ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Πρόβλεψη εκπομπών διοξειδίου του άνθρακα με την χρήση αλγορίθμων Μηχανικής Μάθησης

ΑΛΕΞΑΝΔΡΑ ΚΑΛΛΙΓΕΡΑΚΗ

Α.Μ. 711171071

Επιβλέπουσα Καθηγήτρια: Παναγιώτα Τσελέντη, ΕΔΙΠ

ΑΘΗΝΑ, ΙΟΥΛΙΟΣ 2024

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Πρόβλεψη εκπομπών διοξειδίου του άνθρακα με την χρήση αλγορίθμων Μηχανικής Μάθησης

Prediction of Carbon Dioxide Emissions using Machine Learning Algorithms

Αλεξάνδρα Καλλιγεράκη

Α.Μ. 711171071

Επιβλέπουσα Καθηγήτρια: Παναγιώτα Τσελέντη, ΕΔΙΠ

Εγκρίθηκε από την κάτωθι τριμελή εξεταστική επιτροπή:

Α/Α	ΟΝΟΜΑΤΕΠΩΝΥΜΟ	ΒΑΘΜΙΔΑ/ΙΔΙΟΤΗΤΑ	ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ
1	ΤΣΕΛΕΝΤΗ ΠΑΝΑΓΙΩΤΑ	ΕΔΙΠ	Panagiota Tselenti Digitally signed by Panagiota Tselenti Date: 2024.07.26 21:22:59 +03'00'
2	ΤΡΟΥΣΣΑΣ ΧΡΗΣΤΟΣ	ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ	Christos Troussas Digitally signed by Christos Troussas Date: 2024.07.29 17:15:32 +03'00'
3	ΑΚΡΙΒΗ ΚΡΟΥΣΚΑ	ΕΔΙΠ	Akrivi Krouska Digitally signed by Akrivi Krouska Date: 2024.07.27 08:12:06 +03'00'

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ

Η κάτωθι υπογεγραμμένη Αλεξάνδρα Καλλιγεράκη του Εμμανουήλ, με αριθμό μητρώου 711171071 φοιτήτρια του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

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

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

Η Δηλούσα

Α Καλλιγεράκη

Αλεξάνδρα Καλλιγεράκη

ΑΦΙΕΡΩΣΗ

Η παρούσα διπλωματική εργασία είναι αφιερωμένη στην αγαπημένη μου γιαγιά,

Αιμιλία.

ΕΥΧΑΡΙΣΤΙΕΣ

Καθώς το ταξίδι αυτό τελειώνει, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες σε όλους όσους με υποστήριξαν κατά την διάρκεια της συγγραφής αυτής της διπλωματικής εργασίας και γενικότερα των σπουδών μου.

Πρώτα και κύρια, ευχαριστώ την επιβλέπουσα καθηγήτρια μου, κα. Παναγιώτα Τσελέντη, για την καθοδήγηση και τις πολύτιμες συμβουλές της. Η υπομονή και η ενθάρρυνσή της υπήρξαν καθοριστικοί παράγοντες για την ολοκλήρωση αυτής της έρευνας.

Θα ήθελα να εκφράσω την πιο βαθιά μου ευγνωμοσύνη στην οικογένειά μου, η οποία υπήρξε το σταθερό στήριγμά μου και συνέβαλε καθοριστικά στην επιτυχία μου. Η αδιάκοπη αγάπη τους, η στήριξη, η πίστη τους σε εμένα, καθώς και οι θυσίες που κατέβαλαν, η παρακίνηση και η κατανόηση τους, ιδιαίτερα στις δύσκολες στιγμές, μου έδωσαν την δύναμη να συνεχίσω και να ολοκληρώσω αυτή την απαιτητική προσπάθεια.

Τέλος, ευχαριστώ όλους τους φίλους και τους συναδέλφους μου που ήταν πάντα εκεί για να με ενθαρρύνουν και να μοιραστούν μαζί μου τις δυσκολίες και τις επιτυχίες αυτής της διαδρομής. Η υποστήριξή τους ήταν ανεκτίμητη και πολύτιμη.

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία εξετάζει τη χρήση αλγορίθμων μηχανικής μάθησης για την πρόβλεψη εκπομπών διοξειδίου του άνθρακα (CO_2), στοχεύοντας στην ανάπτυξη και αξιολόγηση διαφόρων μοντέλων για την ακριβή πρόβλεψη των εκπομπών και την αναγνώριση των σημαντικότερων χαρακτηριστικών που τις επηρεάζουν. Αρχικά, πραγματοποιήθηκε εκτενής βιβλιογραφική ανασκόπηση για να αναλυθούν προηγούμενες έρευνες και τεχνικές, ενώ στη συνέχεια παρουσιάστηκε το θεωρητικό υπόβαθρο των αλγορίθμων μηχανικής μάθησης, όπως Gradient Boosting, XGBoost, Random Forest, Decision Tree, Ridge Regression και Lasso Regression. Το πειραματικό μέρος περιλάμβανε τη συλλογή και προεπεξεργασία δεδομένων, την επιλογή χαρακτηριστικών, την εκπαίδευση και αξιολόγηση των μοντέλων, καθώς και τη σύγκριση της απόδοσής τους. Τα αποτελέσματα έδειξαν ότι τα μοντέλα Gradient Boosting και XGBoost παρουσίασαν την καλύτερη απόδοση, με το XGBoost να επιτυγχάνει το χαμηλότερο μέσο τετραγωνικό σφάλμα (MSE) και τον υψηλότερο συντελεστή προσδιορισμού (R^2), καθιστώντας το πιο αποδοτικό μοντέλο. Αναγνωρίστηκαν ορισμένοι περιορισμοί, όπως η περιορισμένη διαθεσιμότητα δεδομένων και η υπερπροσαρμογή των μοντέλων, αλλά επίσης αναδείχθηκε η σημασία της προσεκτικής προεπεξεργασίας δεδομένων και της επιλογής χαρακτηριστικών. Προτάθηκαν κατευθύνσεις για μελλοντική έρευνα, όπως η εφαρμογή πιο προηγμένων αλγορίθμων και η χρήση μεγαλύτερων και πιο ποικιλόμορφων συνόλων δεδομένων για τη βελτίωση της ακρίβειας και της γενίκευσης των προβλέψεων. Τα ευρήματα της έρευνας έχουν σημαντικές εφαρμογές στην ανάπτυξη πολιτικών και στρατηγικών για τη μείωση των εκπομπών CO_2, προσφέροντας πολύτιμα εργαλεία για την αντιμετώπιση της κλιματικής αλλαγής.

Λέξεις Κλειδιά: Μηχανική Μάθηση, Εκπομπές Διοξειδίου του Άνθρακα, Μοντέλο Πρόβλεψης, Κλιματική Αλλαγή

ABSTRACT

This thesis examines the use of machine learning algorithms for predicting carbon dioxide (CO_2) emissions, aiming to develop and evaluate different models for accurate prediction of emissions and identification of the most important characteristics that influence them. Initially, an extensive literature review was conducted to analyse previous research and techniques, followed by a theoretical background of machine learning algorithms such as Simple Linear Regression, Gradient Boosting, XGBoost, Random Forest, Decision Tree, Ridge Regression and Lasso Regression. The experimental part included data collection and pre-processing, feature selection, training and evaluation of the models, and comparison of their performance. The results showed that the Gradient Boosting and XGBoost models performed best, with XGBoost achieving the lowest mean square error (MSE) and the highest coefficient of determination (R^2), making it the most efficient model. Some limitations were acknowledged, such as limited data availability and model overfitting, but the importance of careful data preprocessing and feature selection was also highlighted. Directions for future research were suggested, such as the application of more advanced algorithms and the use of larger and more diverse datasets to improve the accuracy and generalizability of the predictions. The findings of the research have important applications in the development of policies and strategies to reduce CO_2 emissions, providing valuable tools to address climate change.

Keywords: Machine Learning, Carbon Dioxide Emissions, Prediction Model, Climate Change

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΚΑΤΑΛΟΓΟΣ ΑΡΤΙΚΟΛΕΞΩΝ - ΣΥΝΤΟΜΟΓΡΑΦΙΩΝ

ANN Artificial Neural Networks
ARIMA Autoregressive Integrated Moving Average
CatBoost Categorical Boosting
CH_4 Methane
CO_2 Carbon Dioxide
COVID-19 Coronavirus disease
EEA European Environment Agency
GB Gradient Boosting
GBDT Gradient Boosted Decision Trees
GDP Gross Domestic Product
IPCC Intergovernmental Panel on Climate Change
IUCN International Union for Conservation of Nature
kg kilogram
kt kiloton
kWh kilowatt-hour
LDV Light Duty Vehicles
LightGBM Light Gradient-Boosting Machine
LR Lasso Regression
LSTM Long Short-Term Memory
MAPE Mean absolute percentage error
MSE Mean Square Error
N_2O Nitrous oxide
NaN Not a Number
NASA National Aeronautics and Space Administration

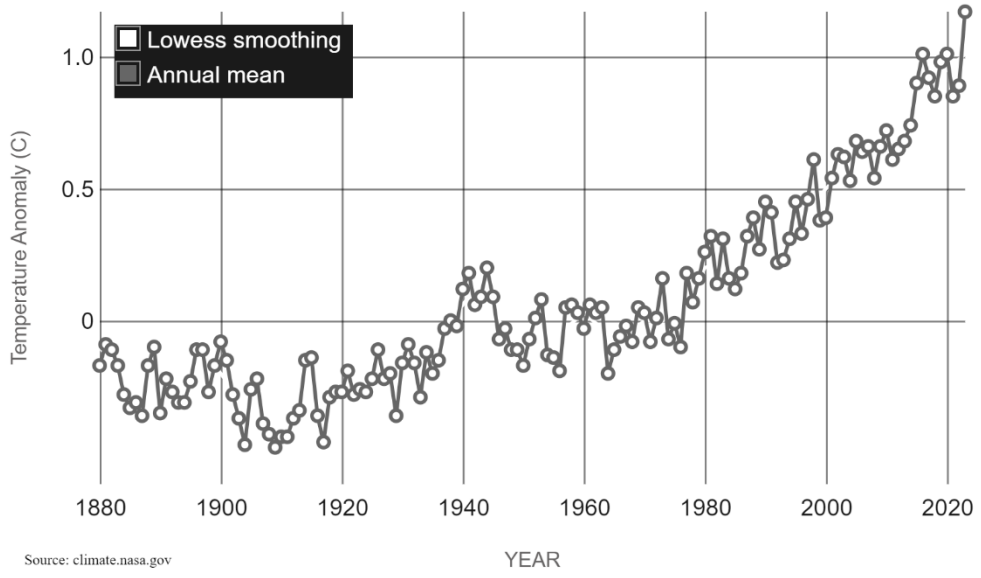
OLS Ordinary least squares
ppm parts per million
RF Random Forest
RFE Recursive Feature Elimination
RMSE Root Mean Square Error
RR Ridge Regression
SARIMAX Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors
SLR Simple Linear Regression
sq. km square metres
SVR Support Vector Regression
VS Visual Studio
XGBoost eXtreme Gradient Boosting
ΑΕΠ Ακαθάριστο Εγχώριο Προϊόν
ΗΠΑ Ηνωμένες Πολιτείες της Αμερικής
MAE Mean Absolute Error
τ.χλμ. τετραγωνικό χιλιόμετρο

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

1.1 Περιγραφή Προβλήματος

Οι αλλαγές που παρατηρούνται στο κλίμα της γης είναι γνωστές εδώ και 800.000 χρόνια [1]. Παρά τις τεράστιες μεταβολές που έχουν παρατηρηθεί κατά την διάρκεια όλων αυτών των ετών, όπως οι οκτώ κύκλοι παγετώνων που έχουν σημειωθεί και οι θερμότερες περίοδοι που ακολουθούσαν, το φαινόμενο της υπερθέρμανσης του πλανήτη που βιώνουμε στις μέρες μας είναι πρωτοφανές, τόσο ως προς την ταχύτητα εξάπλωσης του, όσο και ως προς την έκταση που λαμβάνει.

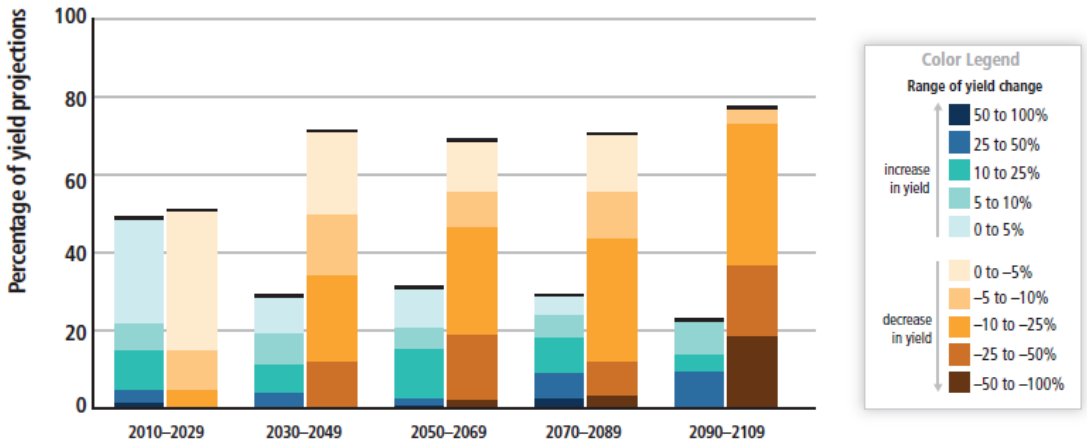
Τις τελευταίες δεκαετίες, διάφορες μελέτες και εκθέσεις έχουν επισημάνει την αύξηση των θερμοκρασιών, τα ακραία καιρικά φαινόμενα, την άνοδο της στάθμης της θάλασσας και τις αλλαγές στα οικοσυστήματα [2]. Σύμφωνα με την έκθεση της IPCC, οι παγκόσμιες θερμοκρασίες έχουν μεταβληθεί με αύξοντα ρυθμό κατά περίπου 1,2 βαθμούς Κελσίου από την προ-βιομηχανική εποχή. Άμεση συνέπεια της αυξημένης θερμοκρασίας, αποτελεί η τήξη των παγετώνων και η άνοδος της στάθμης της θάλασσας, η οποία απειλεί παράκτιες περιοχές και νησιά. Τα ακραία καιρικά φαινόμενα όπως οι καύσωνες, οι πλημμύρες και οι ξηρασίες έχουν γίνει πιο συχνά και έντονα, προκαλώντας σημαντικές ζημιές σε πολλές περιοχές του πλανήτη [3]. Χαρακτηριστικό παράδειγμα αποτελούν οι ανεξέλεγκτες πυρκαγιές που πλήττουν ετησίως το τροπικό δάσος του Αμαζονίου, όπου σύμφωνα με τη δορυφορική παρακολούθηση σε πραγματικό χρόνο [4] έχουν ξεσπάσει παραπάνω από 10.000 πυρκαγιές σε 11.000 τ.χλμ. για το έτος του 2024.



Εικόνα 1: Μέση παγκόσμια θερμοκρασία σε βάθος χρόνου από το 1880 *[5]*

Παράλληλα, οι μεταβολές στα οικοσυστήματα έχουν επηρεάσει την βιοποικιλότητα, με αποτέλεσμα πολλά είδη να χαρακτηρίζονται πλέον ως απειλούμενα με εξαφάνιση. Σύμφωνα με την έρευνα της ΕΕΑ, οι παραπάνω κρίσιμες αλλαγές έχουν προκαλέσει σημαντικά προβλήματα στην δομή και λειτουργία των οικοσυστημάτων, επηρεάζοντας την διαθεσιμότητα των πόρων και την βιωσιμότητα των ειδών [6]. Η ταχεία εξάπλωση των μεταβολών έχει δημιουργήσει σοβαρές επιπτώσεις στην γενετική σύνθεση, την συμπεριφορά και την επιβίωση των ειδών, μιας και περιορίζεται η ικανότητα τους να προσαρμοστούν στα νέα δεδομένα του περιβάλλοντος. Σήμερα, υπολογίζεται πως τουλάχιστον 10.967 είδη που αναφέρονται στον Κόκκινο Κατάλογο Απειλούμενων Ειδών της IUCN έχουν επηρεαστεί από την κλιματική αλλαγή [7], με χαρακτηριστικό παράδειγμα καταγραφής της εξαφάνισης του πρώτου θηλαστικού· του τρωκτικού Bramble Cay melomys (Melomys rubicola) [8].

Η κλιματική αλλαγή, ωστόσο, δεν είναι μόνο ζήτημα του περιβάλλοντος, αλλά και μια σοβαρή απειλή για την ανθρώπινη υγεία και ευημερία. Οι συνεχείς και αυξανόμενες αλλαγές στον καιρό επηρεάζουν τη γεωργία, την παροχή του νερού και την ποιότητα του αέρα, προξενώντας μείζονα θέματα στον τομέα της δημόσιας υγείας και της οικονομίας [9].



Εικόνα 2: Προβλεπόμενη μεταβολή των παγκόσμιων συνολικών αποδόσεων των καλλιεργειών λόγω της κλιματικής αλλαγής *[9]*

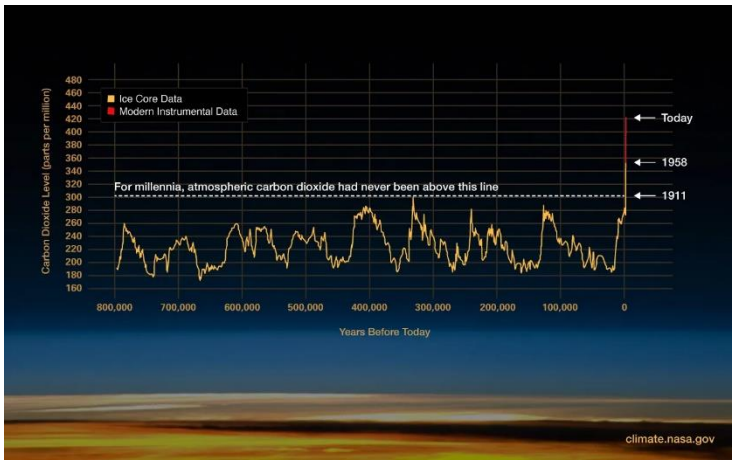
Ένας από τους σημαντικότερους παράγοντες που συμβάλλει στην κλιματική αλλαγή είναι η ραγδαία αύξηση των εκπομπών του διοξειδίου του άνθρακα μετά την περίοδο της Βιομηχανικής Επανάστασης.

Για την κατανόηση της παραπάνω πρότασης είναι σημαντικό να αναλυθεί η έννοια «φαινόμενο του θερμοκηπίου». Ως φαινόμενο του θερμοκηπίου ορίζουμε μια φυσική διαδικασία που είναι απαραίτητη για την διατήρηση της ζωής στη Γη. Αναλυτικότερα, οι ακτίνες του ήλιου διαπερνούν την ατμόσφαιρα και θερμαίνουν την επιφάνεια της Γης, η οποία στην συνέχεια εκπέμπει την θερμότητα αυτή στην ατμόσφαιρα με την μορφή υπέρυθρης ακτινοβολίας. Δυστυχώς όμως, ένας μέρος αυτής της ακτινοβολίας εγκλωβίζεται

από τα λεγόμενα αέρια του θερμοκηπίου. Κάποια από αυτά τα αέρια είναι το διοξείδιο του άνθρακα (CO_2), το μεθάνιο (CH_4) και το υποξείδιο του αζώτου (N_2O). Η ακτινοβολία που εγκλωβίζεται προκαλεί την θέρμανση της κατώτερης ατμόσφαιρας και της επιφάνειας της Γης. Παρότι η διαδικασία αυτή είναι ζωτικής σημασίας για την διατήρηση μιας θερμοκρασίας που δίνει την δυνατότητα ύπαρξης ζωής, η υπερβολική συγκέντρωση των προαναφερθέντων αερίων εντείνουν το φαινόμενο και προκαλούν υπερθέρμανση του πλανήτη [10].

Το διοξείδιο του άνθρακα (CO_2), όπως προαναφέρθηκε, είναι το κύριο ανθρωπογενές αέριο θερμοκηπίου και παίζει καθοριστικό ρόλο στην κλιματική αλλαγή και συγκεκριμένα στην υπερθέρμανση του πλανήτη. Η παραγωγή του προέρχεται κατά βάση από την καύση ορυκτών καυσίμων, όπως το φυσικό αέριο και το πετρέλαιο, από την αποψίλωση των δασών και από διάφορες ανθρώπινες βιομηχανικές διεργασίες [11].

Ποικίλες επιστημονικές μελέτες έχουν αποδείξει πως η αύξηση των εκπομπών του CO_2 είναι ανάλογη της αύξησης της θερμοκρασίας της Γης. Παραδείγματος χάρη, οι αναλύσεις των παγετώνων έχουν αποκαλύψει πως οι αυξήσεις της θερμοκρασίας κατά τα παλαιότερα χρόνια συνδέονται άμεσα με τις υψηλές ποσότητες CO_2 [12]. Επιπρόσθετα, τα κλιματικά μοντέλα αποδεικνύουν πως αν εξαιρέσουμε την αύξηση του CO_2 και των υπολοίπων αερίων του θερμοκηπίου που παράγονται από τους ανθρώπους δεν είναι δυνατόν να επεξηγηθεί επαρκώς η παρατηρούμενη αύξηση της παγκόσμιας υπερθέρμανσης [11].



Εικόνα 3: Αύξηση των επιπέδων CO2 στην ατμόσφαιρα από την Βιομηχανική Επανάσταση, με βάση δείγματα από πυρήνες πάγου και σύγχρονες μετρήσεις *[13]*

Σύμφωνα με την NASA η συγκέντρωση του CO_2 έχει αυξηθεί από 280 ppm σε 420 ppm σήμερα, όπως βλέπουμε και στην Εικόνα 3. Ο ρυθμός αύξησης των τιμών του CO_2 είναι ιδιαίτερα ανησυχητικός και το γεγονός αυτό υπογραμμίζει την ανάγκη για επείγουσες και ουσιώδεις δράσεις για την μείωση των εκπομπών του CO_2 και γενικότερα για την αντιμετώπιση της κλιματικής αλλαγής [12].

1.2 Στόχοι Έρευνας

Βασικοί στόχοι της έρευνας αποτελούν η ανάπτυξη και η αξιολόγηση μοντέλων μηχανικής μάθησης για την πρόβλεψη εκπομπών διοξειδίου του άνθρακα (CO_2), η αναγνώριση των πιο σημαντικών χαρακτηριστικών που επηρεάζουν αυτές τις εκπομπές και η σύγκριση της απόδοσης διαφορετικών αλγορίθμων. Τα μοντέλα αυτά θα εκπαιδευτούν με βάση ιστορικά δεδομένα εκπομπών CO_2 και άλλων σχετικών παραμέτρων που επηρεάζουν την παραγωγή του, όπως η κατανάλωση ηλεκτρικής ενέργειας, ο αστικός πληθυσμός και η κατανάλωση ανανεώσιμης ενέργειας. Τελικός σκοπός αυτής της έρευνας είναι η ανάπτυξη ενός αξιόπιστου μοντέλου που θα μπορεί να προβλέψει με ακρίβεια τις μελλοντικές εκπομπές CO_2, συμβάλλοντας με αυτόν τον τρόπο στην καλύτερη διαχείριση των εκπομπών και στην ανάπτυξη αποτελεσματικών στρατηγικών για τη μείωση τους.

1.3 Σημασία της Μελέτης

Η παρούσα έρευνα αποτελεί σημαντικό εργαλείο για διάφορους τομείς. Το μοντέλο πρόβλεψης που θα αναπτυχθεί θα μπορεί χρησιμοποιηθεί στον κλάδο της περιβαλλοντικής πολιτικής, καθώς θα παρέχει σημαντικά δεδομένα τα οποία θα είναι χρήσιμα για την λήψη αποφάσεων και την διαμόρφωση πολιτικών νομοθεσιών σχετικών με την αντιμετώπιση της κλιματικής αλλαγής.

Παράλληλα, τα αποτελέσματα της μελέτης θα συνεισφέρουν σημαντικά στην προώθηση της γνώσης των περιβαλλοντικών επιστημών. Οι αναλύσεις που θα προκύψουν θα διευκολύνουν στην κατανόηση των παραμέτρων που επηρεάζουν τις εκπομπές CO_2, μιας και το σύνολο δεδομένων που θα χρησιμοποιηθεί αντλεί πληροφορίες από τομείς όπως της ενέργειας, του περιβάλλοντος, της οικονομίας, της δημογραφίας, της κατανάλωσης καυσίμων και της γεωγραφίας.

Αξίζει, επιπλέον, να αναφερθεί πως η συγκεκριμένη εργασία θα συμβάλλει στον τομέα της μηχανικής μάθησης. Η διερεύνηση και η αξιολόγηση διαφόρων αλγορίθμων και τεχνικών θα παράσχει καινούργια γνώση ως προς την αποτελεσματικότητα και την ακρίβεια αυτών των μοντέλων. Τα δεδομένα που θα προκύψουν μπορούν να συντελέσουν βάση για μελλοντικές έρευνες και υλοποιήσεις, συμβάλλοντας στην ανέλιξη των μεθόδων πρόβλεψης και στην ενσωμάτωση της τεχνητής νοημοσύνης σε περιβαλλοντικές εφαρμογές.

1.4 Δομή της Διπλωματικής Εργασίας

Η παρούσα εργασία θα αναλυθεί στις εξής ενότητες:

1. Εισαγωγή: Αρχικά θα παρουσιάσουμε το πρόβλημα, τους στόχους και την σημασία της έρευνας, καθώς και την δομή της διπλωματικής εργασίας.

2. Βιβλιογραφική Ανασκόπηση: Σε αυτό το σημείο θα κάνουμε ανασκόπηση της υπάρχουσας βιβλιογραφίας σχετικά με τις εκπομπές CO_2. Αναλυτικότερα, θα εστιάσουμε σε προηγούμενες έρευνες, στις τεχνικές και μεθοδολογίες που χρησιμοποιήθηκαν και στις πρακτικές εφαρμογές των αποτελεσμάτων τους.

3. Θεωρητικό Υπόβαθρο: Στην συγκεκριμένη ενότητα θα γίνει ανάλυση των αλγορίθμων μηχανικής μάθησης που χρησιμοποιούνται και των θεωρητικών βάσεων τους.

4. Πειραματικό Μέρος: Το Κεφάλαιο 4 αφορά το πειραματικό μέρος της εργασίας και περιγράφει τη διαδικασία από την επιλογή των εργαλείων και τη συλλογή του συνόλου δεδομένων, μέχρι την προεπεξεργασία και την επιλογή χαρακτηριστικών. Αναλύονται οι μέθοδοι εκπαίδευσης και αξιολόγησης διαφόρων μοντέλων πρόβλεψης, καθώς και η σύγκριση των αποτελεσμάτων τους. Τέλος, το κεφάλαιο περιλαμβάνει την αξιολόγηση των αποτελεσμάτων και την πρόβλεψη των εκπομπών CO_2.

