-label="Default select example">
  {% for fuel in fuels %}
    <option value="{{ fuel.fuel }}">{{ fuel.frontend_names }}</option>
  {% endfor %}
</select>
```

...

```
<select id="region" class="form-select" aria-label="Default select example"
  multiple data-choice="true">
  {% for region in regions %}
    {% if loop.first %}
      <option value="{{ region.code }}" selected>{{ region.name }}</option>
    {% else %}
      <option value="{{ region.code }}">{{ region.name }}</option>
    {% endif %}
  {% endfor %}
</select>
```

...

```
<select id="forecastPeriod" class="form-select" aria-label="Default select example">
  {% for period in predictor_period %}
    <option value="{{ period.period }}">{{ period.frontend_names }}</option>
  {% endfor %}
</select>
```

Για την παρουσίαση των τιμών των υγρών καυσίμων αλλά και για τις προβλέψεις, δημιουργήθηκαν δύο διαφορετικά κουμπιά, "Παρουσίαση" και "Πρόβλεψη" όπου χρησιμοποιούνται για την αποστολή των επιλεγμένων πληροφοριών πίσω στον server και τη λήψη των αντίστοιχων αποτελεσμάτων πρόβλεψης ή παρουσίασης τιμών.

Τέλος, υπάρχει ένα κενό στοιχείο `<div>` με το id "myChart", το οποίο χρησιμοποιείται για την εμφάνιση των γραφημάτων, χρησιμοποιώντας την βιβλιοθήκη ApexCharts.js, η οποία φορτώνεται στο τέλος του κώδικα.

```
<button id="currentPricesBtn" type="submit" class="btn btn-primary">
    Παρουσίαση
</button>

<button id="predictBtn" type="submit" class="btn btn-primary">
    Πρόβλεψη
</button>

...

<div id="myChart"></div>
```

Λειτουργικότητα διεπαφής :

Όπως αναφέρθηκε και στην αρχή της συγκεκριμένης ενότητας, για την λειτουργικότητα της ιστοσελίδας μας χρησιμοποιήθηκε η γλώσσα προγραμματισμού JavaScript. Μέσω της JavaScript, επιτυγχάνεται η αλληλεπίδραση της εφαρμογής με τους χρήστες καθώς και η επικοινωνία του frontend με τον server. Η επικοινωνία αυτή βασίζεται σε αντικείμενα τύπου JSON, μία δημοφιλής μορφή αρχείων που χρησιμοποιείται για τη μεταφορά και ανταλλαγή δεδομένων μεταξύ εφαρμογών.

Μια σημαντική τεχνική που χρησιμοποιείται στην εφαρμογή μας, είναι η ακρόαση για το γεγονός 'DOMContentLoaded'. Αυτό το γεγονός πυροδοτείται όταν το αρχικό DOM (Document Object Model) της HTML σελίδας είναι πλήρως φορτωμένο και αναλυμένο, πράγμα που σημαίνει ότι όλα τα στοιχεία της σελίδας είναι πλέον προσβάσιμα από τον κώδικα JavaScript.

Αμέσως μετά την ενεργοποίηση του γεγονότος 'DOMContentLoaded', ο κώδικάς μας πραγματοποιεί μια σειρά διαδικασιών που προετοιμάζουν την ιστοσελίδα για την αλληλεπίδραση με τον χρήστη. Ειδικά, κατασκευάζει μια νέα εμφάνιση Choices για το στοιχείο που έχει το id 'region'. Η σκοπιμότητα αυτής της κατασκευής είναι να παρέχει στον χρήστη την ικανότητα να διαλέξει αρκετούς νομούς της Ελλάδας για να σχηματίσει τις τιμές των καυσίμων σε διάγραμμα, με σκοπό να μπορεί να συγκρίνει την τιμή κάποιου καυσίμου σε διάφορες περιοχές της Ελλάδας.

Η Choices είναι μια βιβλιοθήκη JavaScript που προσθέτει ευέλικτες και διαμορφώσιμες λειτουργίες επιλογής σε στοιχεία εισαγωγής και επιλογής HTML. Εδώ, το στοιχείο 'region' είναι ένα στοιχείο επιλογής που επιτρέπει στους χρήστες να επιλέξουν από μια λίστα διαθέσιμων περιοχών. Με την εφαρμογή της Choices, αυτό το στοιχείο επιλογής γίνεται πολύ πιο διαδραστικό και ευχάριστο για τους χρήστες, προσφέροντας λειτουργίες όπως η αναζήτηση, η πολλαπλή επιλογή και η εξατομίκευση της εμφάνισης.