5. Συμπεράσματα και Μελλοντικές προκλήσεις: Στο Κεφάλαιο 6, θα συνοψιστούν τα κύρια συμπεράσματα της έρευνας, υπογραμμίζοντας τα σημαντικά ευρήματα και τη συμβολή της στη μελέτη των εκπομπών CO2. Θα συζητηθούν οι προοπτικές και οι μελλοντικές προκλήσεις που μπορεί να αντιμετωπιστούν, καθώς και προτάσεις για μελλοντικές ερευνητικές κατευθύνσεις και εφαρμογές της μεθοδολογίας σε άλλους τομείς ή προβλήματα.

ΚΕΦΑΛΑΙΟ 2: Βιβλιογραφική Ανασκόπηση

Η πρόβλεψη των εκπομπών του διοξειδίου του άνθρακα (CO_2) αποτελεί ένα σημαντικό κλάδο μελέτης που έχει προσελκύσει το ενδιαφέρον αρκετών ερευνητών. Στο κεφάλαιο αυτό, θα αναλύσουμε σημαντικές έρευνες που έχουν πραγματοποιηθεί, εστιάζοντας στις τεχνικές που ακολούθησαν και στα αποτελέσματά τους.

Το ερευνητικό άρθρο [14] αποτελεί παράδειγμα διερεύνησης της εφαρμογής διαφορών μοντέλων μηχανικής μάθησης για την πρόβλεψη εκπομπών CO_2. Στην συγκεκριμένη έρευνα, εξετάζονται τέσσερα μοντέλα πρόβλεψης τα οποία βασίζονται στην τεχνική SARIMAX, τα οποία εκπαιδεύονται σε δεδομένα εκπομπών CO_2 που συλλέχθηκαν κατά την διάρκεια διαφορετικών περιόδων της πανδημίας COVID-19. Οι τέσσερις περίοδοι που εξετάστηκαν είναι: πριν την πανδημία (pre), κατά την πανδημία (start), κατά την μετάδοση (trans) και μετά την πανδημία (post). Η έρευνα συγκρίνει την αποτελεσματικότητα αυτών των μοντέλων με στόχο της πρόβλεψη παγκοσμίων εκπομπών CO_2 για τις χρονικές περιόδους 2022 έως 2027, 2022 έως 2054 και 2022 έως 2072. Υπογραμμίζεται το γεγονός ότι οι μέθοδοι μηχανικής μάθησης που εκπαιδεύονται σε δεδομένα από την περίοδο μετά την πανδημία παρουσιάζουν υψηλότερη ακρίβεια, με το μέσο απόλυτο ποσοστό σφάλματος (MAPE) να είναι μόλις 0.09%. Επιπλέον, η μελέτη καταλήγει ότι οι καταστάσεις lockdown μπορούν να μειώσουν τις εκπομπές CO_2 και προτείνει της εφαρμογή τεχνητών lockdown ή μικρότερων εργασιακών ωραρίων για την βελτίωση του περιβάλλοντος.

Πρόσφατη έρευνα [15] επικεντρώνεται στην πρόβλεψη ανθρωπογενών εκπομπών CO_2. Για την επίτευξη του ερευνητικού στόχου, χρησιμοποιείται μια προσέγγιση μηχανικής μάθησης η οποία λαμβάνει υπόψη τα επίπεδα του διοξειδίου του άνθρακα στην ατμόσφαιρα. Κύριος στόχος της μελέτης είναι η διαχείριση της γεωγραφικής ποικιλομορφίας των εκπομπών του CO_2, διαιρώντας τα δεδομένα εκπομπών και χρησιμοποιώντας διάφορους αλγορίθμους μηχανικής μάθησης, όπως τα decision trees, GBDT, LightGBM, XGBoost και CatBoost. Μέσω της υιοθέτησης αυτών των προηγμένων μεθόδων, η μελέτη στοχεύει στην βελτίωση της ακρίβειας των προβλέψεων των εκπομπών CO_2. Ο αλγόριθμος LightGBM παρουσίασε ακρίβεια πρόβλεψης που έφτανε το 95%, ενώ οι υπόλοιποι αλγόριθμοι είχαν ελαφρώς χαμηλότερες αποδόσεις. Οι GBDT και XGBoost πλησίαζαν την απόδοση του LightGBM, αλλά δεν κατάφεραν να τον ξεπεράσουν. Οι decision tree και CatBoost είχαν σημαντικά χαμηλότερες επιδόσεις σε σχέση με τους άλλους αλγόριθμους. Τα αποτελέσματα καθιστούν σαφές ότι αυτή η προσέγγιση μπορεί να αναβαθμίσει σημαντικά την ικανότητα πρόβλεψης σε σύγκριση με τις παραδοσιακές μεθόδους.

Εν συνεχεία, η μελέτη [16] έχει ως στόχο να διερευνήσει τους παράγοντες που επηρεάζουν τις εκπομπές CO_2 στην Κίνα, χρησιμοποιώντας διάφορους αλγορίθμους μηχανικής μάθησης για την πρόβλεψη εκπομπών CO_2. Στη μελέτη χρησιμοποιούνται δεδομένα που έχουν συλλεχθεί από την χρονική περίοδο 2000 έως 2018, τα οποία περιλαμβάνουν οικονομικούς δείκτες, την δομή της βιομηχανίας, την αστικοποίηση, την επένδυση στην

έρευνα και ανάπτυξη, την χρήση ξένου κεφαλαίου και της κατανάλωσης ενέργειας. Για την πρόβλεψη των εκπομπών CO_2 εκπαιδεύονται ποικίλοι αλγόριθμοι, όπως ο Linear Regression, ο Decision Trees, ο Random Forests, ο Support Vector Machines και ο K-Nearest Neighbors. Από τους παραπάνω αλγορίθμους, ο K-Nearest Neighbors αποδείχθηκε ο πιο αποδοτικός καθώς πέτυχε την χαμηλότερη τιμή MAE, η οποία αποδεικνύει πως οι προβλέψεις του ήταν πιο κοντά στις πραγματικές τιμές εκπομπών CO_2. Είχε, επίσης, τον χαμηλότερο MSE, ο οποίος δείχνει ότι είχε τις λιγότερες μεγάλες αποκλίσεις στις προβλέψεις του. Τέλος, εμφάνισε τον χαμηλότερο RMSE, επιβεβαιώνοντας την υψηλή ακρίβεια των προβλέψεών του. Συνολικά, ο αλγόριθμος K-Nearest Neighbors ξεπερνούσε τους υπόλοιπους σε όλες τις μετρικές ακριβείας.

Σε ένα επιστημονικό άρθρο [17] οι συγγραφείς επικεντρώνονται στην πρόβλεψη CO_2 από Light Duty οχήματα (LDV). Η μελέτη χρησιμοποιεί δεδομένα από τον Καναδά για 7.384 LDV από το 2017 έως το 2021, συμπεριλαμβάνοντας μετρήσεις εκπομπών CO_2 και κατανάλωσης καυσίμων. Αλγόριθμοι μηχανικοί μάθησης, όπως ο CatBoost, ο SVR και ο Ridge Regression, χρησιμοποιήθηκαν για την εκπαίδευση των προβλεπτικών μοντέλων. Τα αποτελέσματα που προέκυψαν έδειξαν πως η μέθοδος μηχανικής μάθησης CatBoost είχε την καλύτερη απόδοση, επιτυγχάνοντας την υψηλότερη ακρίβεια με μικρότερο σφάλμα σε σύγκριση με τους άλλους αλγορίθμους. Χαρακτηριστικά, ο CatBoost παρουσίασε MSE 3.83, R-Squared 99.6, RMSE 1.9 και MAE 2.41. Στο τέλος της ανάλυσης, οι μελετητές προτείνουν ότι τα δεδομένα που προέκυψαν από το μοντέλο πρόβλεψης μπορούν να χρησιμοποιηθούν για την ανάπτυξη μεθόδων σχεδιασμού οχημάτων φιλικών προς το περιβάλλον και για την βελτίωση της αποδοτικότητας των καυσίμων.

Μια ακόμη έρευνα [18] στοχεύει στην πρόβλεψη εκπομπών CO_2 στην Ινδία για την επόμενη δεκαετία χρησιμοποιώντας διάφορα μοντέλα χρονολογικών σειρών. Αναλυτικότερα, στο επιστημονικό άρθρο χρησιμοποιούνται τρία στατιστικά μοντέλα (ARIMA, SARIMAX και Holt-Winters), δύο μοντέλα μηχανικής μάθησης (Linear Regression και Random Forest) και ένα μοντέλο βαθιάς μάθησης (LSTM). Το σύνολο των δεδομένων αποτελείται από μονοδιάστατες χρονολογικές σειρές εκπομπών CO_2 από το 1980 έως το 2019. Ως απόρροια προέκυψε ότι τα μοντέλα LSTM, SARIMAX και Holt-Winters είναι τα πιο ακριβή για την πρόβλεψη εκπομπών CO_2, με το μοντέλο LSTM να καταδεικνύεται ως το καλύτερο με 3,101% MAE και 60,635 RMSE. Το μοντέλο LSTM, το οποίο βασίζεται σε νευρωνικά δίκτυα με μνήμη (recurrent neural networks), τονίζεται πως ξεπέρασε τα άλλα μοντέλα στην πρόβλεψη χρονολογικών σειρών λόγω της δυνατότητας του να χειρίζεται την εκθετική ομαλότητα και να διατηρεί μακροχρόνιες εξαρτήσεις στα δεδομένα.

Συνεχίζοντας, μια επιπρόσθετη μελέτη [19] αποσκοπεί και αυτή στην ακριβή πρόβλεψη εκπομπών CO_2 στην Κίνα. Παρουσιάζει ένα προβλεπτικό μοντέλο δύο σταδίων με την χρήση αλγορίθμων μηχανικής μάθησης, όπως ο SVR, ο Random Forest, ο Ridge Regression και ο ANN. Το σύνολο των δεδομένων περιλαμβάνει εννέα ανεξάρτητες μεταβλητές που εκτείνονται από το 1985 έως το 2020. Οι μεταβλητές αυτές περιλαμβάνουν

τα εξής δεδομένα: άμεσες ξένες επενδύσεις, προστιθέμενη αξία στην βιομηχανία, εμπόριο, αστικό πληθυσμό, ΑΕΠ ανά κάτοικο, συνολική κατανάλωση ενέργειας, κατανάλωση άνθρακα, πετρελαίου και φυσικού αερίου. Η διαδικασία δύο σταδίων, η οποία προβλέπει πρώτα τις ανεξάρτητες μεταβλητές και στην συνέχεια τις εκπομπές CO_2, παρουσιάζει σημαντική βελτίωση της ακριβείας σε σχέση με τα μοντέλα ενός σταδίου. Παρατηρείται, επίσης, πως ο συνδυασμός πρόβλεψης των αλγορίθμων SVR-ANN αποδίδει τα χαμηλότερα σφάλματα πρόβλεψης, ενώ το μοντέλο που συνδυάζει τους SVR-RF έχει τα υψηλότερα. Για να γίνουμε πιο συγκεκριμένοι, η μέση μείωση σφαλμάτων πρόβλεψης για τα μοντέλα πρόβλεψης δύο σταδίων σε σχέση με τα μονό-σταδιακά είναι 36,06% για το RMSE και το MAE για τον συνδυασμό SVR-SVR, ενώ για τον SVR-RF, η μείωση είναι μόνο 5,98% για το RMSE και 6,91% για το MAE. Για την συνδυαστική μέθοδο SVR-Ridge, η μείωση είναι 43,05% για το RMSE και για το MAE, ενώ για το SVR-ANN, η μείωση είναι 14,81% για το RMSE και 15,35% για το MAE. Τέλος, επισημαίνεται από τους συγγραφείς ότι, παρότι ο συνδυασμός SVR-Ridge παρουσιάζει μεγαλύτερη βελτίωση απόδοσης σε σχέση με τον Ridge, η προβλεπτική τεχνική SVR-ANN έχει το χαμηλότερο σφάλμα πρόβλεψης σε σχέση με όλες τις υπόλοιπες και αυτό την καθιστά καταλληλότερη για την ακριβής πρόβλεψη των εκπομπών CO_2 στην Κίνα.

Η συνεχής αύξηση των εκπομπών του διοξειδίου του άνθρακα καθιστά αναγκαία την ύπαρξη ολοένα και περισσότερων ερευνών που στοχεύουν στην πρόβλεψή τους. Η παραπάνω ανασκόπηση, αναδεικνύει την αποτελεσματικότητα των αλγορίθμων μηχανικής μάθησης στην δημιουργία προβλεπτικών μοντέλων για τις εκπομπές CO_2. Ειδικότερα, οι αλγόριθμοι SARIMAX, GBDT, LightGBM, K-Nearest Neighbors και SVR-ANN έχουν αποδειχθεί ιδιαίτερα ακριβείς. Παράλληλα, η χρήση δεδομένων από ποικίλες περιόδους και τομείς προσφέρει ένα πλούσιο πλαίσιο για την ανάπτυξη και την βελτίωση μεθόδων εκτίμησης. Οι παραπάνω μελέτες παρέχουν πολύτιμες γνώσεις για το ζήτημα αυτό, προωθώντας την χρήση του κλάδου της μηχανικής μάθησης για την περιβαλλοντική διαχείριση και τη λήψη αποφάσεων.

ΚΕΦΑΛΑΙΟ 3: Θεωρητικό Υπόβαθρο

Το θεωρητικό υπόβαθρο της έρευνας είναι κρίσιμο για την κατανόηση των αλγορίθμων και των τεχνικών που θα χρησιμοποιήσουμε για την πρόβλεψη των εκπομπών του διοξειδίου του άνθρακα με την χρήση μηχανικής μάθησης. Στο παρόν κεφάλαιο, θα αναλύσουμε τις βασικές θεωρίες που υποστηρίζουν την έρευνα μας. Θα εξετάσουμε τους αλγορίθμους μηχανικής μάθησης και τις θεωρητικές βάσεις των μεθόδων που χρησιμοποιούνται.

3.1 Βασικοί Όροι

Παρακάτω θα αναλυθούν κάποιοι βασικοί όροι σχετικά με το θεωρητικό υπόβαθρο της διπλωματικής εργασίας:

- Μηχανική Μάθηση: Η Μηχανική Μάθηση είναι ένα υπο-πεδίο της Τεχνητής Νοημοσύνης που ασχολείται με τη δημιουργία αλγορίθμων και μοντέλων, επιτρέποντας στους υπολογιστές να μαθαίνουν από δεδομένα και να κάνουν προβλέψεις ή να λαμβάνουν αποφάσεις, χωρίς να απαιτείται ρητή προγραμματιστική καθοδήγηση. Στόχος της είναι η βελτίωση της απόδοσης των συστημάτων μέσω της εμπειρίας και της προσαρμογής. Οι αλγόριθμοι της μηχανικής μάθησης κατηγοριοποιούνται συνήθως σε τρεις βασικές κατηγορίες: επιβλεπόμενη μάθηση, μη επιβλεπόμενη μάθηση και ενισχυτική μάθηση [20].

- Supervised Machine Learning algorithms: Οι αλγόριθμοι εκείνοι που μαθαίνουν από ένα σύνολο δεδομένων το οποίο αποτελείται από ετικετοποιημένα[1] παραδείγματα, όπου κάθε σημείο δεδομένων περιλαμβάνει χαρακτηριστικά εισόδου και μια συνδεδεμένη ετικέτα εξόδου. Οι αλγόριθμοι αυτοί χρησιμοποιούν αυτό το σύνολο δεδομένων για να εκπαιδεύσουν ένα μοντέλο που μπορεί να προβλέψει την ετικέτα-στόχο για νέα άγνωστα δεδομένα βασισμένα στα χαρακτηριστικά εισόδου [21]

- Ensemble learning algorithm: Είναι μια τεχνική μηχανικής μάθησης που περιλαμβάνει τον συνδυασμών πολλαπλών μοντέλων για την βελτίωση της συνολικής απόδοσης και ακρίβειας των προβλέψεων. Μέσω της συγκέντρωσης των προβλέψεων από ποικίλα μοντέλα, επιτυγχάνονται καλύτερα αποτελέσματα από ότι τα μεμονωμένα μοντέλα [22].

- Hyperparameter tune: Είναι η διαδικασία βελτίωσης των υπερπαραμέτρων των μοντέλων μηχανικής μάθησης ούτως ώστε να επιτευχθεί η καλύτερη δυνατή απόδοσή τους. Οι υπερπαράμετροι είναι οι ρυθμίσεις εκείνες που καθορίζουν την διαδικασία εκπαίδευσης και την δομή που θα έχει το μοντέλο πρόβλεψης. Η κατάλληλη ρύθμισή τους μπορεί να επηρεάσει σε σημαντικό βαθμό την ακρίβεια και την ικανότητα γενίκευσης του μοντέλου [23].

- Υπερπροσαρμογή (overfitting): Είναι ένα φαινόμενο κατά το οποίο το μοντέλο μηχανικής μάθησης μαθαίνει πολύ καλά τα δεδομένα εκπαίδευσης, περιλαμβάνοντας και το θόρυβο ή τις τυχαίες διακυμάνσεις του, με συνέπεια να αποδίδει

[1] Δεδομένα που έχουν χαρακτηριστεί ή κατηγοριοποιηθεί με μία συγκεκριμένη ετικέτα ή κατηγορία.

πολύ καλά στα δεδομένα που δόθηκαν προς εκπαίδευση αλλά να αποτυγχάνει να προβλέψει αποτελέσματα, σε νέα άγνωστα δεδομένα [24].

- Πολύ-συγραμμικότητα (multicollinearity): Είναι ένα στατιστικό φαινόμενο που εμφανίζεται όταν δύο ή παραπάνω μεταβλητές σε ένα γραμμικό μοντέλο συσχετίζονται μεταξύ τους. Αποτέλεσμα αυτού του φαινομένου είναι η δυσκολία εκτίμησης των συντελεστών των μεταβλητών και η αναξιοπιστία στα στατιστικά συμπεράσματα. Σε κάποιες περιπτώσεις μπορεί να καταστήσει αδύνατη την ακριβή εκτίμηση των συντελεστών [25].

3.2 Αλγόριθμοι Μηχανικής Μάθησης

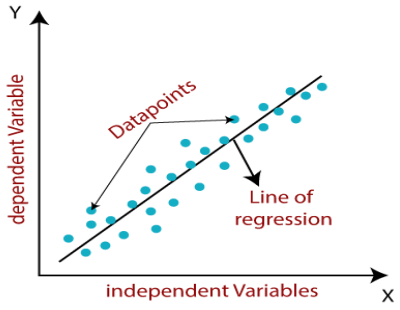
3.2.1 Simple Linear Regression

Η απλή γραμμική παλινδρόμηση (Simple Linear Regression) είναι ένας αλγόριθμος μηχανικής μάθησης με επίβλεψη (supervised machine learning algorithm) που βασίζεται στην συνώνυμη στατιστική μέθοδο.

Χρησιμοποιώντας τον ακριβή ορισμό, ως απλή γραμμική παλινδρόμηση θα ορίζαμε την διαδικασία για τον προσδιορισμό μιας γραμμής που αντιπροσωπεύει καλύτερα την γενική τάση ενός συνόλου [26]. Πιο συγκεκριμένα, η SLR είναι μια στατιστική μέθοδος που χρησιμοποιείται για να μοντελοποιήσει και να αναλύσει τη σχέση μεταξύ μιας εξαρτημένης μεταβλητής και μιας ή περισσοτέρων ανεξάρτητων μεταβλητών. Η εξίσωση που χρησιμοποιείται είναι συνήθως της μορφής:

$$y = b_0 + b_1 x,$$

όπου y είναι η εξαρτημένη μεταβλητή, x η ανεξάρτητη μεταβλητή, b_0 είναι η σταθερά και b_1 είναι ο συντελεστής κλίσης [27].



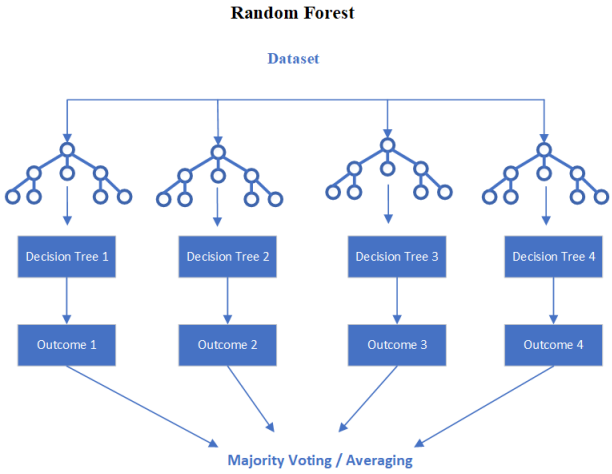
Εικόνα 4: Γραφική Αναπαράσταση Γραμμικής Παλινδρόμησης *[28]*

Η απλή γραμμική παλινδρόμηση έχει τις ρίζες της στη μέθοδο των ελάχιστων τετραγώνων (OLS) που αναπτύχθηκε από τον Adrien-Marie Legendre το 1805 και το Carl Friedrich Gauss το 1809. Αρχικά χρησιμοποιήθηκε στην αστρονομία για την επεξεργασία δεδομένων πειραμάτων [29]. Μια από τις βασικές προϋποθέσεις της SLR αποτελεί η ύπαρξη γραμμικής σχέσης μεταξύ των μεταβλητών, γεγονός που μπορεί να μην ισχύει σε διάφορα σύνολα δεδομένων. Επιπρόσθετα, είναι ευαίσθητη σε τιμές που ξεπερνούν το

όριο (outliers)[2] και θεωρεί δεδομένο πως τα σφάλματα έχουν κανονική κατανομή και ομοσκεδαστικότητα[3] (constant variance) [30]. Κάποια από τα πλεονεκτήματα αυτής της στατιστικής μεθόδου είναι η απλότητα και η ευκολία χρήσης της, η αποτελεσματικότητα της, καθώς παρέχει γρήγορα αποτελέσματα με σχετικά μικρές απαιτήσεις σε υπολογιστική ισχύ και η ερμηνευσιμότητά της.

3.2.2 Random Forest Regressor

Ο αλγόριθμος Random Forest είναι ένας ensemble learning αλγόριθμος που χρησιμοποιεί την τεχνική του bagging. Στο bagging, δημιουργούνται πολλαπλά αντίγραφα του συνόλου εκπαίδευσης με τυχαία δειγματοληψία και αντικατάσταση, και εκπαιδεύονται ξεχωριστά μοντέλα δέντρων απόφασης σε κάθε αντίγραφο. Εν συνεχεία, ο αλγόριθμος συνδυάζει τις προβλέψεις από όλα τα δέντρα για να δώσει την τελική πρόβλεψη, χρησιμοποιώντας τον μέσο όρο για την παλινδρόμηση ή την πλειοψηφική ψήφο για την ταξινόμηση. Ωστόσο, η μοναδικότητα του Random Forest σε σχέση με το απλό bagging έγκειται στο γεγονός ότι, κατά τη διαδικασία εκμάθησης, επιλέγει τυχαία υποσύνολα χαρακτηριστικών σε κάθε διαχωρισμό του δέντρου. Αυτό βοηθά στην αποφυγή της συσχέτισης των δέντρων, η οποία μπορεί να μειώσει την απόδοση του μοντέλου.



Εικόνα 5: Δομή και Λειτουργία του Random Forest

Η αποτελεσματικότητα του αλγορίθμου οφείλεται στην μείωση της διακύμανσης του τελικού μοντέλου μέσω της χρήση πολλαπλών δειγμάτων δεδομένων, γεγονός που συμβάλλει στην μείωση της υπέρ-προσαρμογής. Η «λεπτομέρεια» αυτή είναι πολύ σημαντική μιας και μειώνεται ο κίνδυνος το μοντέλο να προσπαθήσει να εξηγήσει μικρές τυχαίες παραλλαγές στο σύνολο των δεδομένων, οι οποίες μπορεί να περιλαμβάνουν ακραίες τιμές. Οι βασικότερες παράμετροι που πρέπει να ρυθμιστούν (hyperparameter tune) είναι ο αριθμός των δέντρων και το μέγεθος του τυχαίου υποσυνόλου των χαρακτηριστικών που εξετάζονται σε κάθε διαχωρισμό [21].

3.2.3 Gradient Boosting

Ο Gradient Boosting είναι ένας αλγόριθμος μηχανικής μάθησης με επίβλεψη (supervised machine learning algorithm) που χρησιμοποιείται για ταξινόμηση και παλινδρόμηση και αναπτύχθηκε από τον Jerome H. Friedman το 1999. Βασίζεται στη δημιουργία ενός συνόλου αδύναμων μαθητών (weak learners) και στον συνδυασμό τους για τη δημιουργία ενός ισχυρού μοντέλου [31]. Κάθε νέο δέντρο προσαρμόζεται στα υπολείμματα των προβλέψεων των προηγούμενων δέντρων, με τη μέθοδο της ελαχιστοποίησης της απώλειας

[2] Ένα σημείο δεδομένων που διαφέρει σημαντικά από άλλες παρατηρήσεις.
[3] Ίδια πεπερασμένη διακύμανση για όλα τα στοιχεία.

μέσω της κλίσης (gradient descent) [32]. Ο αλγόριθμος αυτός χρησιμοποιεί την τεχνική της ενίσχυσης (boosting) για να βελτιώσει την ακρίβεια πρόβλεψης, κατασκευάζοντας διαδοχικά δέντρα απόφασης, όπου κάθε δέντρο εκπαιδεύεται να διορθώσει τα σφάλματα του προηγούμενου.

Ο GB παρουσιάζει υψηλή ακρίβεια και δυνατότητα χειρισμού ποικίλων τύπων δεδομένων, καθιστώντας τον αποτελεσματικό σε προβλήματα ταξινόμησης και παλινδρόμησης, με δυνατότητα ενσωμάτωσης διάφορων τύπων συναρτήσεων απώλειας. Ωστόσο, έχει υψηλή υπολογιστική πολυπλοκότητα και απαιτεί λεπτομερή ρύθμιση των υπερπαραμέτρων για να αποφευχθεί η υπερπροσαρμογή. Ο αλγόριθμος αυτός χρησιμοποιείται ευρέως σε χρηματοοικονομικές προβλέψεις, ανίχνευση απατών, ιατρικές διαγνώσεις, και πολλές άλλες εφαρμογές όπου η ακρίβεια πρόβλεψης είναι κρίσιμη [32]. Η χρήση πολλαπλών δειγμάτων του αρχικού συνόλου δεδομένων μειώνει τη διακύμανση του τελικού μοντέλου, μειώνοντας έτσι τον κίνδυνο υπερπροσαρμογής [31].

3.2.4 Ridge Regression

Η Ridge Regression είναι μια τεχνική γραμμικής παλινδρόμησης που χρησιμοποιεί την κανονικοποίηση για να αντιμετωπίσει τα προβλήματα της πολύ-συγγραμμικότητας (multicollinearity). Αναπτύχθηκε από τους Hoerl και Kennard το 1970 και προσθέτει έναν όρο ποινής[4] στο μοντέλο για να μειωθεί η πολυπλοκότητα του [33].