```
document.addEventListener('DOMContentLoaded', function () {

  const element = document.getElementById('region');
  const choices = new Choices(element, {
    removeItemButton: true,
    itemSelectText: '',
    searchEnabled: false,
    placeholder: true,
    maxItemCount: 5,
    allowHTML: true
  });
  ...
});
```

Επιπλέον, οι χρήστες μπορούν να αλληλεπιδράσουν με την ιστοσελίδα μας κάνοντας κλικ σε δύο κουμπιά: το 'predictBtn' και το 'currentPricesBtn'. Κάθε ένα από αυτά τα κουμπιά έχει μια λειτουργία "ακρόασης" που περιμένει να γίνει κλικ από τον χρήστη.

```
...
const CurrentPricesButton = document.getElementById('currentPricesBtn');
CurrentPricesButton.addEventListener('click', sendCurrentPrices);

const predictionsButton = document.getElementById('predictBtn');
predictionsButton.addEventListener('click', sendPredictions);
...

```

Όταν το κουμπί 'currentPricesBtn' πατηθεί, ενεργοποιείται η μέθοδος 'sendCurrentPrices'. Αντίστοιχα, όταν το κουμπί 'predictBtn' πατηθεί, ενεργοποιείται η μέθοδος 'sendPredictions'. Αυτές οι συναρτήσεις διαδραματίζουν κρίσιμο ρόλο στην αλληλεπίδραση του χρήστη με την ιστοσελίδα.

- *sendCurrentPrices()*:

```

...
function sendCurrentPrices() {
  const selected_fuel = document.getElementById("fuel");
  const fuel_code = selected_fuel.options[selected_fuel.selectedIndex].value;

  const region_codes = getSelectedRegions();

  const data = {
    fuel: fuel_code,
    region: region_codes
  };

  fetch('/currentValues', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(data)
  })
  .then(response => response.json())
  .then(({ region_names_list,
    current_fuel_prices_list,
    current_dates_list }) => {

    const colors = ['#000000', '#ffe02f', '#366720', '#ed41a6', '#c57a04']

    let currentDataSeries = region_names_list.map((region_name, index) => {
      const xArray = [...current_dates_list[index]];
      const yArray = [...current_fuel_prices_list[index]];
      const formattedData = formatData(yArray, xArray);
      return {
        name: region_name,
        data: formattedData,
        color: colors[index % colors.length]
      };
    });

    chart.updateOptions({
      series: currentDataSeries
    });
  })
  .catch(error => console.error(error));
};
...

```

- *sendPredictions()*:

...

```
function sendPredictions() {

  const selected_fuel = document.getElementById("fuel");
  const fuel_code = selected_fuel.options[selected_fuel.selectedIndex].value;
  const selected_forecastPeriod = document.getElementById("forecastPeriod");
  const forecastPeriod_days = selected_forecastPeriod
    .options[selected_forecastPeriod.selectedIndex]
    .value;

  const data = {
    fuel: fuel_code,
    forecastPeriod: forecastPeriod_days
  };

  fetch('/predictedValues', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(data)
  })
  .then(response => response.json())
  .then(({ region_name,
    current_fuel_prices,
    current_dates,
    predicted_fuel_prices,
    predicted_dates,
    mape,
    rmse }) => {

    const currentDataSeries = [{
      name: region_name,
      data: formatData(current_fuel_prices, current_dates),
      color: '#000000'
    }];

    const [firstHalf, secondHalf] = splitArrayInHalf(predicted_fuel_prices);
    const formattedFirstHalf = formatData(firstHalf,
      predicted_dates
      .slice(0, firstHalf.length));
    const formattedSecondHalf = formatData(secondHalf,
      predicted_dates
      .slice(firstHalf.length));
```

```

const predictedDataSeries = [
  {
    name: `${region_name} Existing Predictions`,
    data: formattedFirstHalf,
    color: '#FF0000'
  },
  {
    name: `${region_name} Future Predictions`,
    data: formattedSecondHalf,
    color: '#00FF00'
  }
];

const series = [...currentDataSeries, ...predictedDataSeries];

chart.updateOptions({
  series: series
});

const accuracy_index_mape_container = document
  .getElementById("accuracy_index_mape_container");
accuracy_index_mape_container.innerHTML = '';
const p1 = document.createElement("p");
p1.className = "text-center mb-0";
p1.id = "accuracy_index_mape";
p1.innerHTML = "Δείκτης Αξιοπιστίας (Mean Absolute Percentage Error) - "
  + region_name + ": " + mape.toFixed(2) + "%";
accuracy_index_mape_container.appendChild(p1);

const accuracy_index_rmse_container = document
  .getElementById("accuracy_index_rmse_container");
accuracy_index_rmse_container.innerHTML = '';
const p2 = document.createElement("p");
p2.className = "text-center mb-0";
p2.id = "accuracy_index_rmse";
p2.innerHTML = "Δείκτης Αξιοπιστίας (Root Mean Square Error) - "
  + region_name + ": " + rmse.toFixed(2) + "€";
accuracy_index_rmse_container.appendChild(p2);
})
.catch(error => console.error(error));
};
});

```

Κάθε μέθοδος ξεκινά συλλέγοντας τα απαραίτητα δεδομένα από την φόρμα της σελίδας. Αυτά τα δεδομένα μπορεί να περιλαμβάνουν την επιλογή του τύπου καυσίμου, την περίοδο πρόβλεψης και την επιλεγμένη περιοχή ή επιλεγμένες περιοχές. Αυτά τα δεδομένα συλλέγονται και οργανώνονται σε ένα αντικείμενο JSON.

Μετά την οργάνωση των δεδομένων, η μέθοδος κάνει μια αίτηση POST στον server. Αυτό σημαίνει ότι τα δεδομένα στέλνονται στον server για επεξεργασία. Η αίτηση POST χρησιμοποιεί τη μέθοδο 'fetch', η οποία επιτρέπει την αποστολή δεδομένων στον server μέσω HTTP. Η αίτηση POST επιστρέφει μια απάντηση, η οποία περιέχει τις πληροφορίες που ζήτησε ο χρήστης. Η απάντηση αυτή, όταν φτάσει στον client, μετατρέπεται από την αρχική της μορφή σε μια μορφή JSON, την οποία μπορεί να αναγνωρίσει και να επεξεργαστεί ο κώδικας JavaScript της ιστοσελίδας. Αυτό το μετατρεπόμενο JSON χρησιμοποιείται στη συνέχεια για να ενημερωθούν δυναμικά στοιχεία της ιστοσελίδας, όπως τα διαγράμματα τιμών καυσίμου και οι δείκτες ακρίβειας προβλέψεων.

Τα διαγράμματα τιμών καυσίμου μπορούν να ανανεωθούν με νέες τιμές, που προκύπτουν από την ανάλυση των δεδομένων από τον server. Αυτό μπορεί να περιλαμβάνει την πρόβλεψη τιμών καυσίμου για μια συγκεκριμένη περίοδο ή την εμφάνιση των τρεχόντων τιμών. Οι δείκτες ακρίβειας προβλέψεων ενημερώνονται επίσης με βάση την απόκριση του server. Αυτό μπορεί να περιλαμβάνει πληροφορίες σχετικά με το πόσο ακριβείς ήταν οι προβλέψεις των τιμών των καυσίμων σε σχέση με τις πραγματικές τιμές.

Ο κώδικας JavaScript συμπληρώνεται με την προσθήκη της κατασκευής και της αποστολής του διαγράμματος τιμών καυσίμου. Αυτή η διαδικασία επιτυγχάνεται μέσω της χρήσης της βιβλιοθήκης ApexCharts, μιας εξαιρετικά δημοφιλούς βιβλιοθήκης διαγραμμάτων JavaScript.