Η βασική ιδέα της RR είναι να ελαχιστοποιήσει την ποσότητα των τετραγωνικών αποκλίσεων μεταξύ των προβλεπόμενων και πραγματικών τιμών, προσθέτοντας έναν όρο ποινής που βασίζεται στο τετράγωνο των συντελεστών. Η εξίσωση είναι της μορφής:

$$minimize \Sigma_{i=1}^{n}(y_i - \beta_0 - \Sigma_{j=1}^{p}\beta_i x_{ij})^2 + \lambda \Sigma_{j=1}^{p}\beta_j^2,$$

όπου λ είναι η παράμετρος κανονικοποίησης που καθορίζει τον βαθμό της ποινής.

Τα πλεονεκτήματα της Ridge Regression περιλαμβάνουν τη μείωση της διακύμανσης των εκτιμήσεων και τη χρησιμότητά της όταν υπάρχουν πολλές ανεξάρτητες μεταβλητές που σχετίζονται στενά μεταξύ τους. Επίσης, είναι λιγότερο ευαίσθητη στις τιμές που ξεπερνούν το όριο (outliers) και βελτιώνει την ακρίβεια του μοντέλου [33]. Ωστόσο, η επιλογή της παραμέτρου λ μπορεί να είναι δύσκολη και απαιτεί συχνά τη χρήση cross-validation. Επίσης, η Ridge Regression δεν θέτει ποτέ συντελεστές ακριβώς στο μηδέν, επομένως δεν είναι τόσο αποτελεσματική για την επιλογή χαρακτηριστικών όπως η Lasso Regression [34].

Η Ridge Regression χρησιμοποιείται ευρέως σε προβλήματα όπου οι ανεξάρτητες μεταβλητές είναι συσχετισμένες, όπως στην οικονομία, τη βιολογία και την κοινωνιολογία. Επίσης, είναι κατάλληλη για μοντελοποίηση δεδομένων υψηλής διάστασης, όπου ο αριθμός των χαρακτηριστικών είναι μεγαλύτερος από τον αριθμό των [33].

[4] Ο όρος ποινής είναι ένα στοιχείο που αποτρέπει τους συντελεστές του μοντέλου από το να γίνουν πολύ μεγάλοι, βοηθώντας το μοντέλο να παραμένει απλό και να αποφεύγει την υπερπροσαρμογή.

3.2.5 Lasso Regression

Η Lasso Regression, ή Least Absolute Shrinkage and Selection Operator, είναι μια γραμμική μέθοδος παλινδρόμησης που επιτρέπει την επιλογή χαρακτηριστικών και την κανονικοποίηση για τη βελτίωση της πρόβλεψης και της ερμηνευσιμότητας του μοντέλου. Εισάγει έναν όρο ποινής που βασίζεται στο απόλυτο μέγεθος των συντελεστών, ενθαρρύνοντας έτσι ορισμένους από αυτούς να γίνουν μηδενικοί και εξαλείφοντας τις λιγότερο σημαντικές μεταβλητές [34].

Η LR επεκτείνει τη γραμμική παλινδρόμηση προσθέτοντας έναν όρο ποινής στο κόστος της συνάρτησης, ο οποίος ορίζεται ως εξής:

$$J(\theta) = \Sigma_{i=1}^{m}(y_i - \theta^T x_i)^2 + \lambda \Sigma_{j=1}^{n}|\theta_j|,$$

όπου λ είναι η υπερπαράμετρος που καθορίζει την ένταση της ποινής. Σε αντίθεση με την Ridge Regressions, η LR μπορεί να οδηγήσει σε ακριβείς μηδενικούς συντελεστές.

Ένα σημαντικό πλεονέκτημα της LR είναι η ικανότητά της να εκτελεί ταυτόχρονα κανονικοποίηση και επιλογή χαρακτηριστικών, κάνοντας το μοντέλο πιο απλό και ερμηνεύσιμο. Αυτό την καθιστά ιδιαίτερα χρήσιμη σε περιπτώσεις όπου υπάρχουν πολλές άσχετες ή ασήμαντες μεταβλητές [34]. Επιπλέον, η Lasso μπορεί να χειριστεί την πολυδιάστατη πολυπλοκότητα και να βελτιώσει τη γενίκευση του μοντέλου [35]. Ωστόσο, ένα μειονέκτημα της LR είναι ότι μπορεί να επιλέξει μόνο μία μεταβλητή από μια ομάδα ισχυρά συσχετισμένων μεταβλητών, κάτι που μπορεί να μην είναι ιδανικό σε όλες τις περιπτώσεις. Επιπρόσθετα, όταν υπάρχουν πολλές σημαντικές μεταβλητές με μικρούς αλλά μηδενικούς συντελεστές, η Lasso μπορεί να μην αποδώσει βέλτιστα [27].

Η LR χρησιμοποιείται συχνά σε προβλήματα υψηλής διαστατικότητας[5] (dimensionality), όπως η ανάλυση γονιδιακής έκφρασης, η επιλογή σημαντικών χαρακτηριστικών σε μεγάλα σύνολα δεδομένων και η οικονομική πρόβλεψη. Είναι ιδιαίτερα αποτελεσματική σε περιπτώσεις όπου η απλότητα και η ερμηνευσιμότητας του μοντέλου είναι κρίσιμες [35].

3.2.6 XGBoost (Extreme Gradient Boosting)

Ο αλγόριθμος XGBoost, που αναπτύχθηκε από τον Tianqi Chen, είναι μια βελτιωμένη εκδοχή του Gradient Boosting. Έχει σχεδιαστεί για να είναι εξαιρετικά αποδοτικός και ταχύς, και χρησιμοποιείται εκτενώς σε διαγωνισμούς[6] μηχανικής μάθησης λόγω της ακρίβειας και της δυνατότητας χειρισμού μεγάλων δεδομένων. Η βασική αρχή του XGBoost βασίζεται στη δημιουργία μιας σειράς από αδύναμα μοντέλα (συνήθως δέντρα απόφασης), τα οποία εκπαιδεύονται διαδοχικά. Σε κάθε βήμα, το νέο δέντρο προσπαθεί να διορθώσει τα σφάλματα των προηγούμενων δέντρων. Χρησιμοποιεί μια συνάρτηση απώλειας για την ελαχιστοποίηση των σφαλμάτων και βελτιώνει συνεχώς την απόδοση του μοντέλου [31].

[5] Ο όρος αναφέρεται στον αριθμό των χαρακτηριστικών που χρησιμοποιούνται για να περιγράψουν κάθε σημείο δεδομένων σε ένα σύνολο δεδομένων.
[6] Σε πλατφόρμες όπως το Kaggle.

Ο XGBoost είναι γνωστός για την υψηλή του ακρίβεια, την ταχύτητα και την αποδοτικότητα στη χρήση πόρων. Έχει ενσωματωμένες λειτουργίες για την αποτροπή της υπερεκπαίδευσης (overfitting) και μπορεί να χειριστεί καλά ελλιπή δεδομένα . Ένας από τους βασικούς περιορισμούς του είναι η πολυπλοκότητα στην παραμετροποίηση. Επίσης, ο XGBoost μπορεί να είναι ευαίσθητος στην ποιότητα των δεδομένων εισόδου και απαιτεί σημαντικό χρόνο και υπολογιστική ισχύ για την εκπαίδευση μεγάλων μοντέλων [36].

Ο αλγόριθμος αυτός χρησιμοποιείται ευρέως σε εφαρμογές όπως η ανάλυση χρηματοοικονομικών δεδομένων, η πρόβλεψη ζήτησης, η ανάλυση κινδύνων και η ανίχνευση απάτης. Η αποτελεσματικότητα του XGBoost τον καθιστά έναν από τους δημοφιλέστερους στην κοινότητα της μηχανικής μάθησης [37].

3.2.7 Decision Tree

Ο Decision Tree είναι αλγόριθμος επίβλεψης μάθησης (supervised learning) που χρησιμοποιείται για την ταξινόμηση και την παλινδρόμηση. Αυτός ο αλγόριθμος κατασκευάζει ένα δέντρο απόφασης, το οποίο αποτελείται από κόμβους, που αντιπροσωπεύουν χαρακτηριστικά δεδομένων και διακλαδώσεις, που αντιπροσωπεύουν τις αποφάσεις που βασίζονται σε αυτά τα χαρακτηριστικά. Οι τελικοί κόμβοι (φύλλα) αντιπροσωπεύουν τις προβλέψεις ή τις ταξινομήσεις. Ο Decision Tree ξεκινά από τη ρίζα και εξετάζει τις τιμές ενός χαρακτηριστικού για να κάνει διακλαδώσεις στον επόμενο κόμβο, μέχρι να φτάσει σε ένα φύλλο. Το κριτήριο διαχωρισμού επιλέγεται για να μεγιστοποιήσει την πληροφόρηση που αποκτάται σε κάθε κόμβο, χρησιμοποιώντας μετρικές όπως η εντροπία ή το Gini impurity[7]. Το δέντρο συνεχίζει να διαχωρίζεται μέχρι να πληρούνται ορισμένα κριτήρια τερματισμού, όπως το ελάχιστο μέγεθος κόμβου ή το μέγιστο βάθος [38].

Τα πλεονεκτήματα του Decision Tree περιλαμβάνουν την εύκολη ερμηνεία και οπτικοποίηση, την απουσία ανάγκης για προ-επεξεργασία δεδομένων (π.χ. κλιμάκωση), την ικανότητα χειρισμού τόσο αριθμητικών όσο και κατηγορικών δεδομένων και την ικανότητα να αποτυπώνουν μη γραμμικές σχέσεις μεταξύ χαρακτηριστικών και στόχων [39]. Ωστόσο, έχει και μειονεκτήματα όπως την τάση για υπερπροσαρμογή (overfitting) ειδικά σε βαθιά δέντρα, την ευαισθησία σε μικρές παραλλαγές στα δεδομένα και την αρνητική επίδραση της απόδοσης από θορυβώδη δεδομένα [38].

Ο Decision Tree χρησιμοποιείται ευρέως σε πολλούς τομείς, όπως τα χρηματοοικονομικά, ο κλάδος της υγείας, η βιομηχανία και το μάρκετινγκ. Επιπλέον, χρησιμοποιείται για προβλήματα ταξινόμησης, όπως η αναγνώριση μοτίβων και η ανίχνευση απάτης, και για προβλήματα παλινδρόμησης, όπως η πρόβλεψη τιμών ακινήτων και η αξιολόγηση κινδύνων [39].

3.3 Μέθοδοι Επιλογής Χαρακτηριστικών

3.3.1 Mutual Information

Η αμοιβαία πληροφορία (Mutual Information) είναι ένα μέτρο της αλληλεξάρτησης μεταξύ δύο τυχαίων μεταβλητών. Στο πλαίσιο της επιλογής χαρακτηριστικών, η αμοιβαία

[7] Είναι ένα μέτρο που δείχνει πόσο «ανακατεμένα» είναι τα δεδομένα σε ένα κόμβο στα δέντρα απόφασης. Αν είναι 0, τότε τα δεδομένα ανήκουν όλα στην ίδια κατηγορία.

πληροφορία χρησιμοποιείται για να εκτιμηθεί η ποσότητα πληροφορίας που παρέχει ένα χαρακτηριστικό σχετικά με την εξαρτημένη μεταβλητή. Η υψηλή αμοιβαία πληροφορία σημαίνει ότι το χαρακτηριστικό είναι σημαντικό για την πρόβλεψη της εξαρτημένης μεταβλητής [40].

Η αμοιβαία πληροφορία μεταξύ δύο τυχαίων μεταβλητών X και Y ορίζεται ως:

$$I(X;Y) = \Sigma_{y\in Y}\Sigma_{x\in X} p(x,y)\log\frac{p(x,y)}{p(x)p(y)},$$

όπου $p(x,y)$ είναι η από κοινού πιθανότητα των X και Y, και $p(x)$ και $p(y)$ είναι οι περιθωριακές πιθανότητες των X και Yαντίστοιχα [41]. Η διαδικασία επιλογής χαρακτηριστικών περιλαμβάνει τα εξής βήματα [40]:

1. Υπολογισμός Αμοιβαίας Πληροφορίας: Για κάθε χαρακτηριστικό του συνόλου δεδομένων, υπολογίζεται η αμοιβαία πληροφορία με την εξαρτημένη μεταβλητή.
2. Κατάταξη Χαρακτηριστικών: Τα χαρακτηριστικά κατατάσσονται με βάση την αμοιβαία πληροφορία τους, από το πιο σημαντικό στο λιγότερο σημαντικό.
3. Επιλογή Καλύτερων Χαρακτηριστικών: Επιλέγονται τα κορυφαία χαρακτηριστικά που έχουν την υψηλότερη αμοιβαία πληροφορία για την κατασκευή του μοντέλου.

Τα πλεονεκτήματα της μεθόδου περιλαμβάνουν το ότι είναι μη παραμετρική και μπορεί να συλλάβει μη γραμμικές σχέσεις μεταξύ των μεταβλητών. Ωστόσο, η εκτίμηση της αμοιβαίας πληροφορίας μπορεί να είναι υπολογιστικά απαιτητική και επηρεάζεται από το μέγεθος του συνόλου δεδομένων και την παρουσία θορύβου [41].

3.3.2 Tree-based Feature Importance

Η μέθοδος επιλογής χαρακτηριστικών βασισμένη στη σημασία των χαρακτηριστικών των δέντρων απόφασης (Tree-based Feature Importance) χρησιμοποιεί μοντέλα δέντρων απόφασης για να αξιολογήσει τη σημασία κάθε χαρακτηριστικού. Οι σημασίες υπολογίζονται από το μέγεθος της μείωσης της αβεβαιότητας (χρησιμοποιώντας την κλίμακα Gini ή την εντροπία) που προκύπτει από τη διάσπαση ενός χαρακτηριστικού σε ένα δέντρο απόφασης. Τα δέντρα απόφασης είναι μη παραμετρικά μοντέλα που χρησιμοποιούν ιεραρχική δομή για να διαχωρίσουν τα δεδομένα. Η συνολική σημασία ενός χαρακτηριστικού είναι η μέση μείωση της αβεβαιότητας που παρέχεται από αυτό το χαρακτηριστικό σε όλο το δέντρο [39]. Η διαδικασία επιλογής χαρακτηριστικών περιλαμβάνει τα εξής βήματα:

1. Εκπαίδευση Μοντέλου Δέντρου Απόφασης: Εκπαίδευση μοντέλου δέντρου απόφασης ή συνόλου δέντρων (όπως τα τυχαία δάση).
2. Υπολογισμός Σημασίας Χαρακτηριστικών: Η σημασία κάθε χαρακτηριστικού υπολογίζεται από την μέση μείωση της αβεβαιότητας που προκύπτει από την χρήση του χαρακτηριστικού σε όλους τους κόμβους του δέντρου.
3. Κατάταξη Χαρακτηριστικών: Όλα τα χαρακτηριστικά κατατάσσονται με βάση την σημασία τους και επιλέγονται τα κορυφαία χαρακτηριστικά για την δημιουργία του μοντέλου.

Τα πλεονεκτήματα αυτής της μεθόδου περιλαμβάνουν την αντιμετώπιση μη γραμμικών σχέσεων και την ερμηνευσιμότητα της σημασίας των χαρακτηριστικών. Ωστόσο στα μειονεκτήματα συγκαταλέγονται δύο περιορισμοί. Τα χαρακτηριστικά με πολλά μοναδικά επίπεδα ή υψηλή διακύμανση τείνουν να έχουν μεγαλύτερη διακύμανση, με αποτέλεσμα να προκαλείται υπερεκτίμηση τους. Τέλος, τα αποτελέσματα μπορεί να είναι ευαίσθητα σε αλλαγές στο σύνολο δεδομένων εκπαίδευσης, ειδικά σε μικρά σύνολα δεδομένων.

3.3.3 Recursive Feature Elimination

Η RFE βασίζεται σε ένα μοντέλο, όπως η γραμμική παλινδρόμηση ή τα δέντρα απόφασης, για να εκτιμήσει τη σημασία των χαρακτηριστικών. Η σημασία κάθε χαρακτηριστικού υπολογίζεται από τους συντελεστές του μοντέλου ή άλλες μετρικές, και το λιγότερο σημαντικό χαρακτηριστικό αφαιρείται σε κάθε επανάληψη [42]. Η διαδικασία επιλογής χαρακτηριστικών περιλαμβάνει τα εξής βήματα:

1. Εκπαίδευση Μοντέλου: Αρχικά, γίνεται εκπαίδευση του μοντέλου με όλα τα διαθέσιμα χαρακτηριστικά του συνόλου δεδομένων.
2. Υπολογισμός Σημασίας Χαρακτηριστικών: Υπολογίζεται η σημασία κάθε χαρακτηριστικού. Για παράδειγμα σε γραμμικά μοντέλα, η σημασία μπορεί να προσδιοριστεί από τα απόλυτα μεγέθη των συντελεστών.
3. Αφαίρεση Χαρακτηριστικών: Το λιγότερο σημαντικό χαρακτηριστικό αφαιρείται.
4. Επανεκπαίδευση Μοντέλου: Το μοντέλο εκπαιδεύεται ξανά με το μειωμένο σύνολο χαρακτηριστικών.
5. Επανάληψη Διαδικασίας: Όλη η διαδικασία επαναλαμβάνεται έως ότου απομείνει ο επιθυμητός αριθμός χαρακτηριστικών.

Τα πλεονεκτήματα αυτής της τεχνικής είναι πρώτον πως μπορούν να μειωθούν τα προβλήματα πολύ-συγγραμμικότητας αφαιρώντας χαρακτηριστικά που σχετίζονται μεταξύ τους και δεύτερον ότι η διαδικασία είναι διαφανής και οι επιλεγμένες μεταβλητές είναι συχνά πιο ερμηνεύσιμες. Από την άλλη, η διαδικασία αυτή απαιτεί πολλαπλές εκπαιδεύσεις του προβλεπτικού μοντέλου, κάτι το οποίο είναι υπολογιστικά δαπανηρό [43].

3.4 Μέθοδοι Αξιολόγησης

Στην υποενότητα αυτή, θα παρουσιαστεί το θεωρητικό υπόβαθρο των μεθόδων αξιολόγησης που χρησιμοποιούνται για την εκτίμηση της απόδοσης των μοντέλων πρόβλεψης εκπομπών CO_2. Οι μέθοδοι αυτές παρέχουν τα απαραίτητα εργαλεία για την ανάλυση και σύγκριση των αποτελεσμάτων των μοντέλων.

3.4.1 Μέσο Τετραγωνικό Σφάλμα (MSE)

Το μέσο τετραγωνικό σφάλμα είναι ένα μέτρο του μέσου μεγέθους των σφαλμάτων που τα μοντέλα κάνουν στις προβλέψεις τους. Ορίζεται ως ο μέσος όρος των τετραγώνων των

διαφορών μεταξύ των πραγματικών τιμών και των προβλεπόμενων τιμών. Η συνάρτηση του MSE είναι η εξής:

$$MSE = \frac{1}{n}\Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

όπου y_i είναι η πραγματική τιμή και $\hat{y}_i$ είναι η προβλεπόμενη τιμή. ΤΟ MSE είναι πάντα μη αρνητικό και οι τιμές κοντά στο μηδέν δείχνουν καλύτερη απόδοση [27].

3.4.2 Συντελεστής Προσδιορισμού (R^2)

Ο συντελεστής προσδιορισμού, ή R^2, είναι ένα στατιστικό μέτρο που δείχνει το ποσοστό της διακύμανσης της εξαρτημένης μεταβλητής που μπορεί να εξηγηθεί από τις ανεξάρτητες μεταβλητές στο μοντέλο. Ορίζεται ως:

$$R^2 = 1 - \frac{\Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2}{\Sigma_{i=1}^{n}(y_i - \bar{y})^2},$$

όπου $\bar{y}$ είναι η μέση τιμή των πραγματικών τιμών. Το R^2 κυμαίνεται από 0 έως 1, με τιμές κοντά στο 1 να δείχνουν καλύτερη προσαρμογή του μοντέλου [27].

ΚΕΦΑΛΑΙΟ 4: Πειραματικό Μέρος

4.1 Εργαλεία

Σε αυτήν την υποενότητα, παρουσιάζονται τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση της παρούσας διπλωματικής εργασίας. Η επιλογή των κατάλληλων εργαλείων ήταν κρίσιμη για την επιτυχή ολοκλήρωση της ανάλυσης και των πειραμάτων. Τα εργαλεία αυτά περιλαμβάνουν λογισμικό και βιβλιοθήκες που διευκόλυναν την επεξεργασία των δεδομένων, την ανάλυση, την οπτικοποίηση και τη μοντελοποίηση.

Πιο συγκεκριμένα, για την ανάπτυξη του κώδικα χρησιμοποιήθηκε το Visual Studio Code (VS Code), ένα ισχυρό και ευέλικτο περιβάλλον ανάπτυξης. Επιπλέον, χρησιμοποιήθηκαν διάφορες επεκτάσεις (extensions) του VS Code που βελτίωσαν την παραγωγικότητα και την εμπειρία προγραμματισμού:

- autoDocstring - Python Docstring Generator: Δημιουργεί αυτόματα docstrings για τις συναρτήσεις της Python.
- Better Comments: Βελτιώνει τα σχόλια του κώδικα με ετικέτες, πληροφοριακά σχόλια κ.λπ..
- GitHub Pull Requests: Εργαλείο για τη διαχείριση αιτημάτων pull request στο GitHub[8].
- GitHub Repositories: Επιτρέπει την απομακρυσμένη περιήγηση και επεξεργασία αποθετηρίων GitHub.
- Jupyter: Υποστήριξη για Jupyter Notebooks, διαδραστικός προγραμματισμός και υπολογισμός.
- Jupyter Cell Tags: Υποστήριξη για ετικέτες κελιών στο Jupyter[9].
- Jupyter Keymap: Παρέχει keymaps[10] για notebooks.
- Pylance: Υψηλής απόδοσης language server για Python.
- Python: Υποστήριξη γλώσσας Python με IntelliSense, debugging και linting.
- Python Debugger: Επέκταση για αποσφαλμάτωση Python χρησιμοποιώντας το debugpy.
- Python Environment Manager: Διαχείριση περιβαλλόντων και πακέτων Python.
- Python Extension Pack: Δημοφιλές πακέτο επεκτάσεων για Python στο VS Code.
- Python Indent: Διορθώνει την εσοχή κώδικα Python.

[8] https://github.com/
[9] https://jupyter.org/
[10] Ένας χάρτης των συντομεύσεων πληκτρολογίου που χρησιμοποιούνται για την εκτέλεση συγκεκριμένων εντολών ή ενεργειών μέσα στο περιβάλλον του Jupyter Notebook.

- Remote Repositories: Επιτρέπει την απομακρυσμένη περιήγηση και επεξεργασία αποθετηρίων git.

Ακόμα, για την υλοποίηση της ανάλυσης και της μοντελοποίησης, χρησιμοποιήθηκαν διάφορες βιβλιοθήκες της Python. Αυτές οι βιβλιοθήκες παρείχαν τα εργαλεία για τη διαχείριση των δεδομένων, την ανάλυση, την προεπεξεργασία, την επιλογή χαρακτηριστικών, την εκπαίδευση και αξιολόγηση των μοντέλων, και την οπτικοποίηση των αποτελεσμάτων (Κώδικας 1). Πιο αναλυτικά:

- `pandas`: Χρησιμοποιήθηκε για τη φόρτωση, την επεξεργασία και την ανάλυση των δεδομένων.
- `numpy`: Παρείχε υποστήριξη για αριθμητικές πράξεις και πίνακες.
- `LabelEncoder`: Χρησιμοποιήθηκε για την κωδικοποίηση κατηγορικών δεδομένων σε αριθμητικές τιμές.
- `matplotlib` και `seaborn`: Χρησιμοποιήθηκαν για την οπτικοποίηση των δεδομένων και των αποτελεσμάτων.
- `pycountry`: Βοήθησε στην επαλήθευση και τον καθαρισμό των ονομάτων των χωρών.
- `train_test_split`: Χρησιμοποιήθηκε για τον διαχωρισμό των δεδομένων σε σύνολα εκπαίδευσης, επαλήθευσης και δοκιμών.
- `missingno`: Παρείχε εργαλεία για την οπτικοποίηση και ανάλυση των ελλειπουσών τιμών στα δεδομένα.
- `SelectKBest` και `mutual_info_regression`: Χρησιμοποιήθηκαν για την επιλογή των καλύτερων χαρακτηριστικών με βάση την αμοιβαία πληροφορία.
- `RandomForestRegressor`: Χρησιμοποιήθηκε για την επιλογή χαρακτηριστικών και για την εκπαίδευση μοντέλων τυχαίου δάσους.
- `RFE`: Χρησιμοποιήθηκε για την αναδρομική εξάλειψη χαρακτηριστικών.
- `LinearRegression`, `GradientBoostingRegressor`, `Ridge`, `Lasso`, `DecisionTreeRegressor`, `xgboost`: Διάφοροι αλγόριθμοι που χρησιμοποιήθηκαν για την εκπαίδευση και αξιολόγηση μοντέλων.
- `StandardScaler`: Χρησιμοποιήθηκε για την κλιμάκωση των δεδομένων.
- `mean_squared_error` και `r2_score`: Μετρικές που χρησιμοποιήθηκαν για την αξιολόγηση των μοντέλων.
- `GridSearchCV`: Χρησιμοποιήθηκε για την εύρεση των καλύτερων υπερπαραμέτρων.

4.2 Σύνολο Δεδομένων

Σε αυτό το σημείο θα παρουσιάσουμε το σύνολο δεδομένων που χρησιμοποιήθηκε για την εκπόνηση της διπλωματικής εργασίας. Τα δεδομένα αντλήθηκαν από το Word Bank Group[11] και συγκεκριμένα από την βάση δεδομένων World Development Indicators[12]. Η συγκεκριμένη βάση περιλαμβάνει αξιόπιστα και επικαιροποιημένα δεδομένα για πάνω από 200 χώρες, καλύπτοντας ένα ευρύ φάσμα δεικτών όπως η οικονομική ανάπτυξη, η υγεία, η εκπαίδευση, η φτώχεια και το περιβάλλον.

Το σύνολο δεδομένων περιλαμβάνει 5586 εγγραφές από 21 στήλες. Οι στήλες που περιλαμβάνονται στο σύνολο δεδομένων είναι οι εξής:

1. Time: Το έτος στο οποίο αναφέρονται τα δεδομένα.
2. Time Code: Κωδικός του έτους.
3. Country Name: Το όνομα της χώρας.
4. Country Code: Κωδικός της χώρας σύμφωνα με τα πρότυπα της World Bank.
5. CO2 emissions (kt): Εκπομπές διοξειδίου του άνθρακα σε χιλιάδες τόνους.
6. Energy use (kg of oil equivalent per capita): Κατανάλωση ενέργειας ανά κάτοικο σε κιλά ισοδύναμου πετρελαίου.
7. Forest area (sq. km): Έκταση δασικών περιοχών σε τετραγωνικά χιλιόμετρα.
8. GDP (current US$): ΑΕΠ σε τρέχουσες τιμές δολαρίων ΗΠΑ.
9. GDP growth (annual %): Ετήσιος ρυθμός αύξησης του ΑΕΠ σε ποσοστό.
10. Population, total: Ο συνολικός πληθυσμός της εκάστοτε χώρας.
11. Population growth (annual %): Ετήσιος ρυθμός αύξησης του πληθυσμού σε ποσοστό.
12. Renewable electricity output (% of total electricity output): Παραγωγή ανανεώσιμης ηλεκτρικής ενέργειας ως ποσοστό της συνολικής παραγωγής ηλεκτρικής ενέργειας.
13. Renewable energy consumption (% of total final energy consumption): Κατανάλωση ανανεώσιμης ενέργειας ως ποσοστό της συνολικής τελικής κατανάλωσης ενέργειας.
14. Urban population: Ο αστικός πληθυσμός.
15. Urban population growth (annual %): Ετήσιος ρυθμός αύξησης του αστικού πληθυσμού σε ποσοστό.
16. Population density (people per sq. km of land area): Πυκνότητα πληθυσμού σε ανθρώπους ανά τετραγωνικό χιλιόμετρο χερσαίας έκτασης.

[11] https://databank.worldbank.org/
[12] https://databank.worldbank.org/source/world-development-indicators/preview/on#

17. Electric power consumption (kWh per capita): Κατανάλωση ηλεκτρικής ενέργειας ανά κάτοικο σε κιλοβατώρες.
18. Access to electricity, urban (% of urban population): Πρόσβαση σε ηλεκτρική ενέργεια στις αστικές περιοχές ως ποσοστό του αστικού πληθυσμού.
19. Total natural resources rents (% of GDP): Έσοδα που προκύπτουν από την εξόρυξη φυσικών πόρων ως ποσοστό του ΑΕΠ.
20. Access to clean fuels and technologies for cooking (% of population): Πρόσβαση σε καθαρές καύσιμες ύλες και τεχνολογίες μαγειρέματος ως ποσοστό του πληθυσμού.
21. Terrestrial protected areas (% of total land area): Χερσαίες προστατευόμενες περιοχές ως ποσοστό της συνολικής χερσαίας έκτασης.

4.3 Μεθοδολογία

Στην παρούσα υποενότητα θα αναλυθεί η μεθοδολογία που ακολουθήθηκε για την επεξεργασία των δεδομένων, την επιλογή των χαρακτηριστικών, την εκπαίδευση και την αξιολόγηση των μοντέλων πρόβλεψης.

4.3.1 Φόρτωση και Αρχική Επισκόπηση του Συνόλου Δεδομένων

Η πρώτη φάση της μεθοδολογίας (Κώδικας 2) περιλαμβάνει τη φόρτωση και την αρχική επισκόπηση του συνόλου δεδομένων. Το σύνολο δεδομένων φορτώθηκε από ένα αρχείο CSV (*newMasterThesisDataSet.csv*) χρησιμοποιώντας τη βιβλιοθήκη `pandas`. Η φόρτωση πραγματοποιήθηκε με την εντολή `pd.read_csv()`, η οποία διαβάζει το αρχείο και το αποθηκεύει σε ένα DataFrame, μια δομή δεδομένων που παρέχει ισχυρά εργαλεία για την ανάλυση και τον χειρισμό δεδομένων.

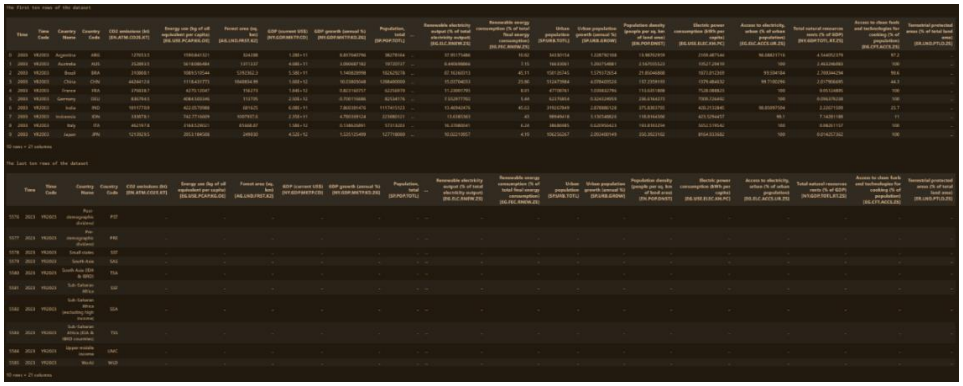
Ακολούθως, χρησιμοποιήθηκε η μέθοδος `data.info()` για να εμφανιστούν βασικές πληροφορίες σχετικά με το σύνολο δεδομένων, όπως ο αριθμός των μη κενών εγγραφών σε κάθε στήλη και οι τύποι δεδομένων. Αυτές οι πληροφορίες είναι κρίσιμες για την κατανόηση της πληρότητας των δεδομένων και για την αναγνώριση πιθανών προβλημάτων που θα πρέπει να αντιμετωπιστούν κατά την προεπεξεργασία. Οι μέθοδοι `data.columns` και `data.shape` χρησιμοποιήθηκαν για να καταγραφούν τα ονόματα των στηλών και οι διαστάσεις του DataFrame, αντίστοιχα, παρέχοντας μια συνοπτική εικόνα της δομής των δεδομένων.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5586 entries, 0 to 5585
Data columns (total 21 columns):
 #   Column                                                                  Non-Null Count  Dtype
---  ------                                                                  --------------  -----
 0   Time                                                                    5586 non-null   int64
 1   Time Code                                                               5586 non-null   object
 2   Country Name                                                            5586 non-null   object
 3   Country Code                                                            5586 non-null   object
 4   CO2 emissions (kt) [EN.ATM.CO2E.KT]                                     5586 non-null   object
 5   Energy use (kg of oil equivalent per capita) [EG.USE.PCAP.KG.OE]        5586 non-null   object
 6   Forest area (sq. km) [AG.LND.FRST.K2]                                   5586 non-null   object
 7   GDP (current US$) [NY.GDP.MKTP.CD]                                      5586 non-null   object
 8   GDP growth (annual %) [NY.GDP.MKTP.KD.ZG]                               5586 non-null   object
 9   Population, total [SP.POP.TOTL]                                         5586 non-null   object
 10  Population growth (annual %) [SP.POP.GROW]                              5586 non-null   object
 11  Renewable electricity output (% of total electricity output) [EG.ELC.RNEW.ZS]  5586 non-null   object
 12  Renewable energy consumption (% of total final energy consumption) [EG.FEC.RNEW.ZS]  5586 non-null   object
 13  Urban population [SP.URB.TOTL]                                          5586 non-null   object
 14  Urban population growth (annual %) [SP.URB.GROW]                        5586 non-null   object
 15  Population density (people per sq. km of land area) [EN.POP.DNST]       5586 non-null   object
 16  Electric power consumption (kWh per capita) [EG.USE.ELEC.KH.PC]         5586 non-null   object
 17  Access to electricity, urban (% of urban population) [EG.ELC.ACCS.UR.ZS]  5586 non-null   object
 18  Total natural resources rents (% of GDP) [NY.GDP.TOTL.RT.ZS]            5586 non-null   object
 19  Access to clean fuels and technologies for cooking (% of population) [EG.CFT.ACCS.ZS]  5586 non-null   object
 20  Terrestrial protected areas (% of total land area) [ER.LND.PTLD.ZS]     5586 non-null   object
dtypes: int64(1), object(20)
memory usage: 916.6+ KB

(5586, 21)
```

Εικόνα 6: Αρχική επισκόπηση του συνόλου δεδομένων

Για να επιθεωρηθούν οι αρχικές και τελικές εγγραφές του συνόλου δεδομένων και να διασφαλιστεί η ορθότητά τους, εκτυπώθηκαν οι πρώτες δέκα (`data.head(10)`) και οι τελευταίες δέκα γραμμές (`data.tail(10)`) του DataFrame. Αυτή η επιθεώρηση βοηθά στην αναγνώριση των τύπων δεδομένων που περιέχονται σε κάθε στήλη και επιτρέπει την προκαταρκτική αναγνώριση τυχόν ανωμαλιών ή ασυνέπειας στα δεδομένα.



Εικόνα 7: Απεικόνιση των πρώτων και των τελευταίων δέκα γραμμών του συνόλου δεδομένων

Τέλος, η εντολή `data.dtypes` χρησιμοποιήθηκε για να εμφανιστούν οι τύποι δεδομένων κάθε στήλης, επιβεβαιώνοντας έτσι τη συνέπεια και την καταλληλόλητά των δεδομένων για την επερχόμενη ανάλυση.

```
[10 rows x 21 columns]
Displaying the data types of each column
Time                                                                    int64
Time Code                                                              object
Country Name                                                           object
Country Code                                                           object
CO2 emissions (kt) [EN.ATM.CO2E.KT]                                    object
Energy use (kg of oil equivalent per capita) [EG.USE.PCAP.KG.OE]       object
Forest area (sq. km) [AG.LND.FRST.K2]                                  object
GDP (current US$) [NY.GDP.MKTP.CD]                                     object
GDP growth (annual %) [NY.GDP.MKTP.KD.ZG]                              object
Population, total [SP.POP.TOTL]                                        object
Population growth (annual %) [SP.POP.GROW]                             object
Renewable electricity output (% of total electricity output) [EG.ELC.RNEW.ZS]  object
Renewable energy consumption (% of total final energy consumption) [EG.FEC.RNEW.ZS]  object
Urban population [SP.URB.TOTL]                                         object
Urban population growth (annual %) [SP.URB.GROW]                       object
Population density (people per sq. km of land area) [EN.POP.DNST]      object
Electric power consumption (kWh per capita) [EG.USE.ELEC.KH.PC]        object
Access to electricity, urban (% of urban population) [EG.ELC.ACCS.UR.ZS]  object
Total natural resources rents (% of GDP) [NY.GDP.TOTL.RT.ZS]           object
Access to clean fuels and technologies for cooking (% of population) [EG.CFT.ACCS.ZS]  object
Terrestrial protected areas (% of total land area) [ER.LND.PTLD.ZS]    object
dtype: object
```

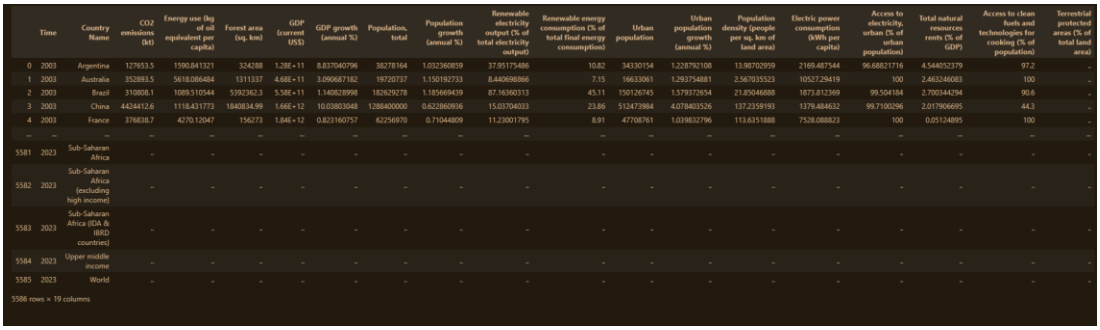
Εικόνα 8: Εμφάνιση των τύπων δεδομένων κάθε στήλης στο σύνολο δεδομένων

4.3.2 Προεπεξεργασία Δεδομένων

Η προεπεξεργασία δεδομένων (Κώδικας 3) περιλαμβάνει τη διαδικασία καθαρισμού και διαχείρισης των δεδομένων για τη βελτίωση της ποιότητας και της αξιοπιστίας τους. Αυτό το σημείο είναι κρίσιμο ούτως ώστε να διασφαλιστεί ότι τα δεδομένα που θα χρησιμοποιηθούν στην ανάλυση είναι καθαρά, πλήρη και κατάλληλα για την ανάλυση.

Σε αυτή τη φάση της προεπεξεργασίας, πρώτα αφαιρέθηκαν οι στήλες που δεν είναι χρήσιμες για την ανάλυση. Συγκεκριμένα, αφαιρέθηκαν οι στήλες «Time Code» και «Country Code». Αυτές οι στήλες θεωρήθηκαν μη χρήσιμες για την ανάλυση δεδομένων και επομένως αφαιρέθηκαν από το DataFrame για να απλοποιηθεί η δομή των δεδομένων. Στον κώδικα για αυτή την ενέργεια χρησιμοποιήθηκε η μέθοδος `drop()` της βιβλιοθήκης `pandas` με τα ορίσματα `axis=1` και `inplace=True`, η οποία εξασφαλίζει ότι οι στήλες αφαιρέθηκαν άμεσα από το αρχικό DataFrame `data` χωρίς να δημιουργηθεί νέο αντικείμενο.

Στη συνέχεια, πραγματοποιήθηκε μετονομασία των στηλών για να γίνουν τα ονόματα πιο κατανοητά. Συγκεκριμένα, ο κώδικας χρησιμοποίησε τη μέθοδο `rename()` της `pandas` για να αφαιρέσει περιττούς χαρακτήρες από τα ονόματα των στηλών, κρατώντας μόνο το τμήμα πριν από τον χαρακτήρα «[». Η συνάρτηση `lambda x: x.split('[')[0]` χρησιμοποιήθηκε ως παράμετρος στην `rename()`, η οποία επεξεργάζεται κάθε όνομα στήλης ξεχωριστά, διασπώντας το όνομα στη θέση του χαρακτήρα «[» και κρατώντας το πρώτο τμήμα. Το όρισμα `inplace=True` εξασφάλισε ότι η αλλαγή πραγματοποιήθηκε άμεσα στο αρχικό DataFrame `data` χωρίς να δημιουργηθεί νέο αντικείμενο. Αυτή η διαδικασία διευκολύνει την κατανόηση των ονομάτων των στηλών, κάνοντας τα δεδομένα πιο ευανάγνωστα και εύχρηστα. Για παράδειγμα, παρατηρώντας την Εικόνα 7 ή την Εικόνα 8 είναι εμφανές πως κάποιοι τίτλοι στηλών περιέχουν χαρακτήρες που δεν είναι χρήσιμοι όπως το υπογραμμισμένο τμήμα στον τίτλο «Energy use (kg of oil equivalent per capita) [EG.USE.PCAP.KG.OE]».

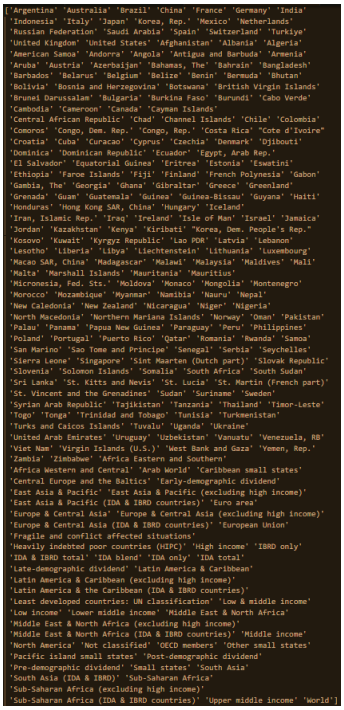


	Time	Country Name	CO2 emissions (kt)	Energy use (kg of oil equivalent per capita)	Forest area (sq. km)	GDP (current US$)	GDP growth (annual %)	Population, total	Population growth (annual %)	Renewable electricity output (% of total electricity output)	Renewable energy consumption (% of total final energy consumption)	Urban population	Urban population growth (annual %)	Population density (people per sq. km of land area)	Electric power consumption (kWh per capita)	Access to electricity, urban (% of urban population)	Total natural resources rents (% of GDP)	Access to clean fuels and technologies for cooking (% of population)	Terrestrial protected areas (% of total land area)
0	2003	Argentina	127653.5	1590.841321	324288	1.28E+11	8.837040796	38278364	1.032360859	37.95175486	10.82	34330154	1.228792108	13.98702959	2169.487544	96.68821716	4.544052379	97.2	..
1	2003	Australia	352893.5	5618.086484	1311337	4.68E+11	3.090687182	19720737	1.150192733	8.440698866	7.15	16633061	1.293754881	2.567035523	10527.29419	100	2.463246083	100	..
2	2003	Brazil	310808.1	1089.510544	5392362.3	5.58E+11	1.140828998	182629278	1.185669439	87.16360313	45.11	150126745	1.579372654	21.85046888	1873.812369	99.504184	2.700344294	90.6	..
3	2003	China	4424412.6	1118.431773	1840834.99	1.66E+12	10.03803048	1288400000	0.622860936	15.03704033	23.86	512473984	4.078403526	137.2359193	1379.484632	99.7100296	2.017906695	44.3	..
4	2003	France	376838.7	4270.12047	156273	1.84E+12	0.823160757	62256970	0.71044809	11.23001795	8.91	47708761	1.039832796	113.6351888	7528.088823	100	0.05124895	100	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
5581	2023	Sub-Saharan Africa	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
5582	2023	Sub-Saharan Africa (excluding high income)	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
5583	2023	Sub-Saharan Africa (IDA & IBRD countries)	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
5584	2023	Upper middle income	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
5585	2023	World	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..

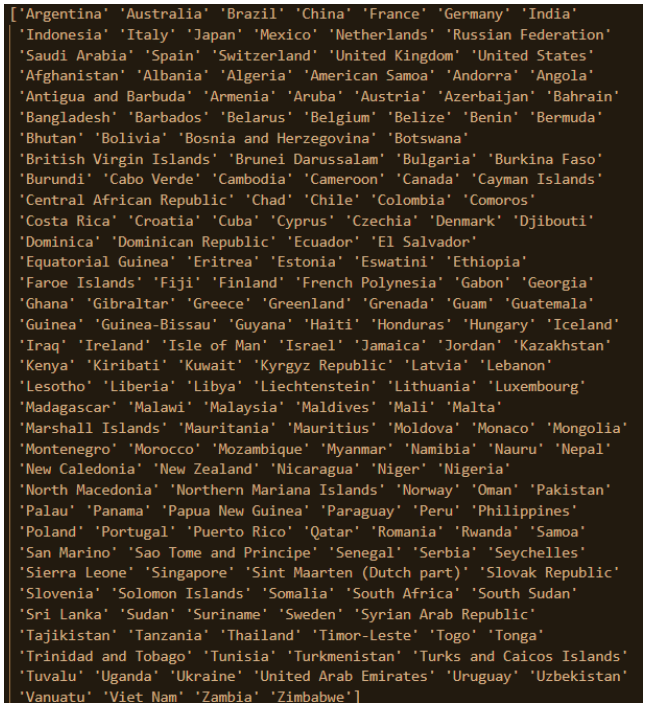
5586 rows × 19 columns

Εικόνα 9: Απεικόνιση του Dataset μετά την αφαίρεση μη χρήσιμων στηλών και την μετονομασία των στηλών για ευκολότερη κατανόηση

Στο επόμενο βήμα, πραγματοποιήθηκε έλεγχος και διόρθωση της στήλης «Country Name» επειδή παρατηρήθηκαν κάποια παράξενα δεδομένα στην προεπισκόπηση του αρχείου CSV. Πρώτα, εξήχθησαν και εκτυπώθηκαν οι μοναδικές τιμές της στήλης «Country Name» πριν την επεξεργασία, χρησιμοποιώντας τη μέθοδο `unique()` της `pandas`, βοηθώντας στον εντοπισμό των ανωμαλιών. Στη συνέχεια, δημιουργήθηκε μια συνάρτηση `is_valid_country()` για την επαλήθευση της εγκυρότητας των ονομάτων χωρών, η οποία χρησιμοποιεί τη βιβλιοθήκη `pycountry` για να ελέγξει αν κάθε όνομα χώρας είναι έγκυρο. Η μέθοδος `lookup` της `pycountry.countries` επιστρέφει την πληροφορία της χώρας αν το όνομα είναι έγκυρο και δημιουργεί εξαίρεση `LookupError` αν δεν βρεθεί. Αν το όνομα είναι έγκυρο, η συνάρτηση επιστρέφει την τιμή `True`, αλλιώς την τιμή `False`. Η συνάρτηση εφαρμόστηκε σε κάθε γραμμή της στήλης «Country Name», διατηρώντας μόνο τις γραμμές με έγκυρα ονόματα χωρών, και τελικά εκτυπώθηκαν οι μοναδικές τιμές της στήλης «Country Name» μετά την επεξεργασία, για να διασφαλιστεί ότι περιλαμβάνονται μόνο έγκυρες χώρες.



Εικόνα 11: Οι τιμές των χωρών πριν την εφαρμογή της is_valid_country()

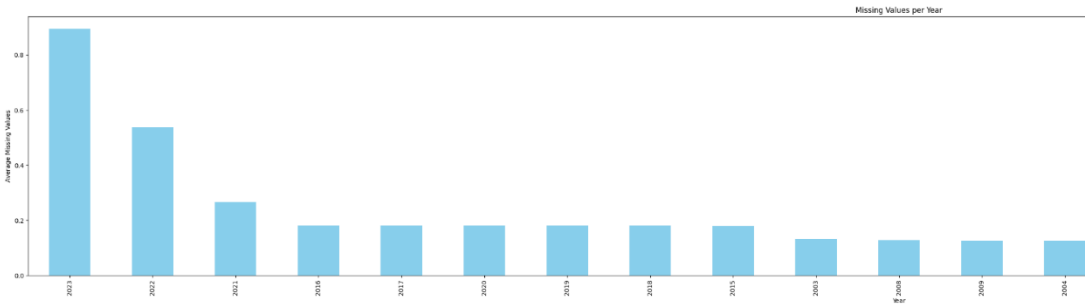


Εικόνα 10: Οι τιμές των χωρών μετά την εφαρμογή της is_valid_country()

Συνεχίζοντας, πραγματοποιήθηκε αντικατάσταση συγκεκριμένων τιμών που θεωρήθηκαν μη έγκυρες ή μη αντιπροσωπευτικές με `NaN`. Οι τιμές «..» και «0» αντικαταστάθηκαν με `NaN` χρησιμοποιώντας τη μέθοδο `replace()` της `pandas`. Αυτές οι τιμές αντιπροσωπεύουν ελλιπή ή μη διαθέσιμα δεδομένα και η αντικατάστασή τους με `NaN` διευκολύνει την επεξεργασία και την ανάλυση των ελλιπών δεδομένων. Η μέθοδος `replace()` αντικαθιστά όλες τις εμφανίσεις της τιμής «..» με `NaN` (χρησιμοποιώντας το `pd.NA`) και το όρισμα `inplace=True` εξασφαλίζει ότι η αλλαγή πραγματοποιείται άμεσα στο DataFrame `data` χωρίς να δημιουργηθεί νέο αντικείμενο. Αντίστοιχα, η μέθοδος

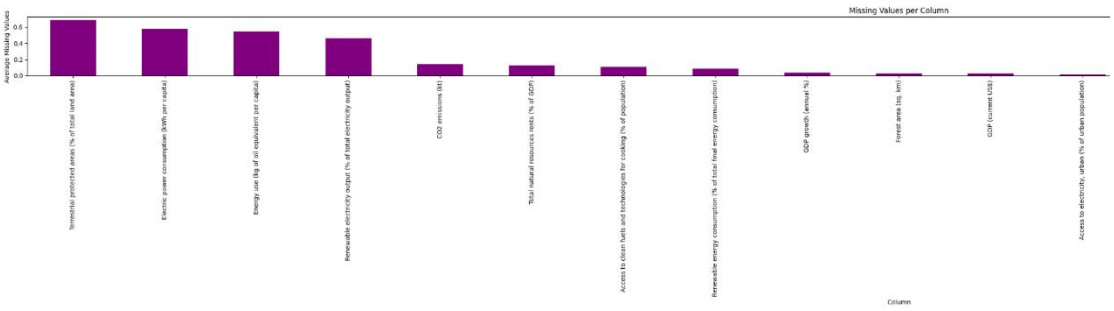
`replace()` αντικαθιστά όλες τις εμφανίσεις της τιμής «0» με `NaN`, με το όρισμα `inplace=True` να διασφαλίζει ότι η αλλαγή εφαρμόζεται άμεσα στο DataFrame `data`.

Η μεθοδολογία που ακολουθήθηκε για τον υπολογισμό και την οπτικοποίηση των ελλιπών τιμών εφαρμόστηκε με παρόμοιο τρόπο ανά έτος, ανά στήλη και ανά χώρα, με συγκεκριμένες διαφορές στον τρόπο υλοποίησης. Αρχικά, για τον υπολογισμό των ελλιπών τιμών ανά έτος, τα δεδομένα ομαδοποιήθηκαν βάσει του έτους (στήλη «Time»), και στη συνέχεια εφαρμόστηκε μια συνάρτηση που υπολόγιζε τον μέσο όρο των ελλιπών τιμών ανά γραμμή και ανά στήλη, ταξινομώντας τα αποτελέσματα με φθίνουσα σειρά. Τα αποτελέσματα αυτά απεικονίστηκαν γραφικά με τη χρήση ραβδογράμματος και στη συνέχεια καθορίστηκε ένα όριο 35% για την αποδεκτή ποσότητα ελλιπών τιμών, αφαιρώντας τα έτη που υπερέβαιναν αυτό το όριο.



Εικόνα 12: Missing Values Per Year

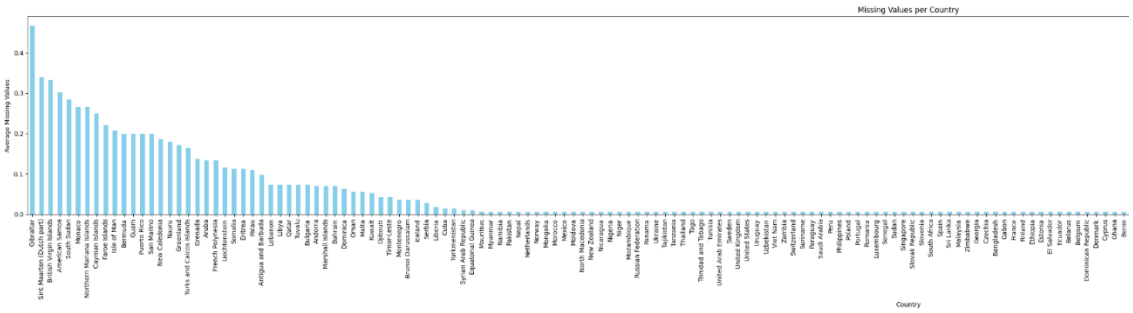
Αντίστοιχα, για τον υπολογισμό των ελλιπών τιμών ανά στήλη, χρησιμοποιήθηκε η μέθοδος `isnull().mean()` για να υπολογιστεί ο μέσος όρος των ελλιπών τιμών για κάθε στήλη, και τα αποτελέσματα ταξινομήθηκαν και απεικονίστηκαν γραφικά με ραβδόγραμμα. Καθορίστηκε και πάλι ένα όριο 35% για την αποδεκτή ποσότητα ελλιπών τιμών, και οι στήλες που υπερέβαιναν αυτό το όριο αφαιρέθηκαν από το DataFrame.



Εικόνα 13: Missing Values per Column

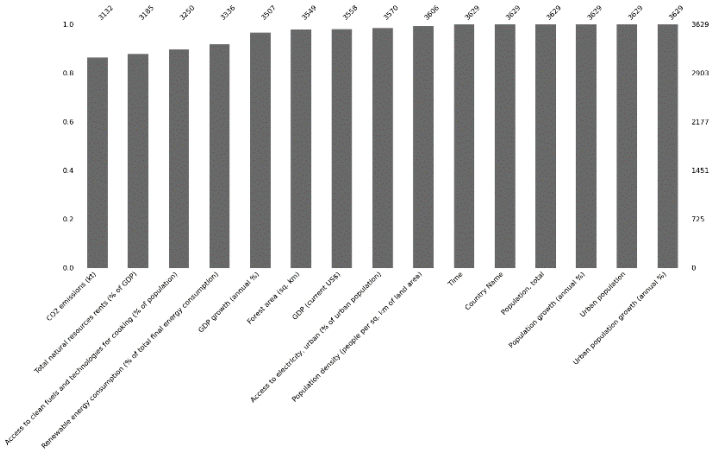
Τέλος, για τον υπολογισμό των ελλιπών τιμών ανά χώρα, τα δεδομένα ομαδοποιήθηκαν βάσει της στήλης «Country Name», και υπολογίστηκε ο μέσος όρος των ελλιπών

τιμών ανά χώρα με την ίδια μέθοδο όπως προηγουμένως. Τα αποτελέσματα ταξινομήθηκαν και απεικονίστηκαν γραφικά, καθορίστηκε όριο 35%, και αφαιρέθηκαν οι χώρες που υπερέβαιναν αυτό το όριο.



Εικόνα 14: Missing Values Per Country

Ωστόσο, επειδή ακόμα υπήρχαν γραμμές με ελλιπείς τιμές χρειάστηκε περαιτέρω ανάλυση του συνόλου δεδομένων. Πρώτα, οι ελλιπείς τιμές ομαδοποιήθηκαν ανά έτος (Time), και υπολογίστηκε ο μέσος όρος της ελλειπτικότητας (ποσοστό των ελλιπών τιμών) για κάθε στήλη. Αυτό επιτεύχθηκε με τη μέθοδο `groupby('Time')` και την εφαρμογή της συνάρτησης `lambda x: x.isnull().mean()`. Οι στήλες ταξινομήθηκαν με βάση τον μέσο όρο της ελλειπτικότητας σε φθίνουσα σειρά. Η βιβλιοθήκη `missingno` χρησιμοποιήθηκε για να δημιουργηθεί ένα ραβδόγραμμα που απεικονίζει τον αριθμό των ελλιπών τιμών για κάθε στήλη. Αυτή η οπτικοποίηση παρέχει μια σαφή εικόνα των στηλών που περιέχουν τα περισσότερα ελλιπή δεδομένα.



Εικόνα 15: Ελλιπείς τιμές ανά στήλη

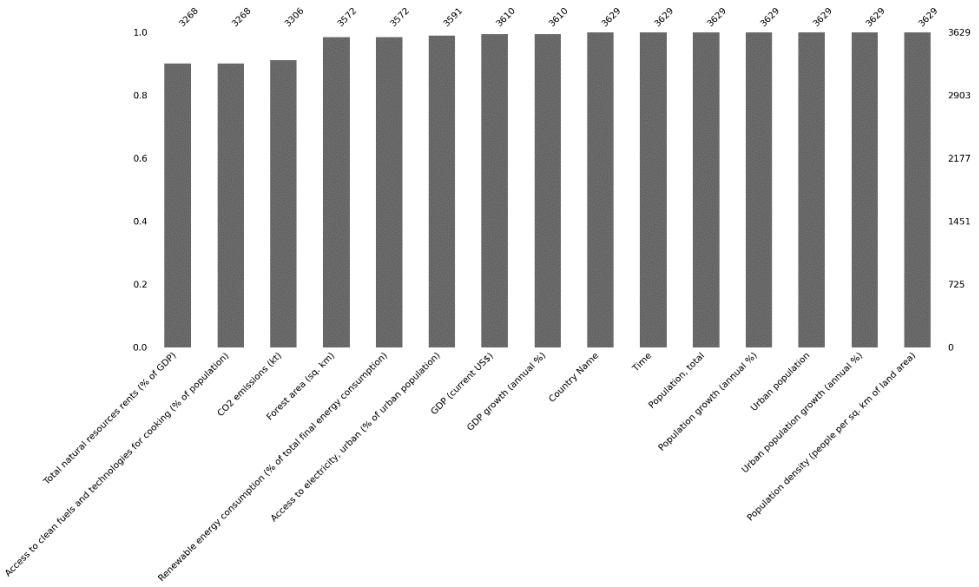
Ακόμα, επιλέχθηκαν όλες οι στήλες που θα έπρεπε να είναι αριθμητικές αλλά ήταν τύπου `'object'`, εξαιρώντας τις στήλες «Time» και «Country Name». Οι επιλεγμένες στήλες μετατράπηκαν σε αριθμητικές τιμές (float) χρησιμοποιώντας τη μέθοδο

`apply(pd.to_numeric, errors='coerce')`. Η παράμετρος `errors='coerce'` εξασφαλίζει ότι οποιεσδήποτε τιμές που δεν μπορούν να μετατραπούν σε αριθμητικές τιμές θα αντικατασταθούν με `NaN`.