Το διάγραμμα τιμών καυσίμου αποτελεί έναν εύκολο τρόπο να απεικονίζονται τα δεδομένα τιμών σε μια ευανάγνωστη, γραφική μορφή. Αυτό επιτρέπει στους χρήστες να κατανοήσουν γρήγορα τις τάσεις και τις μεταβολές στις τιμές των καυσίμων με μια ματιά, αντί να πρέπει να διαβάσουν πίνακες ή αριθμούς.

Τέλος, η δημιουργία του διαγράμματος γίνεται μέσω της κλήσης συγκεκριμένων μεθόδων της βιβλιοθήκης ApexCharts, οι οποίες δέχονται τα δεδομένα των τιμών των καυσίμων ως είσοδο και παράγουν το αντίστοιχο διάγραμμα. Αυτό το διάγραμμα στη συνέχεια εμφανίζεται στον χρήστη, παρέχοντας ένα οπτικό αντίκτυπο των δεδομένων.

Κεφάλαιο 7ο: Παρουσίαση του FuelPricesApp / Case studies

Η εφαρμογή FuelPricesApp διαθέτει έναν πλούσιο συνδυασμό διαδραστικών διεπαφών και λειτουργιών που έχουν ως στόχο να διευκολύνουν τους χρήστες να ανακτούν, να κατανοούν και να συγκρίνουν τις τιμές των καυσίμων. Μέσω της εφαρμογής, οι χρήστες μπορούν να εξάγουν πληροφορίες για τις τρέχουσες τιμές των καυσίμων, να παρακολουθούν την εξέλιξη των τιμών μέσω διαγραμμάτων, να λαμβάνουν προβλέψεις για μελλοντικές τιμές και να αξιολογούν την ακρίβεια των προβλέψεων.

Στις επόμενες ενότητες, θα παρουσιάσουμε δύο περιπτώσεις χρήσης (case studies) που θα αποδείξουν την πρακτική χρησιμότητα και την αποτελεσματικότητα της εφαρμογής στην ενημέρωση και την υποστήριξη των χρηστών της. Με αυτόν τον τρόπο, σκοπεύουμε να προσφέρουμε μια πλήρη και διαφωτιστική εικόνα της εφαρμογής FuelPricesApp, αποκαλύπτοντας τόσο την τεχνολογική της υποδομή όσο και την πρακτική της αξία.

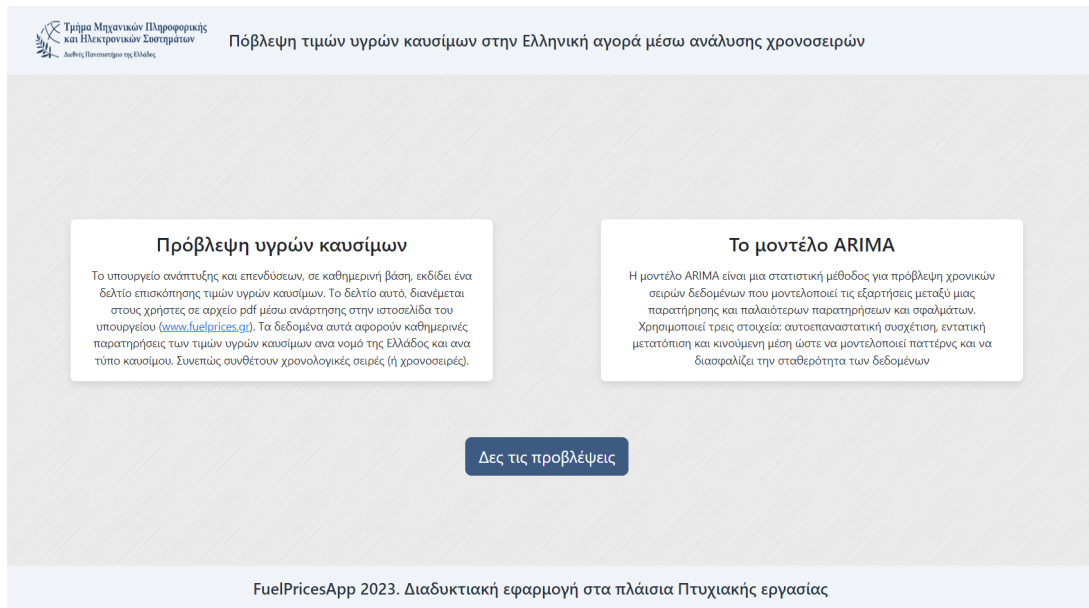
7.1 Αρχική σελίδα εφαρμογής

Η αρχική σελίδα της εφαρμογής FuelPricesApp είναι απλή και καλά οργανωμένη, με στόχο να παρέχει στο χρήστη όλες τις απαραίτητες πληροφορίες για τον σκοπό και τη λειτουργικότητα της εφαρμογής.

Η σελίδα ξεκινά με ένα μενού πλοήγησης (navbar), που περιέχει το λογότυπο του τμήματος της σχολής και την περιγραφή της εφαρμογής ("Πρόβλεψη τιμών υγρών καυσίμων στην Ελληνική αγορά μέσω ανάλυσης χρονοσειρών").

Ακολουθεί μια κεντρική ενότητα, η οποία περιέχει δύο κάρτες (cards) που παρέχουν βασικές πληροφορίες για την εφαρμογή. Η πρώτη κάρτα εξηγεί το πώς το υπουργείο ανάπτυξης και επενδύσεων παρακολουθεί τις τιμές των καυσίμων και πώς αυτά τα δεδομένα είναι διαθέσιμα στους χρήστες μέσω της εφαρμογής. Η δεύτερη κάρτα παρουσιάζει το μοντέλο ARIMA, που είναι η στατιστική μέθοδος που χρησιμοποιείται για την πρόβλεψη των τιμών των καυσίμων.