```
Data columns (total 15 columns):
 #   Column                                                    Non-Null Count  Dtype
---  ------                                                    --------------  -----
 0   Time                                                      3629 non-null   int64
 1   Country Name                                              3629 non-null   object
 2   CO2 emissions (kt)                                        3132 non-null   float64
 3   Forest area (sq. km)                                      3549 non-null   float64
 4   GDP (current US$)                                         3558 non-null   float64
 5   GDP growth (annual %)                                     3507 non-null   float64
 6   Population, total                                         3629 non-null   int64
 7   Population growth (annual %)                              3629 non-null   float64
 8   Renewable energy consumption (% of total final energy consumption)  3336 non-null   float64
 9   Urban population                                          3629 non-null   int64
 10  Urban population growth (annual %)                        3629 non-null   float64
 11  Population density (people per sq. km of land area)       3606 non-null   float64
 12  Access to electricity, urban (% of urban population)      3570 non-null   float64
 13  Total natural resources rents (% of GDP)                  3185 non-null   float64
 14  Access to clean fuels and technologies for cooking (% of population)  3250 non-null   float64
dtypes: float64(11), int64(3), object(1)
```

Εικόνα 16: Απεικόνιση των ορθών data types

Πρόσθετα, εφαρμόστηκε προσαρμοσμένη αποκατάσταση των ελλιπών τιμών (custom imputation) και οπτικοποίηση των αποτελεσμάτων της αποκατάστασης. Αρχικά, αντικαταστάθηκαν οι τιμές `pd.NA` με `np.nan` για να αποφευχθούν πιθανά σφάλματα κατά τη διαδικασία της αποκατάστασης. Στη συνέχεια, τα δεδομένα ομαδοποιήθηκαν με βάση τη στήλη «Country Name» και οι ελλιπείς τιμές συμπληρώθηκαν με τον μέσο όρο της κάθε στήλης για κάθε χώρα και καθ' όλη την διάρκεια των ετών. Μετά την αποκατάσταση, η στήλη «Country Name» προστέθηκε ξανά στην πρώτη θέση του DataFrame για να διατηρηθεί η αρχική δομή των δεδομένων. Ακολούθως, πραγματοποιήθηκε έλεγχος για να διαπιστωθεί αν παραμένουν ελλιπείς τιμές στο DataFrame μετά την αποκατάσταση και την αφαίρεση των γραμμών. Αν υπήρχαν ακόμη ελλιπείς τιμές, έγινε σχετικός έλεγχος για υπολειπόμενες ελλιπείς τιμές. Τέλος, έγινε οπτικοποίηση των ελλιπών τιμών μετά την αποκατάσταση. Αρχικά, υπολογίστηκε η μέση ελλειπτικότητα (nullity) για κάθε στήλη ανά έτος και τα δεδομένα ταξινομήθηκαν με βάση την μέση ελλειπτικότητα των στηλών σε φθίνουσα σειρά. Αυτά τα ταξινομημένα δεδομένα χρησιμοποιήθηκαν για τη δημιουργία γραφημάτων που απεικονίζουν τις ελλιπείς τιμές, χρησιμοποιώντας τη βιβλιοθήκη `missingno`.



Εικόνα 17: Ελλιπείς τιμές ανά στήλη μετά το custom imputation

Παρότι είχε εφαρμοστεί custom imputation, μπορεί κάποιος να παρατηρήσει ότι είχαν παραμείνει ελάχιστες ελλιπείς τιμές. Για τον λόγο αυτό, υπολογίστηκε το ποσοστό των ελλιπών τιμών (`NaN`) για κάθε στήλη εντός κάθε χώρας. Αυτό επιτεύχθηκε με την ομαδοποίηση των δεδομένων ανά χώρα (Country Name) και την εφαρμογή της συνάρτησης `lambda x: x.isnull().mean()`, η οποία υπολογίζει το ποσοστό των ελλιπών τιμών. Φιλτραρίστηκαν οι στήλες που είχαν 100% ελλιπείς τιμές για οποιαδήποτε χώρα. Αυτό έγινε με την επιλογή των στηλών όπου το ποσοστό των `NaN` ήταν 1.0 και την αφαίρεση των γραμμών που ήταν πλήρως κενές. Εντοπίστηκαν οι γραμμές που είχαν τουλάχιστον μία στήλη εντελώς κενή. Αυτό επιτεύχθηκε με τη χρήση της συνάρτησης `apply()` και του `lambda` που ελέγχει αν υπάρχει κάποια στήλη από τις πλήρως ελλιπείς στήλες που είναι κενή. Οι ταυτοποιημένες γραμμές αφαιρέθηκαν από το σύνολο δεδομένων για να διασφαλιστεί ότι δεν υπάρχουν γραμμές με εντελώς κενές στήλες. Έγινε έλεγχος για να διαπιστωθεί αν παραμένουν ελλιπείς τιμές μετά την αφαίρεση των γραμμών. Αν υπήρχαν ακόμη ελλιπείς τιμές, σχετικό μήνυμα θα εμφανιζόταν στην οθόνη. Τέλος, το τελικό καθαρισμένο σύνολο δεδομένων εμφανίστηκε για επιθεώρηση χρησιμοποιώντας τη μέθοδο `display()`. Τα «καθαρισμένα» δεδομένα αποθηκεύτηκαν σε ένα αρχείο Excel για μελλοντική χρήση, εξασφαλίζοντας ότι η τελική έκδοση των δεδομένων είναι διαθέσιμη για ανάλυση.

	Country Name	Time	CO2 emissions (kt)	Forest area (sq. km)	GDP (current US$)	GDP growth (annual %)	Population, total	Population growth (annual %)	Renewable energy consumption (% of total final energy consumption)	Urban population	Urban population growth (annual %)	Population density (people per sq. km of land area)	Access to electricity, urban (% of urban population)	Total natural resources rents (% of GDP)	Access to clean fuels and technologies for cooking (% of population)
0	Argentina	2003	1.276535e+05	324288.000	1.280000e+11	8.837041	38278164	1.032361	10.820000	34330154	1.228792	13.987030	96.688217	4.544052	97.2
1	Australia	2003	3.528935e+05	1311337.000	4.680000e+11	3.090687	19720737	1.150193	7.150000	16633061	1.293755	2.567036	100.000000	2.463246	100.0
2	Brazil	2003	3.108081e+05	5392362.300	5.580000e+11	1.140829	182629278	1.185669	45.110000	150126745	1.579373	21.850469	99.504184	2.700344	90.6
3	China	2003	4.424413e+06	1840834.990	1.660000e+12	10.038030	1288400000	0.622861	23.860000	512473984	4.078404	137.235919	99.710030	2.017907	44.3
4	France	2003	3.768387e+05	156273.000	1.840000e+12	0.823161	62256970	0.710448	8.910000	47708761	1.039833	113.635189	100.000000	0.051249	100.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4996	Uzbekistan	2021	1.170233e+05	37155.000	6.960061e+10	7.403368	34915100	1.975706	1.292778	17606637	1.997521	79.234908	100.000000	20.465161	82.8
4997	Vanuatu	2021	1.194667e+02	4423.000	9.716361e+08	0.648653	319137	2.362742	22.730000	81907	2.909814	26.180230	97.011765	0.568070	6.9
4999	Viet Nam	2021	1.745950e+05	147189.900	3.660000e+11	2.561564	97468029	0.844182	35.132778	37088534	2.733031	310.973232	100.000000	2.547017	96.1
5003	Zambia	2021	4.118822e+03	446258.133	2.209642e+10	6.234922	19473125	2.840806	85.945556	8800295	4.094430	26.195032	85.717453	35.264148	10.2
5004	Zimbabwe	2021	1.021708e+04	173985.100	2.837124e+10	8.468017	15993524	2.045715	80.666111	5166388	2.234724	41.342960	85.328506	6.398452	30.3

Εικόνα 18: Τελική μορφή πλήρως προεπεξεργασμένου συνόλου δεδομένων

4.3.3 Επιλογή Χαρακτηριστικών και Κλιμάκωση

Σε αυτή την υποενότητα, πραγματοποιήθηκε η επιλογή χαρακτηριστικών (Κώδικας 4) και η κλιμάκωση των δεδομένων, προκειμένου να προετοιμαστεί το σύνολο δεδομένων για μοντελοποίηση.

Αρχικά, τα δεδομένα χωρίστηκαν σε χαρακτηριστικά (features) και τη μεταβλητή στόχο (target variable). Τα χαρακτηριστικά περιλαμβάνουν όλες τις στήλες εκτός από την «CO2 emissions (kt)», η οποία αποτελεί τη μεταβλητή στόχο. Η target variable κανονικοποιήθηκε (standardized) χρησιμοποιώντας τον `StandardScaler` από τη βιβλιοθήκη `sklearn`. Αυτό το βήμα είναι σημαντικό για να διασφαλιστεί ότι η κατανομή της μεταβλητής στόχου έχει μέση τιμή 0 και τυπική απόκλιση 1, διευκολύνοντας τη διαδικασία εκπαίδευσης των μοντέλων. Η στήλη «Country Name» κωδικοποιήθηκε χρησιμοποιώντας τον `LabelEncoder`. Αυτό το βήμα είναι απαραίτητο γιατί τα μοντέλα μηχανικής μάθησης δεν μπορούν να επεξεργαστούν άμεσα κατηγορικές μεταβλητές. Η κωδικοποίηση μετατρέπει τις κατηγορικές τιμές σε αριθμητικές, επιτρέποντας στα μοντέλα να τις χρησιμοποιήσουν.

Στην συνέχεια, πραγματοποιείται η διάσπαση του συνόλου δεδομένων σε εκπαιδευτικό, επικύρωσης και δοκιμαστικό σύνολο. Αυτή η διαδικασία είναι απαραίτητη για να διασφαλιστεί ότι το μοντέλο μπορεί να εκπαιδευτεί, να επικυρωθεί και να δοκιμαστεί σε διαφορετικά υποσύνολα των δεδομένων, εξασφαλίζοντας έτσι την αξιοπιστία και τη γενίκευση των αποτελεσμάτων του μοντέλου. Αρχικά, το αρχικό σύνολο δεδομένων χωρίζεται σε δύο μέρη: το εκπαιδευτικό σύνολο (training set) και ένα προσωρινό σύνολο (temporary set). Το εκπαιδευτικό σύνολο περιλαμβάνει το 60% των δεδομένων, ενώ το προσωρινό σύνολο περιλαμβάνει το υπόλοιπο 40%. Αυτό γίνεται χρησιμοποιώντας τη μέθοδο train_test_split από τη βιβλιοθήκη `sklearn`. Ορίζεται επίσης μια σταθερή τυχαία κατάσταση (`random_state=42`) για να διασφαλιστεί ότι τα αποτελέσματα της διάσπασης μπορούν να αναπαραχθούν.

Ακολούθως, πραγματοποιήθηκε η επιλογή χαρακτηριστικών με τη χρήση διαφορετικών μεθόδων και η εφαρμογή των ίδιων παραμέτρων κλιμάκωσης στα σύνολα εκπαίδευσης, επικύρωσης και δοκιμής. Οι τρεις μέθοδοι που χρησιμοποιήθηκαν για την επιλογή χαρακτηριστικών είναι η Mutual Information, η Tree-based Feature Importance και η Recursive Feature Elimination. Αρχικά, χρησιμοποιήθηκε η μέθοδος SelectKBest με τη συνάρτηση `mutual_info_regression` για την επιλογή των 10 πιο σημαντικών χαρακτηριστικών με βάση την αμοιβαία πληροφορία. Αυτή η μέθοδος αξιολογεί το κατά πόσο εξαρτάται η μεταβλητής στόχος από το εκάστοτε χαρακτηριστικό. Εν συνεχεία, χρησιμοποιήθηκε ένας `Random Forest Regressor` για να υπολογίσει τις σημασίες των χαρακτηριστικών. Τα 10 πιο σημαντικά χαρακτηριστικά επιλέχθηκαν βάσει της σημασίας τους. Πρόσθετα, χρησιμοποιήθηκε το μοντέλο γραμμικής παλινδρόμησης (Linear Regression) και η μέθοδος RFE για την αναδρομική εξάλειψη των χαρακτηριστικών, επιλέγοντας τα 10 πιο σημαντικά χαρακτηριστικά αντίστοιχα.

Σε τελικό στάδιο, πραγματοποιήθηκε η κλιμάκωση των επιλεγμένων χαρακτηριστικών χρησιμοποιώντας τον `StandardScaler`, εξασφαλίζοντας ότι όλα τα χαρακτηριστικά έχουν την ίδια κλίμακα. Αρχικά, τα χαρακτηριστικά που επιλέχθηκαν με τη μέθοδο της

Mutual Information κλιμακώθηκαν στο πλήρες σύνολο δεδομένων και στα σύνολα εκπαίδευσης, επικύρωσης και δοκιμής. Στη συνέχεια, εφαρμόστηκε η ίδια διαδικασία για τα χαρακτηριστικά που επιλέχθηκαν με τη μέθοδο Tree-based Feature Importance και την Recursive Feature Elimination. Αυτό διασφαλίζει ότι τα χαρακτηριστικά σε όλα τα σύνολα δεδομένων έχουν την ίδια κλίμακα, αποτρέποντας την υπερβολική επιρροή των χαρακτηριστικών με μεγάλες τιμές. Τέλος, εκτυπώθηκαν τα επιλεγμένα χαρακτηριστικά για κάθε μία από τις τρεις μεθόδους, παρέχοντας μια αναφορά για τα χαρακτηριστικά που κλιμακώθηκαν και χρησιμοποιήθηκαν στα σύνολα δεδομένων.

```
Selected Features (Mutual Information):
 Index(['Country Name', 'Forest area (sq. km)', 'GDP (current US$)',
       'Population, total',
       'Renewable energy consumption (% of total final energy consumption)',
       'Urban population', 'Urban population growth (annual %)',
       'Population density (people per sq. km of land area)',
       'Access to electricity, urban (% of urban population)',
       'Access to clean fuels and technologies for cooking (% of population)'],
      dtype='object')

Selected Features (Tree-based):
 ['GDP (current US$)' 'Urban population' 'Population, total'
 'Renewable energy consumption (% of total final energy consumption)'
 'Forest area (sq. km)'
 'Population density (people per sq. km of land area)'
 'GDP growth (annual %)' 'Urban population growth (annual %)'
 'Country Name'
 'Access to clean fuels and technologies for cooking (% of population)']

Selected Features (RFE):
 Index(['Country Name', 'Time', 'GDP growth (annual %)',
       'Population growth (annual %)',
       'Renewable energy consumption (% of total final energy consumption)',
       'Urban population growth (annual %)',
       'Population density (people per sq. km of land area)',
       'Access to electricity, urban (% of urban population)',
       'Total natural resources rents (% of GDP)',
       'Access to clean fuels and technologies for cooking (% of population)'],
      dtype='object')
```

Εικόνα 19: Επιλεγμένα χαρακτηριστικά με τρεις διαφορετικές τεχνικές

4.3.4 Αρχικοποίηση Μοντέλων και Συναρτήσεις Αξιολόγησης

Σε αυτό το στάδιο της μεθοδολογίας (Κώδικας 5), πραγματοποιήθηκε η αρχικοποίηση διαφόρων μοντέλων και η δημιουργία μιας συνάρτησης αξιολόγησης των μοντέλων. Αυτή η διαδικασία διασφαλίζει ότι τα μοντέλα εκπαιδεύονται, αξιολογούνται και επικυρώνονται με συνεπή και αξιόπιστο τρόπο.

Διάφορα μοντέλα αρχικοποιήθηκαν για χρήση στην εκπαίδευση και αξιολόγηση όπως ο αλγόριθμος Linear Regression, ο Random Forest Regressor, ο Gradient Boosting Regressor, ο XGBoost, ο Ridge Regression, ο Lasso Regression, και ο Decision Tree Regressor. Για κάθε μοντέλο, ορίστηκε μια σταθερή τυχαία κατάσταση (`random_state=40`) για να διασφαλιστεί ότι τα αποτελέσματα μπορούν να αναπαραχθούν.

Παράλληλα, δημιουργήθηκε μια συνάρτηση αξιολόγησης για να αξιολογήσει την απόδοση των μοντέλων χρησιμοποιώντας τα σύνολα εκπαίδευσης και επικύρωσης. Η συ-

νάρτηση `evaluate_model` εκπαιδεύει το μοντέλο, προβλέπει τις τιμές για τα σύνολα εκπαίδευσης και επικύρωσης, και υπολογίζει τα μέτρα απόδοσης όπως το μέσο τετραγωνικό σφάλμα (MSE) και τον συντελεστή προσδιορισμού (R2).

4.3.5 Εκπαίδευση και Αρχική Αξιολόγηση Μοντέλων

Εδώ (Κώδικας 6) πραγματοποιήθηκε η εκπαίδευση και η αρχική αξιολόγηση διαφόρων αλγορίθμων μηχανικής μάθησης με τη χρήση τριών διαφορετικών τεχνικών επιλογής χαρακτηριστικών: Mutual Information, Tree-based Feature Importance και Recursive Feature Elimination (RFE). Ο στόχος ήταν να προσδιοριστεί ποια τεχνική επιλογής χαρακτηριστικών αποδίδει καλύτερα για κάθε αλγόριθμο.

Οι αλγόριθμοι που χρησιμοποιήθηκαν , όπως προαναφέραμε είναι οι Linear Regression, Random Forest Regressor, Gradient Boosting Regressor, XGBoost, Ridge Regression, Lasso Regression και Decision Tree Regressor. Για κάθε αλγόριθμο, εφαρμόστηκε η διαδικασία εκπαίδευσης και αξιολόγησης χρησιμοποιώντας τα χαρακτηριστικά που επιλέχθηκαν από τις τρεις τεχνικές επιλογής χαρακτηριστικών.

Χρησιμοποιήθηκε η συνάρτηση `evaluate_model` για να εκπαιδευτεί και να αξιολογηθεί κάθε αλγόριθμος με τα επιλεγμένα χαρακτηριστικά. Η αξιολόγηση περιλάμβανε τον υπολογισμό του μέσου τετραγωνικού σφάλματος (MSE) και του συντελεστή προσδιορισμού (R2) στα σύνολα εκπαίδευσης και επικύρωσης, όπως έχει αναφερθεί παραπάνω.

4.3.6 Εκπαίδευση των Μοντέλων

Σε αυτό το βήμα της μεθοδολογίας (Κώδικας 7), εκπαιδεύτηκαν και βελτιστοποιήθηκαν διάφοροι αλγόριθμοι μηχανικής μάθησης χρησιμοποιώντας παρόμοιες υπερπαραμέτρους για να διασφαλιστεί η δυνατότητα αξιόπιστης σύγκρισης των αποτελεσμάτων. Για κάθε αλγόριθμο, ορίστηκε ένα πλέγμα υπερπαραμέτρων και εφαρμόστηκε `GridSearchCV` για την εύρεση των βέλτιστων τιμών.

Συγκεκριμένα, για τον αλγόριθμο Linear Regression, χρησιμοποιήθηκαν τα χαρακτηριστικά που επιλέχθηκαν με βάση τη σημασία τους από δέντρα απόφασης χωρίς την ανάγκη βελτιστοποίησης υπερπαραμέτρων. Για τους αλγορίθμους Random Forest, Gradient Boosting και XGBoost, οι υπερπαράμετροι περιλάμβαναν τον αριθμό των δέντρων (`n_estimators`), το μέγιστο βάθος των δέντρων (`max_depth`), και για το Gradient Boosting και το XGBoost, τον ρυθμό εκμάθησης (`learning_rate`). Επιπλέον, για τον XGBoost, χρησιμοποιήθηκαν οι υπερπαράμετροι υποδειγματοληψίας (`subsample`) και το ποσοστό των χαρακτηριστικών (`colsample_bytree`). Για τους αλγορίθμους Ridge και Lasso Regression, βελτιστοποιήθηκε η παράμετρος κλιμάκωσης (`alpha`). Τέλος, για τον αλγόριθμο Decision Tree Regressor, οι υπερπαράμετροι περιλάμβαναν το μέγιστο βάθος των δέντρων, τον ελάχιστο αριθμό δειγμάτων για το διαχωρισμό ενός κόμβου και τον ελάχιστο αριθμό δειγμάτων σε ένα φύλλο.

Μετά τη βελτιστοποίηση, τα καλύτερα μοντέλα αξιολογήθηκαν χρησιμοποιώντας το σύνολο δοκιμής για τον υπολογισμό του μέσου τετραγωνικού σφάλματος (MSE) και του συντελεστή προσδιορισμού (R^2). Αυτή η διαδικασία διασφαλίζει ότι τα μοντέλα έχουν

τις καλύτερες δυνατές παραμέτρους και αξιολογούνται με ακρίβεια, παρέχοντας συγκρίσιμα αποτελέσματα μεταξύ των διαφορετικών αλγορίθμων.

4.3.7 Σύγκριση Μοντέλων Πρόβλεψης

Για την σύγκριση των μοντέλων (Κώδικας 8) χρησιμοποιήθηκαν διάφορα διαγράμματα και μετρικές για να αξιολογηθούν οι επιδόσεις τους. Τα μοντέλα συγκρίθηκαν με βάση τα προβλεπόμενα αποτελέσματα και τα υπολείμματα (residuals), παρέχοντας έτσι μια ολοκληρωμένη εικόνα της απόδοσής τους.

Αρχικά, δημιουργήθηκαν διαγράμματα διασποράς (scatter plots) με γραμμές παλινδρόμησης για κάθε μοντέλο. Τα διαγράμματα αυτά δείχνουν τη σχέση μεταξύ των πραγματικών τιμών εκπομπών CO_2 και των προβλεπόμενων τιμών για κάθε μοντέλο. Η γραμμή παλινδρόμησης βοηθά στην οπτικοποίηση της ακρίβειας των προβλέψεων. Στη συνέχεια, δημιουργήθηκαν διαγράμματα υπολειμμάτων (residual plots) για κάθε μοντέλο. Τα υπολείμματα είναι οι διαφορές μεταξύ των πραγματικών και των προβλεπόμενων τιμών. Τα διαγράμματα αυτά βοηθούν στην αναγνώριση των προτύπων και των ανωμαλιών στις προβλέψεις των μοντέλων. Τέλος, δημιουργήθηκαν διαγράμματα κατανομής των υπολειμμάτων για κάθε μοντέλο. Αυτά τα διαγράμματα δείχνουν την κατανομή των σφαλμάτων και βοηθούν στην κατανόηση της απόδοσης του κάθε μοντέλου σε σχέση με την ακρίβεια των προβλέψεών τους.

4.3.8 Δημιουργία τελικού DataFrame

Σε αυτό το τελικό στάδιο της μεθοδολογίας (Κώδικας 9), δημιουργήθηκε ένα DataFrame που περιέχει τις πραγματικές τιμές εκπομπών CO_2 και τις προβλεπόμενες τιμές από τα διάφορα εκπαιδευμένα μοντέλα. Το DataFrame αυτό επιτρέπει την άμεση σύγκριση των πραγματικών και προβλεπόμενων τιμών, διευκολύνοντας την ανάλυση της απόδοσης των μοντέλων. Μετά τη δημιουργία του DataFrame, τα αποτελέσματα αποθηκεύτηκαν σε ένα αρχείο Excel με όνομα `model_predictions.xlsx`, χρησιμοποιώντας τη μέθοδο `to_excel` της `pandas`. Αυτό το αρχείο επιτρέπει την περαιτέρω ανάλυση, τεκμηρίωση και παρουσίαση των αποτελεσμάτων, διευκολύνοντας την επεξεργασία και διαχείριση των δεδομένων πρόβλεψης.

4.4 Αξιολόγηση Αποτελεσμάτων

Η ενότητα αξιολόγησης αποτελεσμάτων επικεντρώνεται στην ανάλυση και την ερμηνεία των αποτελεσμάτων που προέκυψαν από τα εκπαιδευμένα μοντέλα πρόβλεψης. Σε αυτή την ενότητα, θα εξεταστούν τα κύρια αποτελέσματα των μοντέλων, θα συγκριθούν οι επιδόσεις τους, θα αναλυθούν οι μετρικές αξιολόγησης και θα γίνει οπτική αναπαράσταση των αποτελεσμάτων για καλύτερη κατανόηση.

4.4.1 Αποτελέσματα Επιλογής Χαρακτηριστικών

Στο σημείο αυτό, θα παρουσιάσουμε τα αποτελέσματα της επιλογής χαρακτηριστικών για κάθε αλγόριθμο. Χρησιμοποιήσαμε τρεις διαφορετικές τεχνικές επιλογής χαρακτηριστικών: Mutual Information, Tree-based Feature Importance και Recursive Feature Elimination (RFE). Τα μοντέλα εκπαιδεύτηκαν και αξιολογήθηκαν χρησιμοποιώντας το

μέσο τετραγωνικό σφάλμα (MSE) και τον συντελεστή προσδιορισμού (R^2) σε ένα σύνολο επικύρωσης. Τα αποτελέσματα είναι τα εξής:

1. Linear Regression

Τεχνική Επιλογής	Training MSE	Validation MSE	Training R^2	Validation R^2
Mutual Information	0.0962	0.0848	0.9057	0.8747
Tree-based	0.0960	0.0851	0.9059	0.8743
RFE	0.9611	0.6415	0.0576	0.0524

Πίνακας 1: Αποτελέσματα εκπαίδευσης και επικύρωσης για τον αλγόριθμο Linear Regression με διάφορες τεχνικές επιλογής χαρακτηριστικών

Στην περίπτωση της γραμμικής παλινδρόμησης, οι τεχνικές Mutual Information και Tree-based Feature Importance έδωσαν πολύ παρόμοια αποτελέσματα, με υψηλά R^2 και χαμηλά MSE τόσο στο εκπαιδευτικό (training) όσο και στο επικυρωτικό (validation) σύνολο (set). Αντίθετα, η τεχνική RFE παρουσίασε σημαντικά χειρότερη απόδοση, με πολύ υψηλότερα MSE και πολύ χαμηλότερο R^2.

2. Random Forest Regressor

Τεχνική Επιλογής	Training MSE	Validation MSE	Training R^2	Validation R^2
Mutual Information	0.0045	0.0115	0.9956	0.9830
Tree-based	0.0040	0.0100	0.9960	0.9852
RFE	0.0248	0.1295	0.9756	0.8087

Πίνακας 2: Αποτελέσματα εκπαίδευσης και επικύρωσης για τον αλγόριθμο Random Forest Regressor με διάφορες τεχνικές επιλογής χαρακτηριστικών

Για τον αλγόριθμο Random Forest, η τεχνική Tree-based Feature Importance παρουσίασε την καλύτερη απόδοση, ακολουθούμενη από την τεχνική Mutual Information. Η τεχνική RFE είχε σημαντικά χαμηλότερο R^2 και υψηλότερο MSE στο επικυρωτικό σύνολο, καθιστώντας την λιγότερο κατάλληλη για αυτό το μοντέλο.

3. Gradient Boosting

Τεχνική Επιλογής	Training MSE	Validation MSE	Training R^2	Validation R^2
Mutual Information	0.0007	0.0047	0.9993	0.9931
Tree-based	0.0008	0.0046	0.9993	0.9931
RFE	0.0288	0.1415	0.9717	0.7910

Πίνακας 3: Αποτελέσματα εκπαίδευσης και επικύρωσης για τον αλγόριθμο Gradient Boosting Regressor με διάφορες τεχνικές επιλογής χαρακτηριστικών

Στην περίπτωση του Gradient Boosting, οι τεχνικές Mutual Information και Tree-based Feature Importance παρουσίασαν παρόμοια και εξαιρετική απόδοση, με πολύ χαμηλά MSE και υψηλά R^2. Η τεχνική RFE ήταν πολύ λιγότερο αποδοτική, όπως φαίνεται από το υψηλότερο MSE και το χαμηλότερο R^2 στο επικυρωτικό σύνολο.

4. XGBoost

Τεχνική Επιλογής	Training MSE	Validation MSE	Training R^2	Validation R^2
Mutual Information	0.0000	0.0150	1.0000	0.9779
Tree-based	0.0000	0.0157	1.0000	0.9768
RFE	0.0000	0.1061	1.0000	0.8432

Πίνακας 4: Αποτελέσματα εκπαίδευσης και επικύρωσης για τον αλγόριθμο XGBoost Regressor με διάφορες τεχνικές επιλογής χαρακτηριστικών

Για τον XGBoost, οι τεχνικές Mutual Information και Tree-based Feature Importance παρουσίασαν εξαιρετική απόδοση με ελάχιστα MSE και υψηλό R^2. Η τεχνική RFE δεν ήταν αποτελεσματική, παρουσιάζοντας πολύ υψηλότερο MSE και χαμηλότερο R^2.

5. Ridge Regression

Τεχνική Επιλογής	Training MSE	Validation MSE	Training R^2	Validation R^2
Mutual Information	0.0962	0.0851	0.9057	0.8743
Tree-based	0.0960	0.0853	0.9059	0.8740
RFE	0.9611	0.6414	0.0576	0.0525

Πίνακας 5: Αποτελέσματα εκπαίδευσης και επικύρωσης για τον αλγόριθμο Ridge Regression με διάφορες τεχνικές επιλογής χαρακτηριστικών

Στην περίπτωση του Ridge Regression, οι τεχνικές Mutual Information και Tree-based Feature Importance έδωσαν παρόμοια και ικανοποιητικά αποτελέσματα, ενώ η τεχνική RFE παρουσίασε σημαντικά χειρότερη απόδοση.

6. Lasso Regression

Τεχνική Επιλογής	Training MSE	Validation MSE	Training R^2	Validation R^2
Mutual Information	1.0199	0.6793	0.0000	-0.0035
Tree-based	1.0199	0.6793	0.0000	-0.0035
RFE	1.0199	0.6793	0.0000	-0.0035

Πίνακας 6: Αποτελέσματα εκπαίδευσης και επικύρωσης για τον αλγόριθμο Lasso Regression με διάφορες τεχνικές επιλογής χαρακτηριστικών

Για τον Lasso Regression, όλες οι τεχνικές επιλογής χαρακτηριστικών είχαν την ίδια απόδοση, η οποία ήταν εξαιρετικά χαμηλή, με μηδενικό R^2 στο εκπαιδευτικό σύνολο και αρνητικό R^2 στο επικυρωτικό σύνολο. Αυτό υποδηλώνει ότι ο Lasso Regression δεν ήταν κατάλληλος για αυτό το συγκεκριμένο σύνολο δεδομένων.

7. Decision Tree Regressor

Τεχνική Επιλογής	Training MSE	Validation MSE	Training R^2	Validation R^2
Mutual Information	0.0000	0.0044	1.0000	0.9935
Tree-based	0.0000	0.0045	1.0000	0.9933
RFE	0.0000	0.4878	1.0000	0.2794

Πίνακας 7: Αποτελέσματα εκπαίδευσης και επικύρωσης για τον αλγόριθμο Decision Tree Regressor με διάφορες τεχνικές επιλογής χαρακτηριστικών

Για τον Decision Tree Regressor, οι τεχνικές Mutual Information και Tree-based Feature Importance έδωσαν εξαιρετικά αποτελέσματα με μηδενικό MSE και R^2 κοντά στο 1. Η τεχνική RFE είχε πολύ χαμηλότερη απόδοση, όπως φαίνεται από το υψηλό MSE και το χαμηλό R^2 στο επικυρωτικό σύνολο.

Συμπερασματικά, τα αποτελέσματα δείχνουν ότι για τους περισσότερους αλγόριθμους, τα σύνολα χαρακτηριστικών που επιλέχθηκαν μέσω των τεχνικών Mutual Information και Tree-based Feature Importance παρουσίασαν καλύτερες επιδόσεις σε σχέση με το RFE. Αντίθετα, μέσω της τεχνικής RFE δεν παρουσιάστηκε εξίσου καλή απόδοση σε καμία περίπτωση, γεγονός που υποδεικνύει ότι δεν ήταν η βέλτιστη τεχνική επιλογής χαρακτηριστικών για τα δεδομένα και τα μοντέλα που εξετάστηκαν.

4.4.2 Αποτελέσματα Εκπαίδευσης Μοντέλων

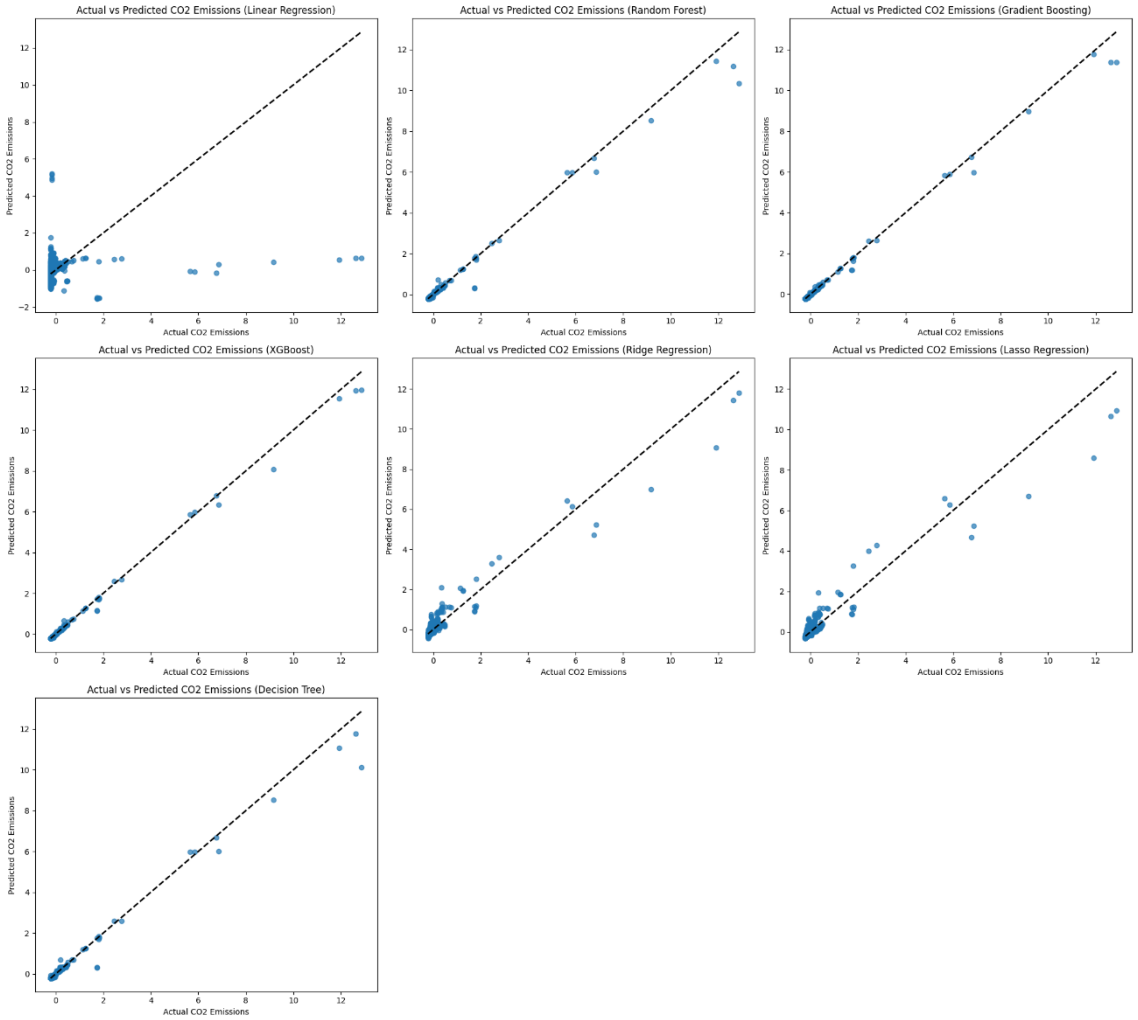
Μετά την εκπαίδευση των μοντέλων και τη χρήση των επιλεγμένων χαρακτηριστικών, τα τελικά αποτελέσματα των μοντέλων στο σύνολο δοκιμής παρουσιάζονται στον παρακάτω πίνακα:

Αλγόριθμος	Χαρακτηριστικά	Test MSE	Test R^2
Linear Regression	Tree-based	1.5480	-0.2276
Random Forest	Mutual Information	0.0241	0.9809
Gradient Boosting	Mutual Information	0.0089	0.9929
XGBoost	Mutual Information	0.0065	0.9949
Ridge Regression	Tree-based	0.0801	0.9365
Lasso Regression	Tree-based	0.0982	0.9221
Decision Tree	Mutual Information	0.0245	0.9805

Πίνακας 8: Συγκριτικός πίνακας αποτελεσμάτων εκπαίδευσης μοντέλων με τα επιλεγμένα χαρακτηριστικά, που παρουσιάζει το μέσο τετραγωνικό σφάλμα (MSE) και τον συντελεστή προσδιορισμού (R^2) για το σύνολο δοκιμής

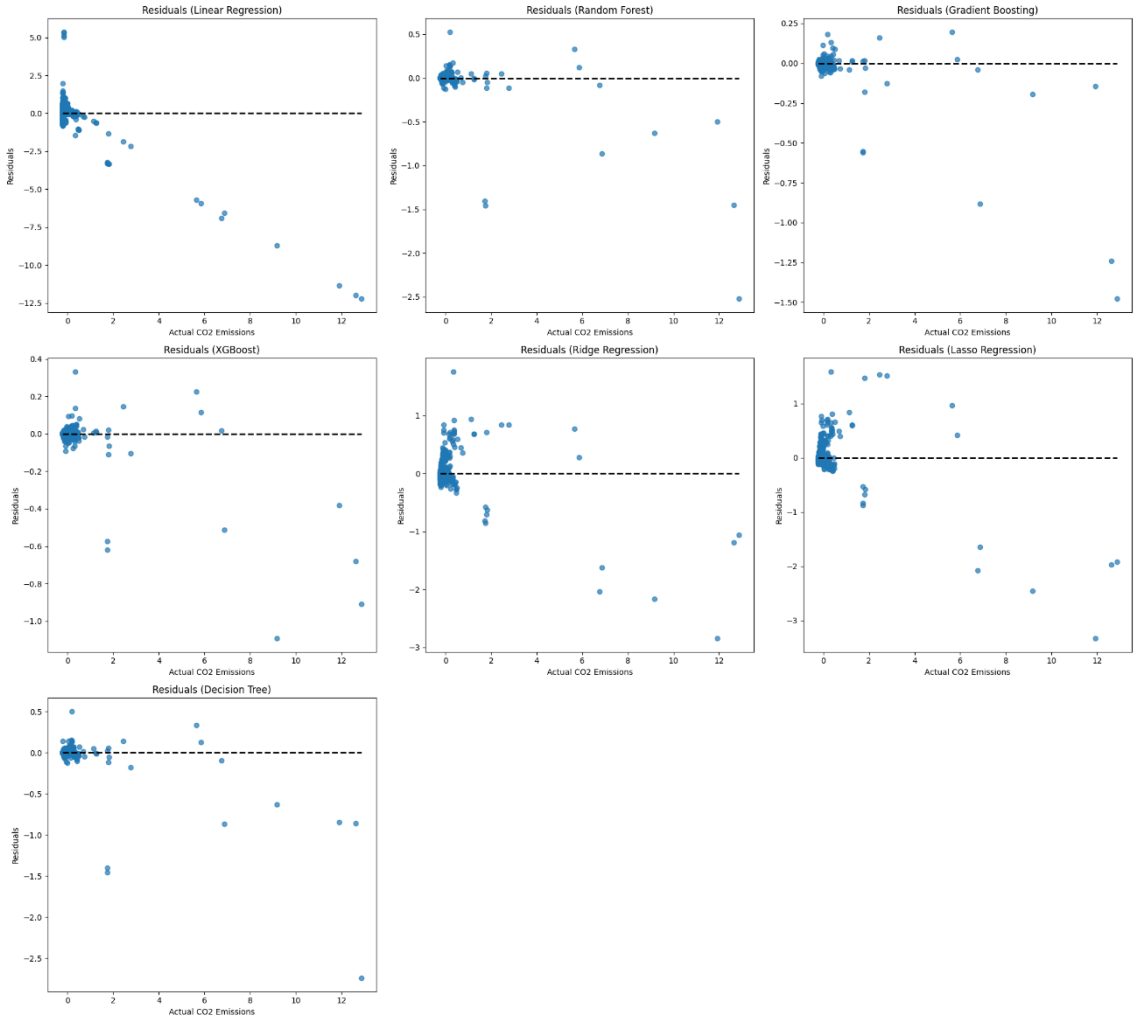
Η σύγκριση των αποτελεσμάτων δείχνει ότι το μοντέλο Linear Regression παρουσίασε τα χειρότερα αποτελέσματα με υψηλό MSE (1.5480) και αρνητικό R^2 (-0.2276), υποδηλώνοντας ότι το μοντέλο δεν ήταν κατάλληλο για την πρόβλεψη των εκπομπών CO_2 με τα επιλεγμένα χαρακτηριστικά. Αντίθετα, το μοντέλο Random Forest με χαρακτηριστικά Mutual Information είχε πολύ καλές επιδόσεις με MSE 0.0241 και R^2 0.9809, δείχνοντας υψηλή ακρίβεια και αξιοπιστία στις προβλέψεις. Το μοντέλο Gradient Boosting παρουσίασε εξαιρετικά αποτελέσματα με MSE 0.0089 και R^2 0.9929, υποδεικνύοντας ότι είναι ένα από τα πιο αποτελεσματικά μοντέλα για την πρόβλεψη των εκπομπών CO_2, ενώ το XGBoost είχε τις καλύτερες επιδόσεις με MSE 0.0065 και R^2 0.9949, καθιστώντας το πιο αποδοτικό μοντέλο από όλα τα εξεταζόμενα. Το Ridge Regression παρουσίασε καλά αποτελέσματα με MSE 0.0801 και R^2 0.9365, δείχνοντας ότι μπορεί να είναι αξιόπιστο για την πρόβλεψη των εκπομπών CO_2, αν και όχι τόσο αποτελεσματικό όσο τα μοντέλα Gradient Boosting και XGBoost. Το Lasso Regression είχε επίσης ικανοποιητικές επιδόσεις με MSE 0.0982 και R^2 0.9221, αλλά υστερεί αρκετά σε σχέση με τα άλλα μοντέλα. Τέλος, το Decision Tree παρουσίασε πολύ καλές επιδόσεις με MSE 0.0245 και R^2 0.9805, δείχνοντας ότι μπορεί να είναι μια καλή επιλογή για την πρόβλεψη των εκπομπών CO_2. Συνολικά, τα αποτελέσματα δείχνουν ότι τα μοντέλα Gradient Boosting και XGBoost με χαρακτηριστικά Mutual Information είναι τα πιο αποδοτικά για την πρόβλεψη των εκπομπών CO_2, με πολύ χαμηλά MSE και υψηλά R^2, ενώ τα μοντέλα Random Forest και Decision Tree είχαν επίσης πολύ καλές επιδόσεις. Αντίθετα, το μοντέλο Linear Regression παρουσίασε τις χειρότερες επιδόσεις, υποδεικνύοντας ότι δεν είναι κατάλληλο για αυτό το συγκεκριμένο πρόβλημα πρόβλεψης.

Για την περαιτέρω ανάλυση και την οπτική αξιολόγηση των αποτελεσμάτων, δημιουργήθηκαν διαγράμματα διασποράς με γραμμές παλινδρόμησης, διαγράμματα υπολειμμάτων και διαγράμματα κατανομής υπολειμμάτων για κάθε μοντέλο. Αυτά τα διαγράμματα βοηθούν στην καλύτερη κατανόηση της απόδοσης των μοντέλων και στην αναγνώριση τυχόν ανωμαλιών ή μοτίβων στα δεδομένα πρόβλεψης.



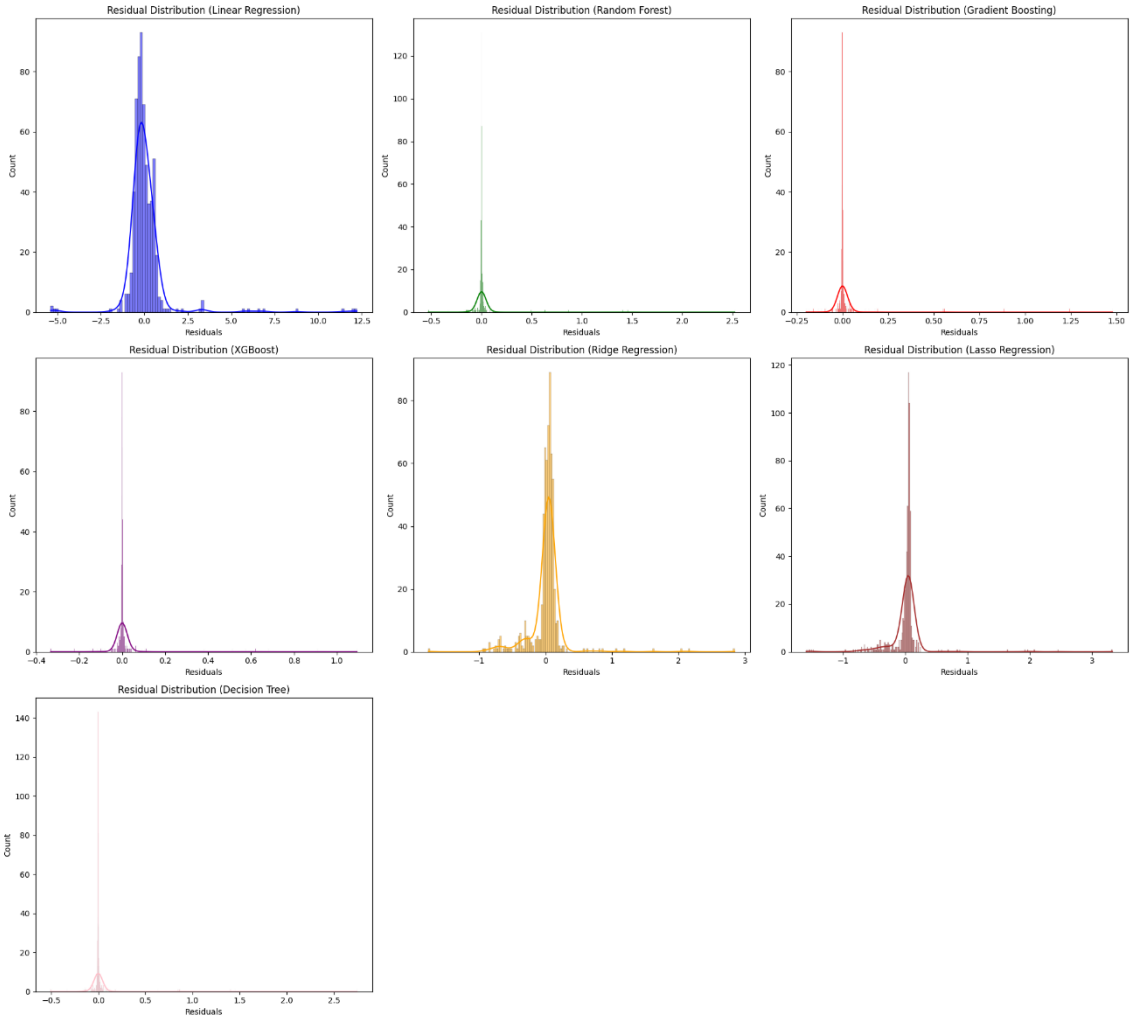
Εικόνα 20: Διαγράμματα Διασποράς με Γραμμή Παλινδρόμησης

Τα διαγράμματα διασποράς με γραμμές παλινδρόμησης δείχνουν τη σχέση μεταξύ των πραγματικών και προβλεπόμενων τιμών CO_2 για κάθε μοντέλο. Η διαγώνια γραμμή αντιπροσωπεύει την τέλεια πρόβλεψη. Το Linear Regression αποκλίνει σημαντικά από αυτή τη γραμμή, ιδιαίτερα για υψηλότερες τιμές CO_2. Τα μοντέλα Random Forest, Gradient Boosting, και XGBoost έχουν προβλέψεις που ευθυγραμμίζονται πολύ καλά με την διαγώνια γραμμή, δείχνοντας υψηλή ακρίβεια. Τα μοντέλα Ridge και Lasso Regression έχουν προβλέψεις που αποκλίνουν περισσότερο, ενώ το Decision Tree παρουσιάζει πολύ καλή προσαρμογή.



Εικόνα 21: Διαγράμματα Υπολειμμάτων

Τα διαγράμματα υπολειμμάτων δείχνουν τη διαφορά μεταξύ των πραγματικών και των προβλεπόμενων τιμών για κάθε μοντέλο. Το Linear Regression παρουσιάζει διασκορπισμένα υπολείμματα, ιδιαίτερα για υψηλότερες τιμές CO_2, υποδεικνύοντας κακή προσαρμογή του μοντέλου. Τα μοντέλα Random Forest, Gradient Boosting, και XGBoost έχουν υπολείμματα συγκεντρωμένα γύρω από το μηδέν, δείχνοντας υψηλή ακρίβεια στις προβλέψεις. Τα μοντέλα Ridge και Lasso Regression παρουσιάζουν μεγαλύτερη διασπορά στα υπολείμματα, ενώ το Decision Tree έχει παρόμοια συμπεριφορά με το Random Forest.



Εικόνα 22: Διαγράμματα Κατανομής Υπολειμμάτων

Τα διαγράμματα κατανομής υπολειμμάτων δείχνουν την κατανομή των σφαλμάτων πρόβλεψης για κάθε μοντέλο. Το Linear Regression παρουσιάζει ευρύτερη κατανομή με μακρύτερες ουρές, υποδηλώνοντας μεγαλύτερα σφάλματα πρόβλεψης. Τα μοντέλα Random Forest, Gradient Boosting, και XGBoost παρουσιάζουν στενότερες και πιο συγκεντρωμένες κατανομές, δείχνοντας ότι τα σφάλματα είναι μικρότερα και πιο προβλέψιμα. Το Ridge και Lasso Regression έχουν πιο διασκορπισμένες κατανομές, ενώ το Decision Tree παρουσιάζει στενή κατανομή, υποδεικνύοντας υψηλή ακρίβεια.

4.5 Ανάλυση Αποτελεσμάτων

Συνοψίζοντας, τα μοντέλα Gradient Boosting και XGBoost παρουσίασαν την καλύτερη απόδοση κατά την πρόβλεψη των εκπομπών CO_2, με το XGBoost να εμφανίζει το χαμηλότερο MSE (0.0065) και το υψηλότερο R^2 (0.9949), καθιστώντας το πιο αποδοτικό μοντέλο. Το Gradient Boosting επίσης είχε εξαιρετική απόδοση με MSE 0.0089 και R^2 0.9929, ενώ το Random Forest και το Decision Tree με χαρακτηριστικά Mutual Information έδειξαν υψηλή ακρίβεια με MSE 0.0241 και R^2 0.9809 και MSE 0.0245 και R^2 0.9805, αντίστοιχα.

Η υψηλή απόδοση των μοντέλων Gradient Boosting και XGBoost μπορεί να αποδοθεί στην ικανότητά τους να χειρίζονται μη γραμμικές σχέσεις και αλληλεπιδράσεις μεταξύ των χαρακτηριστικών πιο αποτελεσματικά από τα άλλα μοντέλα, με τη χρήση των χαρακτηριστικών Mutual Information να συμβάλλει σημαντικά στη βελτίωση της απόδοσης. Η μεθοδολογία που ακολουθήθηκε στην παρούσα έρευνα αποδείχθηκε αποτελεσματική για την πρόβλεψη των εκπομπών CO_2 με την προσεκτική διαδικασία προεπεξεργασίας δεδομένων, επιλογής χαρακτηριστικών και εκπαίδευσης των μοντέλων να συμβάλλουν στην ανάπτυξη μοντέλων υψηλής απόδοσης.

Ωστόσο, υπήρξαν ορισμένοι περιορισμοί και προκλήσεις. Η περιορισμένη διαθεσιμότητα δεδομένων αποτέλεσε σημαντικό πρόβλημα, καθώς οδήγησε σε ελλιπή δεδομένα, απαιτώντας επιπλέον προσεκτική προεπεξεργασία για την αντικατάσταση των missing values με κατάλληλες μεθόδους. Αυτό περιόρισε το εύρος και την ποικιλία των δεδομένων, ενδεχομένως επηρεάζοντας την ικανότητα των μοντέλων να γενικεύουν σε νέα δεδομένα. Επιπρόσθετα, η υπερπροσαρμογή εντοπίστηκε όταν τα μοντέλα παρουσίασαν πολύ χαμηλά σφάλματα (MSE) και υψηλούς συντελεστές προσδιορισμού (R^2) στα δεδομένα εκπαίδευσης, αλλά η απόδοσή τους ήταν χαμηλότερη στα δεδομένα επικύρωσης και δοκιμής.

Αναλυτικότερα, στην έρευνα παρατηρήθηκε υπερπροσαρμογή στο μοντέλο XGBoost με χαρακτηριστικά Mutual Information. Συγκεκριμένα, το μοντέλο εμφάνισε τέλειο Training R^2 (1.0000) και εξαιρετικά χαμηλό Training MSE (0.0000) στα δεδομένα εκπαίδευσης, δείχνοντας ότι το μοντέλο είχε μάθει σχεδόν τέλεια τα δεδομένα εκπαίδευσης, συμπεριλαμβανομένου του θορύβου. Ωστόσο, όταν το μοντέλο αξιολογήθηκε στα δεδομένα επικύρωσης και δοκιμής, η απόδοσή του ήταν χαμηλότερη (Validation MSE: 0.0150, Test MSE: 0.0065) και το R^2μειώθηκε (Validation R^2: 0.9779, Test R^2: 0.9949). Αυτή η διαφορά στην απόδοση μεταξύ των συνόλων δεδομένων είναι ένδειξη ότι το μοντέλο είχε υπερπροσαρμοστεί στα δεδομένα εκπαίδευσης, γεγονός που επηρεάζει την ικανότητά του να γενικεύει σε νέα, μη γνωστά δεδομένα.

Παρά τις προκλήσεις, τα ευρήματα της έρευνας υπογραμμίζουν την αξία της προσεκτικής επιλογής χαρακτηριστικών και της ρύθμισης υπερπαραμέτρων για τη βελτίωση της γενίκευσης των μοντέλων.

ΚΕΦΑΛΑΙΟ 5: Συμπεράσματα και Μελλοντικές Προκλήσεις

Η παρούσα διπλωματική εργασία εξετάζει τη χρήση αλγορίθμων μηχανικής μάθησης για την πρόβλεψη εκπομπών διοξειδίου του άνθρακα (CO_2). Μέσα από την ανάλυση, την επιλογή χαρακτηριστικών και την εκπαίδευση διαφόρων μοντέλων, καταφέραμε να αναδείξουμε την απόδοση των πιο αποδοτικών αλγορίθμων. Τα βασικά ευρήματα περιλαμβάνουν την υψηλή απόδοση των μοντέλων Gradient Boosting και XGBoost, με το XGBoost να εμφανίζει το χαμηλότερο MSE (0.0065) και το υψηλότερο R^2 (0.9949), καθιστώντας το πιο αποδοτικό μοντέλο. Επιπλέον, η προσεκτική διαδικασία προεπεξεργασίας δεδομένων, η επιλογή των κατάλληλων χαρακτηριστικών και η ρύθμιση των υπερπαραμέτρων συνέβαλαν σημαντικά στην ανάπτυξη μοντέλων υψηλής απόδοσης. Ωστόσο, οι περιορισμοί της μελέτης, όπως η περιορισμένη διαθεσιμότητα δεδομένων και η υπερπροσαρμογή, αποτέλεσαν σημαντικές προκλήσεις που επηρέασαν την απόδοση των μοντέλων.

Επιπρόσθετα, η έρευνα αυτή παρέχει σημαντικά ευρήματα και κατευθύνσεις για μελλοντική έρευνα και βελτιώσεις. Οι βασικές προτάσεις για το μέλλον περιλαμβάνουν την εφαρμογή πιο προηγμένων αλγορίθμων, όπως τα νευρωνικά δίκτυα (Deep Learning) και οι υποστηρικτικοί διανυσματικοί μηχανισμοί (Support Vector Machines), που μπορεί να βελτιώσουν την απόδοση και τη γενίκευση των προβλέψεων. Η χρήση μεγαλύτερων και πιο ποικιλόμορφων συνόλων δεδομένων από διαφορετικές γεωγραφικές περιοχές ή χρονικές περιόδους μπορεί να βελτιώσει την ακρίβεια και την αξιοπιστία των αποτελεσμάτων, ενισχύοντας τη γενίκευση των μοντέλων. Η εφαρμογή πιο εξελιγμένων μεθόδων για τη διαχείριση των missing values, την αντιμετώπιση των ανωμαλιών (outliers) και την κανονικοποίηση των δεδομένων μπορεί να συμβάλει στη βελτίωση της ποιότητας των δεδομένων. Επιπλέον, η διερεύνηση και εφαρμογή πιο αποτελεσματικών μεθόδων επιλογής χαρακτηριστικών μπορεί να βοηθήσει στην κατανόηση των σημαντικότερων παραγόντων που επηρεάζουν τις εκπομπές CO_2, οδηγώντας σε πιο ακριβείς προβλέψεις. Η χρήση τεχνικών κανονικοποίησης και άλλων στρατηγικών για τη βελτίωση της γενίκευσης των μοντέλων μπορεί να μειώσει τα ποσοστά υπερπροσαρμογής και να βελτιώσει την απόδοση των προβλέψεων.

Τέλος, τα ευρήματα της παρούσας έρευνας μπορούν να έχουν σημαντικές εφαρμογές στην ανάπτυξη πολιτικών και στρατηγικών για τη μείωση των εκπομπών CO_2. Η εφαρμογή των προβλεπτικών μοντέλων μπορεί να βοηθήσει τους υπεύθυνους χάραξης πολιτικής να λάβουν πιο ενημερωμένες αποφάσεις για την αντιμετώπιση της κλιματικής αλλαγής. Ακόμα, η συνεχιζόμενη βελτίωση και εξέλιξη των μεθόδων πρόβλεψης μπορεί να συμβάλει στη δημιουργία ενός πιο βιώσιμου μέλλοντος. Αυτές οι προτάσεις και τα συμπεράσματα υπογραμμίζουν τη σημασία της συνεχιζόμενης έρευνας και ανάπτυξης στον τομέα της πρόβλεψης εκπομπών CO_2, προσφέροντας νέες ευκαιρίες για τη βελτίωση της περιβαλλοντικής διαχείρισης και τη μείωση των επιπτώσεων της κλιματικής αλλαγής.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] T. Adrian, N. Boyarchenko, D. Giannone, A. Prasad, D. Seneviratne και Y. Xiao, «800,000 Years of Climate Risk,» *SSRN Electronic Journal*, p. 65, 09 09 2022.

[2] IPCC, Global Warming of 1.5°C: IPCC Special Report on Impacts of Global Warming of 1.5°C above Pre-industrial Levels in Context of Strengthening Response to Climate Change, Sustainable Development, and Efforts to Eradicate Poverty, Cambridge University Press, 2022.

[3] USGCRP, D. R. Reidmiller, C. W. Avery, D. R. Easterling, K. E. Kunkel, K. L. Lewis, T. K. Maycock και B. C. Stewart, «Impacts, Risks, and Adaptation in the United States: The Fourth National Climate Assessment, Volume II,» U.S. Government Publishing Office, Washington, 2018.

[4] «Amazon Fire Dashboard,» [Ηλεκτρονικό]. Available: https://amzfire.servirglobal.net/dashboard/. [Πρόσβαση 04 04 2024].

[5] NASA, «Global Temperature,» 2023. [Ηλεκτρονικό]. Available: https://climate.nasa.gov/vital-signs/global-temperature/. [Πρόσβαση 04 04 2024].

[6] EEA, «European Climate Risk Assessment,» EU publications, 2024.

[7] IUCN, «Species and climate change,» IUCN, 2019. [Ηλεκτρονικό]. Available: https://www.iucn.org/resources/issues-brief/species-and-climate-change. [Πρόσβαση 10 04 2024].

[8] G. Fulton, «The Bramble Cay melomys: the first mammalian extinction due to human-induced climate change,» *Pacific Conservation Biology*, pp. 1-3, 01 01 2017.

[9] Field, C.B., V.R. Barros, D.J. Dokken, K.J. Mach, M.D. Mastrandrea, T.E. Bilir, M. Chatterjee, K.L. Ebi, Y.O. Estrada, R.C. Genova, B. Girma, E.S. Kissel, A.N. Levy, S. MacCracken, P.R. Mastrandrea, and L.L. White (eds.), «Summary for policymakers,» σε *Climate Change 2014: Impacts, Adaptation, and Vulnerability. Part A: Global and Sectoral Aspects*, Cambridge and New York, NY, Cambridge University Press, 2014, pp. 14-20.

[10] UCAR | Center for Science Education, «The Greenhouse Effect».

[11] R. K. Pachauri και L. Mayer, Climate change 2014: synthesis report, Intergovernmental Panel on Climate Change, 2015, pp. 44-46.

[12] M. Maslin, «Forty years of linking orbits to ice ages,» *Nature*, τόμ. 540, αρ. 7632, pp. 208-209, 7 12 2016.

[13] NASA Science, «Graphic: The relentless rise of carbon dioxide,» [Ηλεκτρονικό]. Available: https://science.nasa.gov/resource/graphic-the-relentless-rise-of-carbon-dioxide/. [Πρόσβαση 10 04 2024].

[14] Y. Meng και H. Noman, «Predicting CO2 Emission Footprint Using AI through Machine Learning,» *Atmosphere*, τόμ. 13, αρ. 11, p. 1871, 11 2022.

[15] Z. Ji, H. Song, L. Lei, M. Sheng, K. Guo και S. Zhang, «A Novel Approach for Predicting Anthropogenic CO2 Emissions Using Machine Learning Based on Clustering of the CO2 Concentration,» *Atmosphere*, τόμ. 15, αρ. 3, p. 323, 03 2024.

[16] S. Li, Y. W. Siu και G. Zhao, «Driving Factors of CO2 Emissions: Further Study Based on Machine Learning,» *Frontiers in Environmental Science*, τόμ. 9, 23 08 2021.

[17] Y. Natarajan, G. Wadhwa, K. R. Sri Preethaa και A. Paul, «Forecasting Carbon Dioxide Emissions of Light-Duty Vehicles with Different Machine Learning Algorithms,» *Electronics*, τόμ. 12, αρ. 10, p. 2288, 18 05 2023.

[18] S. Kumari και S. K. Singh, «Machine learning-based time series models for effective CO2 emission prediction in India,» *Environmental Science and Pollution Research*, τόμ. 30, αρ. 55, pp. 116601-116616, 01 11 2023.

[19] C. Wang, M. Li και J. Yan, «Forecasting carbon dioxide emissions: application of a novel two-stage procedure based on machine learning models,» *Journal of Water and Climate Change*, τόμ. 14, αρ. 2, pp. 477-493, 01 02 2023.

[20] T. M. Mitchell, Machine Learning, New York: McGraw-Hill Education, 1997, p. 432.

[21] A. Burkov, The Hundred-Page Machine Learning Book, Andriy Burkov, 2019.

[22] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms, CRC Press, 2012.

[23] J. Bergstra και Y. Bengio, «Random search for hyper-parameter optimization,» *Journal of Machine Learning Research,* τόμ. 13, pp. 281-305, 2012.

[24] M. B. Christopher, Pattern Recognition and Machine Learning, Springer, 2006.

[25] D. N. Gujarati και D. C. Porter, Basic Econometrics, McGraw-Hill Irwin, 2009, pp. 321-323.

[26] Britannica και K. Stewart, «Linear Regression,» 2024. [Ηλεκτρονικό]. Available: https://www.britannica.com/topic/linear-regression. [Πρόσβαση 09 05 2024].

[27] G. James, D. Witten, T. Hastie και R. Tibshirani, An Introduction to Statistical Learning, Springer New York, NY, 2022.

[28] Javatpoint, «Linear Regression in Machine Learning,» [Ηλεκτρονικό]. Available: https://www.javatpoint.com/linear-regression-in-machine-learning. [Πρόσβαση 15 05 2024].

[29] S. M. Stigler, The History of Statistics: The Measurement of Uncertainty Before 1900, Harvard University Press, 1989.

[30] N. R. Draper και H. Smith, Applied Regression Analysis, John Wiley & Sons, 1998, p. 736.

[31] J. H. Friedman, «Greedy function approximation: A gradient boosting machine.,» *The Annals of Statistics,* τόμ. 29, αρ. 5, pp. 1189-1232, 10 2001.

[32] A. Natekin και A. Knoll, «Gradient boosting machines, a tutorial,» *Front Neurorobot,* τόμ. 7, αρ. 21, 4 12 2013.

[33] A. E. Hoerl και R. W. Kennard, «Ridge Regression: Biased Estimation for Nonorthogonal Problems,» *Technometrics,* τόμ. 12, αρ. 1, pp. 55-67, 1970.

[34] R. Tibshirani, «Regression Shrinkage and Selection via the Lasso,» *Journal of the Royal Statistical Society. Series B (Methodological),* τόμ. 58, αρ. 1, pp. 267-288, 1996.

[35] T. Hastie, R. Tibshirani και J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, NY: Springer New York, 2013, p. 536.

[36] T. H. Chen, M. Benesty, V. Khotilovich και Y. Tang, «Xgboost: extreme gradient boosting,» *R package version 0.4-2,* τόμ. 1, αρ. 4, pp. 1-4, 2015.

[37] T. Chen και C. Guestrin, «XGBoost: A Scalable Tree Boosting System,» *In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* pp. 785-794, 2016.

[38] J. Quinlan, «Induction of Decision Trees,» *Machine Learning,* τόμ. 1, αρ. 1, pp. 81-106, 1986.

[39] L. Breiman, J. Friedman, R. Olshen και C. Stone, Classification and Regression Trees, Wadsworth & Brooks, 1984.

[40] R. Battiti, «Using mutual information for selecting features in supervised neural net learning,» *IEEE Transactions on Neural Networks,* τόμ. 5, αρ. 4, pp. 537-550, 1994.

[41] J. Vergara και P. Estévez, «A review of feature selection methods based on mutual information,» *Neural Computing and Applications,* τόμ. 24, αρ. 1, pp. 175-186, 2014.

[42] M. Kuhn και K. Johnson, Applied Predictive Modeling, New York, NY: Springer New York, 2013.

[43] G. Chandrashekar και F. Sahin, «A survey on feature selection methods,» *Computers & Electrical Engineering,* τόμ. 40, αρ. 1, pp. 16-28, 2014.

[44] G. Louppe, Understanding Random Forests: From Theory to Practice, 2014.

ΠΑΡΑΡΤΗΜΑ: Τελικός Κώδικας

Στο παράρτημα αυτό παρατίθεται ο πλήρης κώδικας που χρησιμοποιήθηκε για την ανάλυση των δεδομένων και την εκπαίδευση των μοντέλων μηχανικής μάθησης. Ο κώδικας περιλαμβάνει όλα τα στάδια της διαδικασίας, από την προεπεξεργασία των δεδομένων μέχρι την τελική αξιολόγηση των μοντέλων. Κάθε τμήμα του κώδικα είναι δομημένο σύμφωνα με τις ενότητες που αναλύονται στο κύριο σώμα της εργασίας, επιτρέποντας έτσι την αναπαραγωγή των αποτελεσμάτων και την περαιτέρω ανάλυση.

Κώδικας 1: Εισαγωγή Βιβλιοθηκών