Στο κέντρο της σελίδας υπάρχει ένα κουμπί που προσκαλεί τους χρήστες να δουν τις προβλέψεις των τιμών των καυσίμων. Στο τέλος της σελίδας, υπάρχει μια σταθερή γραμμή πλοήγησης (navbar) που παρουσιάζει την εφαρμογή ως μέρος της πτυχιακής εργασίας.



Σχήμα 7.1: Αρχική σελίδα της εφαρμογής FuelPricesApp

7.2 Παρουσίαση τιμών υγρών καυσίμων

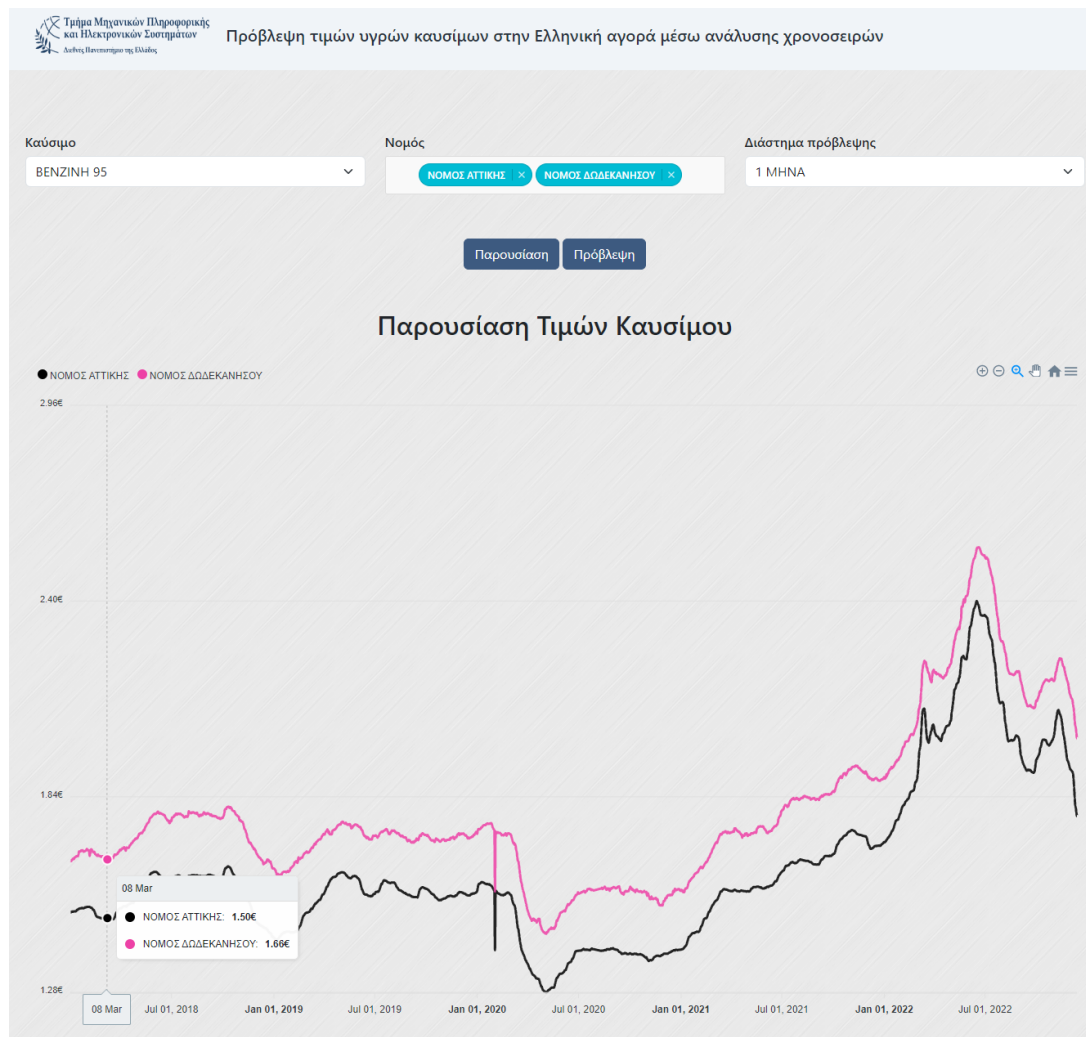
Η δεύτερη σελίδα της εφαρμογής FuelPricesApp οργανώνεται γύρω από την επιλογή καυσίμων και περιοχών και την παρουσίαση των τιμών των καυσίμων.

Όπως και στην αρχική σελίδα, η σελίδα ξεκινά με μια γραμμή πλοήγησης που περιέχει το λογότυπο του τμήματος της σχολής και την περιγραφή της εφαρμογής (“Πρόβλεψη τιμών υγρών καυσίμων στην Ελληνική αγορά μέσω ανάλυσης χρονοσειρών”).

Στη συνέχεια, ο χρήστης μπορεί να επιλέξει το είδος του καυσίμου, την περιοχή και το διάστημα πρόβλεψης μέσα από αναπτυσσόμενα μενού.

Υπάρχουν δύο κουμπιά, “Παρουσίαση” και “Πρόβλεψη”. Για το κομμάτι της διαγραμματικής παρουσίασης των τιμών των υγρών καυσίμων, όταν ο χρήστης πατήσει το αντίστοιχο κουμπί, η εφαρμογή παρουσιάζει τις τρέχουσες τιμές των καυσίμων για την επιλεγμένη περιοχή και το είδος του καυσίμου. Είναι αξιόλογο εδώ να σημειωθεί ότι, ο χρήστης έχει την δυνατότητα να επιλέξει την οπτικοποίηση των τιμών στο ίδιο γράφημα μέχρι και για πέντε νομούς της Ελλάδας.

Τέλος, η σελίδα περιέχει το διάγραμμα που παρουσιάζει τις τιμές των καυσίμων. Αυτό το διάγραμμα ενημερώνεται ανάλογα με τις επιλογές του χρήστη.



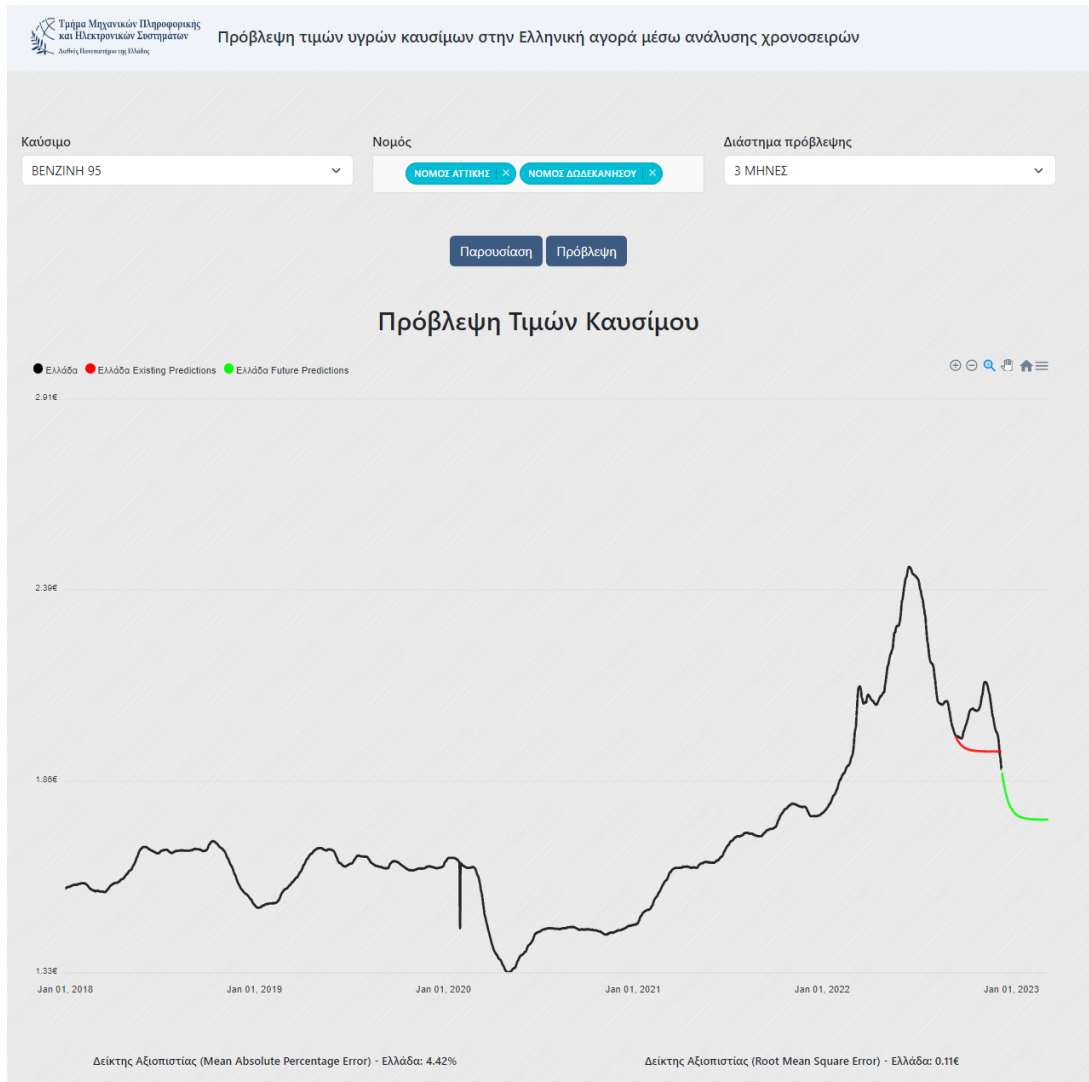
Σχήμα 7.2: Διεπαφή παρουσίασης τιμών υγρών καυσίμων της εφαρμογής FuelPricesApp

7.3 Πρόβλεψη μελλοντικών τιμών

Στην ίδια ιστοσελίδα, υπάρχει η λειτουργία "Πρόβλεψη", που ενεργοποιείται με την πίεση του αντίστοιχου κουμπιού. Κατόπιν της ενεργοποίησης αυτής, η εφαρμογή παρουσιάζει οπτικοποιημένες τις προβλέψεις που δημιουργήθηκαν από το μοντέλο ARIMA για τις τιμές του επιλεγμένου καυσίμου στην επικράτεια.

Το γράφημα παρουσιάζει τις ιστορικές τιμές των καυσίμων με μαύρο, τις προβλέψεις με πράσινο χρώμα για τις μελλοντικές ημερομηνίες, ανάλογα με τον αριθμό των ημερών που έχει επιλέξει ο χρήστης από την επιλογή "Διάστημα Πρόβλεψης". Παράλληλα, οι αντίστοιχες προβλέψεις για τις παρελθοντικές ημερομηνίες απεικονίζονται με κόκκινο χρώμα, επίσης βάσει του επιλεγμένου διαστήματος πρόβλεψης. Αυτό πραγματοποιείται για να επιτραπεί στον χρήστη την οπτική αξιολόγηση των αποτελεσμάτων της πρόβλεψης.

Επιπλέον, στο κάτω μέρος του διαγράμματος υπάρχουν δύο στοιχεία, που παρουσιάζουν μετρήσεις της ακρίβειας της πρόβλεψης, όπως το Mean Absolute Percentage Error (MAPE) και το Root Mean Square Error (RMSE). Αυτές οι μετρήσεις μπορούν να παρέχουν στον χρήστη μια εκτίμηση της αξιοπιστίας της πρόβλεψης.



Σχήμα 7.3: Διεπαφή προβλέψεων τιμών υγρών καυσίμων της εφαρμογής FuelPricesApp

Κεφάλαιο 8ο: Συμπεράσματα - Μελλοντική εργασία

8.1 Συμπεράσματα

Καθώς ολοκληρώνουμε αυτήν την πτυχιακή εργασία, επιβεβαιώνεται ότι οι εργασίες που αφορούν την εξόρυξη και την ανάλυση δεδομένων, ειδικά των χρονοσειρών, είναι πολύπλοκες και απαιτητικές. Η προσπάθεια να προβλέψουμε τις τιμές των υγρών καυσίμων αποτελεί ένα πολυδιάστατο πρόβλημα, το οποίο είναι επηρεασμένο από πολυάριθμες μεταβλητές.

Η προσέγγιση που ακολουθήσαμε περιλάμβανε την ανάπτυξη μιας διαδικτυακής εφαρμογής που θα μπορούσε να συλλέγει και να αναλύει δεδομένα των υγρών καυσίμων, καθώς και την χρήση εξελιγμένων τεχνικών ανάλυσης χρονοσειρών και μοντέλων πρόβλεψης. Τα συμπεράσματα που προέκυψαν από αυτήν την εργασία είναι σημαντικά, τόσο για την επιστημονική κοινότητα όσο και για την βιομηχανία των καυσίμων.

Η πρώτη δυσκολία που αντιμετωπίσαμε ήταν η ανάλυση των χρονοσειρών. Οι τιμές των υγρών καυσίμων παρουσιάζουν μεγάλη μεταβλητότητα, η οποία οφείλεται σε διάφορους παράγοντες, όπως οι διακυμάνσεις της τιμής του πετρελαίου, οι φορολογικές πολιτικές, οι εποχικές επιρροές (υγειονομικές καταστάσεις), αλλά και οι πολιτικές εντάσεις σε παγκόσμιο επίπεδο (πόλεμοι).

Η δεύτερη δυσκολία ήταν η προσπάθεια πρόβλεψης αυτών των τιμών. Παρότι χρησιμοποιήσαμε σύγχρονες μεθόδους πρόβλεψης χρονοσειρών όπως το μοντέλο ARIMA, οι προβλέψεις μας δεν ήταν πάντα ακριβείς, καθώς οι τιμές των υγρών καυσίμων είναι εξαιρετικά απρόβλεπτες. Οι προβλέψεις είναι συχνά επηρεασμένες από μη-προβλέψιμες αλλαγές στην αγορά καυσίμων, τις οποίες τα μοντέλα δεν μπορούν να λάβουν υπόψη.

Σε γενικές γραμμές, αυτή η πτυχιακή εργασία μας προσέφερε πολύτιμες γνώσεις και εμπειρίες στην ανάλυση χρονοσειρών και την πρόβλεψη τιμών. Παρότι αντιμετωπίσαμε πολλές προκλήσεις και δυσκολίες, είμαστε πεπεισμένοι ότι τα διδάγματα που αποκτήσαμε είναι άκρως ενημερωτικά και χρήσιμα για τις μελλοντικές μας επιστημονικές προσπάθειες.