```
# Libraries
import pandas as pd;
import numpy as np;
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt
import seaborn as sns
import pycountry;
from sklearn.model_selection import train_test_split
import missingno as msno
from sklearn.feature_selection import SelectKBest, mutual_info_regres-
sion
from sklearn.ensemble import RandomForestRegressor
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score
import matplotlib.pyplot as plt
from sklearn.linear_model import Ridge, Lasso
from sklearn.tree import DecisionTreeRegressor
import xgboost as xgb
```

Κώδικας 2: Φόρτωση και Αρχική Επισκόπηση του Συνόλου Δεδομένων

```
# Load the dataset
csv_file_path = 'C:\\Users\\30698\\Documents\\Μαθήματα\\Διπλωματική
Εργασία\\DataSet\\newMasterThesisDataSet.csv'
data = pd.read_csv(csv_file_path)

# Display basic information about the dataset
```

```
data.info() # Shows a summary of the DataFrame including the number of
non-null entries and data types
data.columns # Lists all column names
data.shape # Displays the dimensions of the DataFrame (number of rows
and columns)

# Print some random rows to inspect the dataset before processing
print("The first ten rows of the dataset")
display(data.head(10))

print("The last ten rows of the dataset")
display(data.tail(10))

print("Displaying the data types of each column")
print(data.dtypes)
```

Κώδικας 3: Προεπεξεργασία Δεδομένων

```
# Remove columns that are not useful
data.drop('Time Code', axis=1, inplace=True) # Remove the 'Time Code'
column as it is not useful for the analysis
data.drop('Country Code', axis=1, inplace=True) # Remove the 'Country
Code' column as it is not useful for the analysis

# Rename the columns to make the names more understandable
data.rename(columns=lambda x: x.split(' [')[0], inplace=True)
data

# Check the values of the Country Name column because some odd data
were observed in the csv preview

countriesBeforeEdit = data['Country Name'].unique()
print(countriesBeforeEdit)
len(countriesBeforeEdit)

# Create a function to exclude rows that do not match to a real coun-
try
def is_valid_country(name):
    if pd.isna(name):
        return False
    try:
        #if the country name is valid
        pycountry.countries.lookup(name)
        return True
```

```
    except LookupError:
        #if the country is not found, return False
        return False

# Apply the function to filter the dataframe to include only valid
countries
data = data[data['Country Name'].apply(is_valid_country)]

# Get unique country names after filtering
countriesAfterEdit = data['Country Name'].unique()
print(countriesAfterEdit)
len(countriesAfterEdit)

# Replace the values ".." and "0" with NaN to facilitate the pro-
cessing of missing data
data.replace('..', pd.NA, inplace=True)
data.replace('0', pd.NA, inplace=True)

# Calculate the number of missing values per year and line and sort
the years by the number of missing values
mean_missing_per_year = data.groupby('Time').apply(lambda x:
x.isnull().mean().mean()).sort_values(ascending=False)

# Display the list of years by percentage of missing values per column
mean_missing_per_year

# Graphic representation
plt.figure(figsize=(40,7))
mean_missing_per_year.plot(kind='bar', color='skyblue')
plt.title('Missing Values per Year')
plt.xlabel('Year')
plt.ylabel('Average Missing Values')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()

# Set the threshold for the acceptable percentage of missing values
per line and year
row_absent_limit = 0.35 #35%

# Search for values that exceed the specified limit
rows_to_exclude = mean_missing_per_year[ mean_missing_per_year >
row_absent_limit].index.tolist()

# Display the years to be excluded
```

```
print("Years to be excluded:", rows_to_exclude)

# Remove the above values from the dataframe
cleaned_data = data[~data['Time'].isin(rows_to_exclude)]

# Calculate the missing values per column
column_missing_values = cleaned_data.isnull().mean().sort_values(as-
cending=False)

# Display the number of missing values for each column
column_missing_values

# Graphic representation
plt.figure(figsize=(40,7))
column_missing_values.plot(kind='bar', color='purple')
plt.title('Missing Values per Column')
plt.xlabel('Column')
plt.ylabel('Average Missing Values')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()

# Set the threshold for the acceptable percentage of missing values
per column
column_absent_limit = 0.35 #47%

# Search for values that exceed the specified limit
columns_to_exclude = column_missing_values[column_missing_values >
column_absent_limit].index.tolist()

# Display the columns to be excluded
print("Columns to be excluded:", columns_to_exclude)


# Remove the above values from the dataframe
cleaned_data = cleaned_data.drop(columns=columns_to_exclude)

# Display the remaining columns of the dataset
remaining_columns = cleaned_data.columns.tolist()
remaining_columns

# Calculate the missing values per country
# Calculate the mean missing values for each country
mean_missing_per_country = cleaned_data.groupby('Country Name').ap-
ply(lambda x: x.isnull().mean().mean()).sort_values(ascending=False)
```

```
# Display the list of years by percentage of missing values per column
mean_missing_per_country

# Graphic representation
plt.figure(figsize=(40,7))
mean_missing_per_country.plot(kind='bar', color='skyblue')
plt.title('Missing Values per Country')
plt.xlabel('Country')
plt.ylabel('Average Missing Values')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()

# Set the threshold for the acceptable percentage of missing values
per country
# Set a threshold for acceptable missing values
country_absent_limit = 0.35  #35%

# Identify countries that exceed the missing value limit
countries_to_exclude = mean_missing_per_country[mean_missing_per_coun-
try > country_absent_limit].index.tolist()

# Display the countries to be excluded
print("Countries to be excluded:", countries_to_exclude)

# Remove the identified countries from the dataframe
cleaned_data = cleaned_data[~cleaned_data['Country Name'].isin(coun-
tries_to_exclude)]

#Graphically analyse the missingness of the data
# Visualize missing values as a matrix for the cleaned_data DataFrame
# Group the data by 'Time' and calculate the mean nullity for each
column.
mean_missing_per_column = cleaned_data.groupby('Time').apply(lambda x:
x.isnull().mean())

# Visualize missing values using the cleaned_data DataFrame but sorted
by the average nullity of columns in descending order of missing data
sorted_columns = mean_missing_per_column.mean().sort_values(ascend-
ing=False).index
sorted_data = cleaned_data[sorted_columns]
print(mean_missing_per_column.columns)

# Visualize the number of missing values as a bar chart
```

```
msno.bar(sorted_data)

# Convert all columns that should be numeric but are 'object' type to
float
numeric_columns = cleaned_data.columns.drop(['Time', 'Country
Name'])  # Excluding non-numeric columns
cleaned_data[numeric_columns] = cleaned_data[numeric_columns].ap-
ply(pd.to_numeric, errors='coerce')
cleaned_data.info()

# Custom Imputation
# Replace pd.NA with np.nan to avoid errors
copy_cleaned_data = cleaned_data.fillna(np.nan)

# Group the data by 'Country Name' and fill missing values with the
mean of each column
copy_cleaned_data = cleaned_data.groupby(cleaned_data['Country
Name']).transform(lambda x: x.fillna(x.mean()))

# Insert 'Country Name' back at the first position
copy_cleaned_data.insert(0, 'Country Name', cleaned_data['Country
Name'])

# Display the resulting DataFrame
display(copy_cleaned_data)

# Check if any null values remain
if copy_cleaned_data.isnull().any().any():
    print("Some NaNs remain after imputation and row removal, requir-
ing further handling.")
else:
    print("All rows with completely empty columns have been success-
fully removed.")

# Visualization after imputation
# Visualize missing values as a matrix for the cleaned_data DataFrame
# Group the data by 'Time' and calculate the mean nullity for each
column.
mean_missing_per_column = copy_cleaned_data.groupby('Time').ap-
ply(lambda x: x.isnull().mean())

# Visualize missing values using the cleaned_data DataFrame but sorted
by the average nullity of columns in descending order of missing data
sorted_columns = mean_missing_per_column.mean().sort_values(ascend-
ing=False).index
```

```
sorted_data = copy_cleaned_data[sorted_columns]
print(mean_missing_per_column.columns)

# Plot the missing values as a bar chart
msno.bar(sorted_data)

# Calculate Missing Values Per Country and Column
# Calculate the percentage of NaNs for each column in each country
country_column_nan = copy_cleaned_data.groupby('Country Name').ap-
ply(lambda x: x.isnull().mean())

# Filter to find columns where the percentage of NaNs is 1 (100% miss-
ing)
completely_missing = country_column_nan[country_column_nan ==
1.0].dropna(how='all', axis=1)

print("Columns with 100% missing data per country:")
print(completely_missing)

# Identify the rows that have at least one column completely empty
rows_to_delete = copy_cleaned_data.apply(lambda x: any(x[com-
pletely_missing.columns].isnull()), axis=1)

# Remove these rows from the dataset
cleaned_data_final = copy_cleaned_data[~rows_to_delete]

# Check again for remaining NaNs to ensure complete cleaning
if cleaned_data_final.isnull().any().any():
    print("Some NaNs remain after row removal, requiring further han-
dling.")
else:
    print("All rows with completely empty columns have been success-
fully removed.")