8.2 Μελλοντική εργασία

Υπάρχουν αρκετές δυνατότητες για περαιτέρω επέκταση και εξέλιξη της παρούσας πτυχιακής εργασίας. Επιπλέον, αυτή η εργασία μπορεί να αποτελέσει πηγή έμπνευσης για τη δημιουργία νέων, καινοτόμων εργασιών που θα εμβαθύνουν περαιτέρω στην εν λόγω θεματική ενότητα. Μερικές από αυτές θα μπορούσαν να είναι οι εξής :

- *Βελτίωση της ακρίβειας πρόβλεψης*: Μπορεί να γίνει περαιτέρω έρευνα για την εύρεση καλύτερων μοντέλων πρόβλεψης για τις τιμές των υγρών καυσίμων. Αυτό μπορεί να περιλαμβάνει τη χρήση πιο προηγμένων τεχνικών μηχανικής μάθησης ή ακόμη και τη χρήση νευρωνικών δικτύων

- *Προσθήκη περισσότερων παραγόντων:* Η ανάλυση μπορεί να γίνει πιο σύνθετη με την προσθήκη περισσότερων παραγόντων που επηρεάζουν την τιμή των υγρών καυσίμων, όπως οι διεθνείς τιμές του πετρελαίου, η πολιτική κατάσταση, οι οικονομικοί δείκτες κ.λπ.
- *Επέκταση της εφαρμογής σε άλλες αγορές:* Η διαδικτυακή εφαρμογή μπορεί να προσαρμοστεί για να καλύψει και άλλες αγορές εκτός από τις τιμές των υγρών καυσίμων. Για παράδειγμα, θα μπορούσε να χρησιμοποιηθεί για την παρακολούθηση και την πρόβλεψη των τιμών των τροφίμων, των μεταλλευμάτων ή των ακινήτων
- *Εφαρμογή διαφορετικών τεχνικών ανάλυσης χρονοσειρών:* Θα μπορούσαμε να εφαρμόσουμε διαφορετικές τεχνικές ανάλυσης χρονοσειρών, όπως οι στοχαστικές μέθοδοι ή οι μη γραμμικές μέθοδοι, και να συγκρίνουμε τα αποτελέσματα με αυτά της τρέχουσας εργασίας
- *Έρευνα για την επίδραση της αστάθειας των τιμών των καυσίμων:* Θα ήταν ενδιαφέρον να γίνει μια έρευνα που θα εξετάζει πώς η αστάθεια των τιμών των καυσίμων επηρεάζει την οικονομία, τη βιομηχανία και την καθημερινή ζωή

BIBΛΙΟΓΡΑΦΙΑ

- [1] D. Yergin, *The New Map: Energy, Climate, and the Clash of Nations*. Penguin Press, 2020.
- [2] I. E. A. (IEA), “Oil market report,” 2021.
- [3] M. S. Rahman and E. Velayutham, “The impact of covid-19 pandemic on the global economy: Emphasis on the crude oil market,” *Energy Research Social Science*, vol. 68, p. 101693, 2020.
- [4] I. R. E. A. (IRENA), “Covid-19 and the energy transition: Global stimulus packages and the energy transition,” *Insights from IRENA’s Global Renewables Outlook*, 2020.
- [5] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*. Prentice Hall, 1994.
- [6] R. J. Hyndman and G. Athanasopoulos, “Forecasting: principles and practice,” *OTexts*, 2018.
- [7] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*. Springer, 3 ed., 2016.
- [8] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. John Wiley Sons, 5 ed., 2015.
- [9] S. Seabold and J. Perktold, “Statsmodels: Econometric and statistical modeling with python,” *Proceedings of the 9th Python in Science Conference*, vol. 57, p. 61, 2010.
- [10] J. D. Hamilton, *Time Series Analysis*. Princeton University Press, 1994.
- [11] J. W. Taylor, “Exponential smoothing for forecasting in r,” *International Journal of Forecasting*, vol. 30, no. 2, pp. 603–610, 2010.
- [12] G. E. Box and G. M. Jenkins, “Time series analysis: Forecasting and control,” *Holden-Day San Francisco*, 1970.
- [13] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*. Springer, 2 ed., 2002.
- [14] R. J. Hyndman and Y. Khandakar, “Forecasting time series using r,” *Monash University, Department of Econometrics and Business Statistics*, 2008.
- [15] G. P. Zhang, “Time series forecasting using a hybrid arima and neural network model,” *Neurocomputing*, vol. 50, pp. 159–175, 2003.
- [16] R. Larsen, *HTML5 and CSS3 for the Real World*. SitePoint, 2015.
- [17] E. A. Meyer, *CSS: The Definitive Guide*. O’Reilly Media, 4th ed., 2017.
- [18] K. Simpson, *You Don’t Know JS: Up & Going*. O’Reilly Media, 2015.
- [19] M. Haverbeke, *Eloquent JavaScript: A Modern Introduction to Programming*. No Starch Press, 2018.
- [20] D. Cochran and I. Whitley, *Bootstrap 4 Cookbook*. Packt Publishing, 2021.

- [21] J. Chhipa, “Apexcharts.js: Interactive svg charts.” <https://github.com/apexcharts/apexcharts.js>, 2021. Accessed: yyyy-mm-dd.
- [22] J. VanderPlas, *Python Data Science Handbook: Essential Tools for Working with Data*. O’Reilly Media, 2022.
- [23] A. C. Müller and S. Guido, *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O’Reilly Media, 2022.
- [24] S. Ramírez, “Fastapi: A modern, fast (high-performance) web framework for building apis with python 3.7+ based on standard python type hints,” 2018. Accessed: 2023-05-05.
- [25] P. DuBois, *MySQL Cookbook: Solutions for Database Developers and Administrators*. O’Reilly Media, 2008.
- [26] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science Business Media, 2009.