# Display the resulting DataFrame
display(cleaned_data_final)

# Save copy_cleaned_data into an excel file
cleaned_data_final.to_excel('cleaned_data_final.xlsx',
sheet_name='cleaned_data_final', index=False)
```

Κώδικας 4: Επιλογή Χαρακτηριστικών και Κλιμάκωση

```
# Defining the features and target variables
X = cleaned_data_final.drop('CO2 emissions (kt)', axis=1)
y = cleaned_data_final['CO2 emissions (kt)']

# Standarize the target variable
scaler = StandardScaler()
y_scaled = scaler.fit_transform(y.values.reshape(-1, 1)).flatten()


# Label encode 'Country Name'
label_encoder = LabelEncoder()
X['Country Name'] = label_encoder.fit_transform(X['Country Name'])

# 2. Split the original dataset into training, validation, and test
sets
# Split the data into training and temporary sets
X_train, X_temp, y_train, y_temp = train_test_split(X, y_scaled,
test_size=0.4, random_state=42)

# Split the temporary set into validation and test sets
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp,
test_size=0.5, random_state=42)

# 3. Select features using different methods and trasform the train-
ing, validation and test sets using the same scaling parameters

# Method 1: SelectKBest with mutual_info_regression
mi_selector = SelectKBest(mutual_info_regression, k=10)
X_train_mi_selected = mi_selector.fit_transform(X_train, y_train)
X_val_mi_selected = mi_selector.transform(X_val)
X_test_mi_selected = mi_selector.transform(X_test)
mi_selected_features = X_train.columns[mi_selector.get_support()]

# Method 2: Tree-based feature importance
rf = RandomForestRegressor(random_state=42)
rf.fit(X_train, y_train)
feature_importances = rf.feature_importances_
importances_df = pd.DataFrame({'Feature': X_train.columns, 'Im-
portance': feature_importances})
importances_df = importances_df.sort_values(by='Importance', ascend-
ing=False).head(10)
tree_selected_features = importances_df['Feature'].values

X_train_tree_selected = X_train[tree_selected_features]
X_val_tree_selected = X_val[tree_selected_features]
```

```
X_test_tree_selected = X_test[tree_selected_features]

# Method 3: Recursive Feature Elimination (RFE)
model = LinearRegression()
rfe_selector = RFE(model, n_features_to_select=10, step=1)
rfe_selector = rfe_selector.fit(X_train, y_train)
rfe_selected_features = X_train.columns[rfe_selector.support_]

X_train_rfe_selected = rfe_selector.transform(X_train)
X_val_rfe_selected = rfe_selector.transform(X_val)
X_test_rfe_selected = rfe_selector.transform(X_test)
# 4. Scale the selected features
scaler = StandardScaler()

# Mutual Information selected features
X_mi_scaled = scaler.fit_transform(X[mi_selected_features])
X_train_mi_scaled = scaler.fit_transform(X_train_mi_selected)
X_val_mi_scaled = scaler.transform(X_val_mi_selected)
X_test_mi_scaled = scaler.transform(X_test_mi_selected)

# Tree-based selected features
X_tree_scaled = scaler.fit_transform(X[tree_selected_features])
X_train_tree_scaled = scaler.fit_transform(X_train_tree_selected)
X_val_tree_scaled = scaler.transform(X_val_tree_selected)
X_test_tree_scaled = scaler.transform(X_test_tree_selected)

# RFE selected features
X_rfe_scaled = scaler.fit_transform(X[rfe_selected_features])
X_train_rfe_scaled = scaler.fit_transform(X_train_rfe_selected)
X_val_rfe_scaled = scaler.transform(X_val_rfe_selected)
X_test_rfe_scaled = scaler.transform(X_test_rfe_selected)

# Output the selected features
print("\nSelected Features (Mutual Information):\n", mi_selected_fea-
tures)
print("\nSelected Features (Tree-based):\n", tree_selected_features)
print("\nSelected Features (RFE):\n", rfe_selected_features)
```

Κώδικας 5: Εκπαίδευση Μοντέλων και Συναρτήσεις Αξιολόγησης

```
# 1. Initialize the models
lr = LinearRegression()
rf = RandomForestRegressor(random_state=40)
```

```
gb = GradientBoostingRegressor(random_state=40)
xgboost = xgb.XGBRegressor(random_state=40)
ridge = Ridge(random_state=40)
lasso = Lasso(random_state=40)
dt = DecisionTreeRegressor(random_state=40)

# Print initialized models
print("Models initialized:")
print("Linear Regression:", lr)
print("Random Forest Regressor:", rf)
print("Gradient Boosting Regressor:", gb)
print("XGBoost Regressor:", xgboost)
print("Ridge Regression:", ridge)
print("Lasso Regression:", lasso)
print("Decision Tree Regressor:", dt)

# 2. Create an Evaluation Function
def evaluate_model(model, X_train, y_train, X_val, y_val):
    model.fit(X_train, y_train)
    y_train_pred = model.predict(X_train)
    y_val_pred = model.predict(X_val)

    train_mse = mean_squared_error(y_train, y_train_pred)
    val_mse = mean_squared_error(y_val, y_val_pred)
    train_r2 = r2_score(y_train, y_train_pred)
    val_r2 = r2_score(y_val, y_val_pred)

    print(f"Training MSE: {train_mse:.4f}")
    print(f"Validation MSE: {val_mse:.4f}")
    print(f"Training R2: {train_r2:.4f}")
    print(f"Validation R2: {val_r2:.4f}")
```

Κώδικας 6: Εκπαίδευση και Αρχική Αξιολόγηση Μοντέλων

```
# 1. Linear Regression
# Mutual Information Selected Feautures
print("\nLinear Regression (Mutual Information):")
evaluate_model(lr, X_train_mi_scaled, y_train, X_val_mi_scaled, y_val)

# Tree Based Selected Feautured
print("\nLinear Regression (Tree-based):")
evaluate_model(lr, X_train_tree_scaled, y_train, X_val_tree_scaled,
y_val)
```

```
# RFE Selected Features
print("\nLinear Regression (RFE):")
evaluate_model(lr, X_train_rfe_scaled, y_train, X_val_rfe_scaled,
y_val)

# 2. Random Forest Regressor
# Mutual Information Selected Feautures
print("\nRandom Forest (Mutual Information):")
evaluate_model(rf, X_train_mi_scaled, y_train, X_val_mi_scaled, y_val)

#Tree Based Selected Feautured
print("\nRandom Forest (Tree-based):")
evaluate_model(rf, X_train_tree_scaled, y_train, X_val_tree_scaled,
y_val)

# RFE Selected Features
print("\nRandom Forest (RFE):")
evaluate_model(rf, X_train_rfe_scaled, y_train, X_val_rfe_scaled,
y_val)

# 3. Gradient Boosting Regressor
# Mutual Information Selected Feautures
print("\nGradient Boosting (Mutual Information):")
evaluate_model(gb, X_train_mi_scaled, y_train, X_val_mi_scaled, y_val)

# Tree Based Selected Feautured
print("\nGradient Boosting (Tree-based):")
evaluate_model(gb, X_train_tree_scaled, y_train, X_val_tree_scaled,
y_val)

# RFE Selected Features
print("\nGradient Boosting (RFE):")
evaluate_model(gb, X_train_rfe_scaled, y_train, X_val_rfe_scaled,
y_val)

# 4. XGBoost
# Mutual Information Selected Feautures
print("\nXGBoost (Mutual Information)")
evaluate_model(xgboost, X_train_mi_scaled, y_train, X_val_mi_scaled,
y_val)

# Tree Based Selected Feautured
print("\nXGBoost (Tree-based)")
```

```
evaluate_model(xgboost, X_train_tree_scaled, y_train,
X_val_tree_scaled, y_val)

# RFE Selected Features
print("\nXGBoost (RFE)")
evaluate_model(xgboost, X_train_rfe_scaled, y_train, X_val_rfe_scaled,
y_val)

#  5. Ridge Regression
# Mutual Information Selected Feautures
ridge = Ridge()
print("\nRidge Regression (Mutual Information):")
evaluate_model(ridge, X_train_mi_scaled, y_train, X_val_mi_scaled,
y_val)

# Tree Based Selected Feautured
print("\nRidge Regression (Tree-based):")
evaluate_model(ridge, X_train_tree_scaled, y_train, X_val_tree_scaled,
y_val)

# RFE Selected Features
print("\nRidge Regression (RFE):")
evaluate_model(ridge, X_train_rfe_scaled, y_train, X_val_rfe_scaled,
y_val)

# 6. Lasso Regression
# Mutual Information Selected Feautures

print("\nLasso Regression (Mutual Information):")
evaluate_model(lasso, X_train_mi_scaled, y_train, X_val_mi_scaled,
y_val)

# Tree Based Selected Feautured
print("\nLasso Regression (Tree-based):")
evaluate_model(lasso, X_train_tree_scaled, y_train, X_val_tree_scaled,
y_val)

# RFE Selected Features
print("\nLasso Regression (RFE):")
evaluate_model(lasso, X_train_rfe_scaled, y_train, X_val_rfe_scaled,
y_val)

# 7. Decision Tree Regressor
# Mutual Information Selected Feautures
print("\nDecision Tree Regressor (Mutual Information):")
```

```
evaluate_model(dt, X_train_mi_scaled, y_train, X_val_mi_scaled, y_val)

# Tree Based Selected Feautured
print("\nDecision Tree Regressor (Tree-based):")
evaluate_model(dt, X_train_tree_scaled, y_train, X_val_tree_scaled,
y_val)

# RFE Selected Features
print("\nDecision Tree Regressor (RFE):")
evaluate_model(dt, X_train_rfe_scaled, y_train, X_val_rfe_scaled,
y_val)
```

Κώδικας 7: Εκπαίδευση των Μοντέλων

```
# 1. Linear Regression with Tree-based Features
# Test the model on the test set
y_test_pred_lr = lr.predict(X_test_tree_scaled)
test_mse_lr = mean_squared_error(y_test, y_test_pred_lr)
test_r2_lr = r2_score(y_test, y_test_pred_lr)

print(f"Test MSE: {test_mse_lr:.4f}")
print(f"Test R2: {test_r2_lr:.4f}")

# 2. Random Forest with Mutual Information Features
# Define the parameter grid for Random Forest
param_grid_rf = {
    'n_estimators': [100, 200, 300],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'bootstrap': [True, False]
}

# Initialize GridSearchCV for Random Forest
grid_search_rf = GridSearchCV(estimator=rf, param_grid=param_grid_rf,
cv=3, scoring='r2', n_jobs=-1, verbose=2)

# Fit the model
grid_search_rf.fit(X_train_mi_scaled, y_train)

# Get the best parameters and best model
best_params_rf = grid_search_rf.best_params_
best_model_rf = grid_search_rf.best_estimator_
```

```
print(f"Best Parameters for Random Forest: {best_params_rf}")

# Test the best model on the test set
y_test_pred_rf = best_model_rf.predict(X_test_mi_scaled)
test_mse_rf = mean_squared_error(y_test, y_test_pred_rf)
test_r2_rf = r2_score(y_test, y_test_pred_rf)

print(f"Test MSE: {test_mse_rf:.4f}")
print(f"Test R2: {test_r2_rf:.4f}")


# 3. Gradient Boosting with Mutual Information Features
# Define the parameter grid for Gradient Boosting
param_grid_gb = {
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 4, 5]
}



# Initialize GridSearchCV for Gradient Boosting
grid_search_gb = GridSearchCV(estimator=gb, param_grid=param_grid_gb,
cv=3, scoring='r2', n_jobs=-1, verbose=2)

# Fit the model
grid_search_gb.fit(X_train_mi_scaled, y_train)

# Get the best parameters and best model
best_params_gb = grid_search_gb.best_params_
best_model_gb = grid_search_gb.best_estimator_

print(f"Best Parameters for Gradient Boosting: {best_params_gb}")

# Test the best model on the test set
y_test_pred_gb = best_model_gb.predict(X_test_mi_scaled)
test_mse_gb = mean_squared_error(y_test, y_test_pred_gb)
test_r2_gb = r2_score(y_test, y_test_pred_gb)

print(f"Test MSE: {test_mse_gb:.4f}")
print(f"Test R2: {test_r2_gb:.4f}")

# 4. XGBoost with Mutual Information Features
# Define the parameter grid for XGBoost
param_grid_xgb = {
```

```
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 4, 5],
    'subsample': [0.8, 0.9, 1.0],
    'colsample_bytree': [0.8, 0.9, 1.0]
}

# Initialize GridSearchCV for XGBoost
grid_search_xgb = GridSearchCV(estimator=xgboost,
param_grid=param_grid_xgb, cv=3, scoring='r2', n_jobs=-1, verbose=2)

# Fit the model
grid_search_xgb.fit(X_train_mi_scaled, y_train)

# Get the best parameters and best model
best_params_xgb = grid_search_xgb.best_params_
best_model_xgb = grid_search_xgb.best_estimator_

print(f"Best Parameters for XGBoost: {best_params_xgb}")

# Test the best model on the test set
y_test_pred_xgb = best_model_xgb.predict(X_test_mi_scaled)
test_mse_xgb = mean_squared_error(y_test, y_test_pred_xgb)
test_r2_xgb = r2_score(y_test, y_test_pred_xgb)

print(f"Test MSE: {test_mse_xgb:.4f}")
print(f"Test R2: {test_r2_xgb:.4f}")

# 5. Ridge Regression with Tree-based Feature
# Define the parameter grid for Ridge Regression
param_grid_ridge = {
    'alpha': [0.1, 1.0, 10.0]
}

# Initialize GridSearchCV for Ridge Regression
grid_search_ridge = GridSearchCV(estimator=ridge,
param_grid=param_grid_ridge, cv=3, scoring='r2', n_jobs=-1, verbose=2)

# Fit the model
grid_search_ridge.fit(X_train_tree_scaled, y_train)

# Get the best parameters and best model
best_params_ridge = grid_search_ridge.best_params_
best_model_ridge = grid_search_ridge.best_estimator_
```

```
print(f"Best Parameters for Ridge Regression: {best_params_ridge}")

# Test the best model on the test set
y_test_pred_ridge = best_model_ridge.predict(X_test_tree_scaled)
test_mse_ridge = mean_squared_error(y_test, y_test_pred_ridge)
test_r2_ridge = r2_score(y_test, y_test_pred_ridge)

print(f"Test MSE: {test_mse_ridge:.4f}")
print(f"Test R2: {test_r2_ridge:.4f}")

# 6. Lasso Regression with Tree-based Features
# Define the parameter grid for Lasso Regression
param_grid_lasso = {
    'alpha': [0.01, 0.1, 1.0]
}

# Initialize GridSearchCV for Lasso Regression
grid_search_lasso = GridSearchCV(estimator=lasso,
param_grid=param_grid_lasso, cv=3, scoring='r2', n_jobs=-1, verbose=2)

# Fit the model
grid_search_lasso.fit(X_train_tree_scaled, y_train)


# Get the best parameters and best model
best_params_lasso = grid_search_lasso.best_params_
best_model_lasso = grid_search_lasso.best_estimator_

print(f"Best Parameters for Lasso Regression: {best_params_lasso}")

# Test the best model on the test set
y_test_pred_lasso = best_model_lasso.predict(X_test_tree_scaled)
test_mse_lasso = mean_squared_error(y_test, y_test_pred_lasso)
test_r2_lasso = r2_score(y_test, y_test_pred_lasso)

print(f"Test MSE: {test_mse_lasso:.4f}")
print(f"Test R2: {test_r2_lasso:.4f}")

# 7. Decision Tree Regressor with Mutual Information Features
# Define the parameter grid for Decision Tree
param_grid_dt = {
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
```

```
# Initialize GridSearchCV for Decision Tree Regressor
grid_search_dt = GridSearchCV(estimator=dt, param_grid=param_grid_dt,
cv=3, scoring='r2', n_jobs=-1, verbose=2)

# Fit the model
grid_search_dt.fit(X_train_mi_scaled, y_train)

# Get the best parameters and best model
best_params_dt = grid_search_dt.best_params_
best_model_dt = grid_search_dt.best_estimator_

print(f"Best Parameters for Decision Tree: {best_params_dt}")

# Test the best model on the test set
y_test_pred_dt = best_model_dt.predict(X_test_mi_scaled)
test_mse_dt = mean_squared_error(y_test, y_test_pred_dt)
test_r2_dt = r2_score(y_test, y_test_pred_dt)

print(f"Test MSE: {test_mse_dt:.4f}")
print(f"Test R2: {test_r2_dt:.4f}")
```

Κώδικας 8: Σύγκριση Μοντέλων Πρόβλεψης

```
# 1. Scatter Plot with Regression Line
# Create a figure with 2x2 subplots for scatter plots with regression
line
fig, axes = plt.subplots(3, 3, figsize=(20, 18))

# Linear Regression
axes[0, 0].scatter(y_test, y_test_pred_lr, alpha=0.7)
axes[0, 0].plot([y_test.min(), y_test.max()], [y_test.min(),
y_test.max()], linestyle='--', color='k', lw=2)
axes[0, 0].set_xlabel('Actual CO2 Emissions')
axes[0, 0].set_ylabel('Predicted CO2 Emissions')
axes[0, 0].set_title('Actual vs Predicted CO2 Emissions (Linear Re-
gression)')

# Random Forest
axes[0, 1].scatter(y_test, y_test_pred_rf, alpha=0.7)
axes[0, 1].plot([y_test.min(), y_test.max()], [y_test.min(),
y_test.max()], linestyle='--', color='k', lw=2)
axes[0, 1].set_xlabel('Actual CO2 Emissions')
```

```
axes[0, 1].set_ylabel('Predicted CO2 Emissions')
axes[0, 1].set_title('Actual vs Predicted CO2 Emissions (Random For-
est)')

# Gradient Boosting
axes[0, 2].scatter(y_test, y_test_pred_gb, alpha=0.7)
axes[0, 2].plot([y_test.min(), y_test.max()], [y_test.min(),
y_test.max()], linestyle='--', color='k', lw=2)
axes[0, 2].set_xlabel('Actual CO2 Emissions')
axes[0, 2].set_ylabel('Predicted CO2 Emissions')
axes[0, 2].set_title('Actual vs Predicted CO2 Emissions (Gradient
Boosting)')

# XGBoost
axes[1, 0].scatter(y_test, y_test_pred_xgb, alpha=0.7)
axes[1, 0].plot([y_test.min(), y_test.max()], [y_test.min(),
y_test.max()], linestyle='--', color='k', lw=2)
axes[1, 0].set_xlabel('Actual CO2 Emissions')
axes[1, 0].set_ylabel('Predicted CO2 Emissions')
axes[1, 0].set_title('Actual vs Predicted CO2 Emissions (XGBoost)')

# Ridge Regression
axes[1, 1].scatter(y_test, y_test_pred_ridge, alpha=0.7)
axes[1, 1].plot([y_test.min(), y_test.max()], [y_test.min(),
y_test.max()], linestyle='--', color='k', lw=2)
axes[1, 1].set_xlabel('Actual CO2 Emissions')
axes[1, 1].set_ylabel('Predicted CO2 Emissions')
axes[1, 1].set_title('Actual vs Predicted CO2 Emissions (Ridge Regres-
sion)')

# Lasso Regression
axes[1, 2].scatter(y_test, y_test_pred_lasso, alpha=0.7)
axes[1, 2].plot([y_test.min(), y_test.max()], [y_test.min(),
y_test.max()], linestyle='--', color='k', lw=2)
axes[1, 2].set_xlabel('Actual CO2 Emissions')
axes[1, 2].set_ylabel('Predicted CO2 Emissions')
axes[1, 2].set_title('Actual vs Predicted CO2 Emissions (Lasso Regres-
sion)')

# Decision Tree
axes[2, 0].scatter(y_test, y_test_pred_dt, alpha=0.7)
axes[2, 0].plot([y_test.min(), y_test.max()], [y_test.min(),
y_test.max()], linestyle='--', color='k', lw=2)
axes[2, 0].set_xlabel('Actual CO2 Emissions')
axes[2, 0].set_ylabel('Predicted CO2 Emissions')
```

```
axes[2, 0].set_title('Actual vs Predicted CO2 Emissions (Decision
Tree)')

# Hide unused subplots
axes[2, 1].axis('off')
axes[2, 2].axis('off')

# Adjust layout
plt.tight_layout()
plt.show()

# 2. Residual Plot
# Residual Plot for All Models
# Create a figure with 3x3 subplots for residual plots
fig, axes = plt.subplots(3, 3, figsize=(20, 18))

# Linear Regression
axes[0, 0].scatter(y_test, y_test_pred_lr - y_test, alpha=0.7)
axes[0, 0].hlines(0, y_test.min(), y_test.max(), linestyle='--',
color='k', lw=2)
axes[0, 0].set_xlabel('Actual CO2 Emissions')
axes[0, 0].set_ylabel('Residuals')
axes[0, 0].set_title('Residuals (Linear Regression)')

# Random Forest
axes[0, 1].scatter(y_test, y_test_pred_rf - y_test, alpha=0.7)
axes[0, 1].hlines(0, y_test.min(), y_test.max(), linestyle='--',
color='k', lw=2)
axes[0, 1].set_xlabel('Actual CO2 Emissions')
axes[0, 1].set_ylabel('Residuals')
axes[0, 1].set_title('Residuals (Random Forest)')

# Gradient Boosting
axes[0, 2].scatter(y_test, y_test_pred_gb - y_test, alpha=0.7)
axes[0, 2].hlines(0, y_test.min(), y_test.max(), linestyle='--',
color='k', lw=2)
axes[0, 2].set_xlabel('Actual CO2 Emissions')
axes[0, 2].set_ylabel('Residuals')
axes[0, 2].set_title('Residuals (Gradient Boosting)')

# XGBoost
axes[1, 0].scatter(y_test, y_test_pred_xgb - y_test, alpha=0.7)
axes[1, 0].hlines(0, y_test.min(), y_test.max(), linestyle='--',
color='k', lw=2)
axes[1, 0].set_xlabel('Actual CO2 Emissions')
```

```
axes[1, 0].set_ylabel('Residuals')
axes[1, 0].set_title('Residuals (XGBoost)')

# Ridge Regression
axes[1, 1].scatter(y_test, y_test_pred_ridge - y_test, alpha=0.7)
axes[1, 1].hlines(0, y_test.min(), y_test.max(), linestyle='--',
color='k', lw=2)
axes[1, 1].set_xlabel('Actual CO2 Emissions')
axes[1, 1].set_ylabel('Residuals')
axes[1, 1].set_title('Residuals (Ridge Regression)')

# Lasso Regression
axes[1, 2].scatter(y_test, y_test_pred_lasso - y_test, alpha=0.7)
axes[1, 2].hlines(0, y_test.min(), y_test.max(), linestyle='--',
color='k', lw=2)
axes[1, 2].set_xlabel('Actual CO2 Emissions')
axes[1, 2].set_ylabel('Residuals')
axes[1, 2].set_title('Residuals (Lasso Regression)')

# Decision Tree
axes[2, 0].scatter(y_test, y_test_pred_dt - y_test, alpha=0.7)
axes[2, 0].hlines(0, y_test.min(), y_test.max(), linestyle='--',
color='k', lw=2)
axes[2, 0].set_xlabel('Actual CO2 Emissions')
axes[2, 0].set_ylabel('Residuals')
axes[2, 0].set_title('Residuals (Decision Tree)')

# Hide unused subplots
axes[2, 1].axis('off')
axes[2, 2].axis('off')

# Adjust layout
plt.tight_layout()
plt.show()

# 3. Distribution Plot

# Distribution Plot for All Models
# Create a figure with 3x3 subplots for distribution plots
fig, axes = plt.subplots(3, 3, figsize=(20, 18))

# Linear Regression
sns.histplot(y_test - y_test_pred_lr, kde=True, ax=axes[0, 0],
color='blue')
axes[0, 0].set_title('Residual Distribution (Linear Regression)')
```

```
axes[0, 0].set_xlabel('Residuals')

# Random Forest
sns.histplot(y_test - y_test_pred_rf, kde=True, ax=axes[0, 1],
color='green')
axes[0, 1].set_title('Residual Distribution (Random Forest)')
axes[0, 1].set_xlabel('Residuals')

# Gradient Boosting
sns.histplot(y_test - y_test_pred_gb, kde=True, ax=axes[0, 2],
color='red')
axes[0, 2].set_title('Residual Distribution (Gradient Boosting)')
axes[0, 2].set_xlabel('Residuals')

# XGBoost
sns.histplot(y_test - y_test_pred_xgb, kde=True, ax=axes[1, 0],
color='purple')
axes[1, 0].set_title('Residual Distribution (XGBoost)')
axes[1, 0].set_xlabel('Residuals')

# Ridge Regression
sns.histplot(y_test - y_test_pred_ridge, kde=True, ax=axes[1, 1],
color='orange')
axes[1, 1].set_title('Residual Distribution (Ridge Regression)')
axes[1, 1].set_xlabel('Residuals')

# Lasso Regression
sns.histplot(y_test - y_test_pred_lasso, kde=True, ax=axes[1, 2],
color='brown')
axes[1, 2].set_title('Residual Distribution (Lasso Regression)')
axes[1, 2].set_xlabel('Residuals')

# Decision Tree
sns.histplot(y_test - y_test_pred_dt, kde=True, ax=axes[2, 0],
color='pink')
axes[2, 0].set_title('Residual Distribution (Decision Tree)')
axes[2, 0].set_xlabel('Residuals')

# Hide unused subplots
axes[2, 1].axis('off')
axes[2, 2].axis('off')

# Adjust layout
plt.tight_layout()
plt.show()
```

Κώδικας 9: Δημιουργία DataFrame για Πραγματικές και Προβλεπόμενες τιμές

```
# Create DataFrame for actual and predicted values
results = pd.DataFrame({
    'Actual': y_test,
    'Predicted_RF': y_test_pred_rf,
    'Predicted_LR': y_test_pred_lr,
    'Predicted_GB': y_test_pred_gb,
    'Predicted_XGB': y_test_pred_xgb,
    'Predicted_Ridge': y_test_pred_ridge,
    'Predicted_Lasso': y_test_pred_lasso,
    'Predicted_DT': y_test_pred_dt
})

# Save to an Excel file
results.to_excel('model_predictions.xlsx', sheet_name='Predictions',
index=False)

print("Results saved to model_predictions.xlsx")
```