ΠΡΟΒΛΕΨΗ ΚΛΙΜΑΤΙΚΗΣ ΖΩΝΗΣ ΑΠΟ ΚΑΙΡΙΚΑ ΔΕΔΟΜΕΝΑ: ΕΝΑ ΠΡΟΒΛΗΜΑ ΤΑΞΙΝΟΜΗΣΗΣ ΒΑΣΙΣΜΕΝΟ ΣΤΗΝ ΑΝΑΛΥΣΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Ραφαήλ Παπαστάμου - 58307
Φοιτητής 8ου Εξαμήνου
rafapapa14@ee.duth.gr

Απόστολος Θωμάς Παπάγγελος - 58313
Φοιτητής 8ου Εξαμήνου
apospapa16@ee.duth.gr

ABSTRACT

Παραδοσιακά οι προβλέψεις του καιρού γίνονται με τη βοήθεια πολύπλοκων μοντέλων φυσικής τα οποία βασίζονται σε μετρήσεις ατμοσφαιρικών συνθηκών σε εκτενείς χρονικές περιόδους, όπου απαιτείται μεγάλη υπολογιστική δύναμη.

Στη μετεωρολογία είναι αρκετά συνηθισμένο να εξετάζονται οι σχέσεις και τα μοτίβα που χαρακτηρίζουν το εκάστοτε κλίμα, προκειμένου να γίνονται πιο ακριβείς προβλέψεις.

Επίσης, τα τελευταία χρόνια παρατηρείται μία μεγάλη «έκρηξη» στην επιστήμη της ανάλυσης δεδομένων και τη μαθησιακή μάθηση και έχουν «κατακτήσει» πολλούς διαφορετικούς επιστημονικούς – και μη – κλάδους. Η επιστήμη της μετεωρολογίας είναι μια από αυτές και έχει πολλά να κερδίσει από την ανάλυση δεδομένων για να κάνει πιο ακριβείς και αξιόπιστες προγνώσεις.

Στο πλάισιο αυτό, αποφασίστηκε να ερευνηθούν μέθοδοι ανάλυσης δεδομένων και μηχανικής μάθησης, με σκοπό τη πραγματοποίηση προβλέψεων κατηγοριοποίησης κλιμάτων βασισμένα σε μετεωρολογικά δεδομένα, σε πραγματικό χρόνο, απαιτώντας έτσι την ελάχιστη υπολογιστική δύναμη.

Στην έρευνα χρησιμοποιήθηκαν διάφορες τεχνικές ανάλυσης δεδομένων και μηχανικής μάθησης, χρησιμοποιώντας δημόσια μετεωρολογικά δεδομένα, με σκοπό τη δημιουργία του βέλτιστου μοντέλου, το οποίο να μπορεί να προβλέψει σε ποια κλιματική ζώνη ανήκει μία ημερήσια μετεωρολογική μέτρηση.

Μετά από πολλά πειράματα με διαφορετικές μεθόδους και στρατηγικές, το μοντέλο που τελικά επιλέχτηκε, προέκυψε από τον συνδυασμό των μεθόδων «TPOTClassifier» και «XGBoostClassifier» σε χαρακτηριστικά που προέκυψαν από τη μέθοδο «Recursive Feature Elimination with Cross Validation».

Το F1 score που πέτυχε το μοντέλο είναι 0.7251, το οποίο σημαίνει ότι το μοντέλο έχει αρκετά καλή απόδοση στις προβλέψεις που πραγματοποιεί.

1. ΕΙΣΑΓΩΓΗ

Οι κλιματικές ζώνες είναι κρίσιμοι και καθοριστικοί παράγοντες του φυσικού περιβάλλοντος και επηρεάζουν διάφορες ανθρώπινες δραστηριότητες, συμπεριλαμβανομένης της γεωργίας και του πολεοδομικού σχεδιασμού. Η ακριβής ταξινόμηση αυτών των ζωνών με βάση τα καιρικά δεδομένα είναι απαραίτητη τόσο για επιστημονική έρευνα όσο και για πρακτικές εφαρμογές. Με την έλευση των μεγάλων δεδομένων και των προηγμένων τεχνικών μηχανικής μάθησης, είναι πλέον δυνατό να αξιοποιηθούν τεράστιες ποσότητες μετεωρολογικών δεδομένων για την πρόβλεψη των κλιματικών ζωνών με υψηλή ακρίβεια.

Αυτή η μελέτη θα εστιάσει στην πρόβλεψη κλιματικών ζωνών χρησιμοποιώντας δεδομένα καιρού μέσω μιας ολοκληρωμένης ανάλυσης δεδομένων και προσέγγισης μηχανικής μάθησης. Αξιοποιώντας διάφορα μοντέλα, θα αναπτυχθεί ένα ισχυρό σύστημα ταξινόμησης που θα μπορεί να προσδιορίσει με ακρίβεια τις κλιματικές ζώνες από τα καιρικά μοτίβα. Δηλαδή, δίνοντας στο σύστημα τα καιρικά δεδομένα για ένα μέρος, θα προβλέπεται το είδος του κλίματός του.

Το κίνητρο πίσω από αυτή τη μελέτη πηγάζει από την ανάγκη για ακριβή ταξινόμηση των κλιματικών ζωνών για την υποστήριξη της λήψης αποφάσεων σε τομείς όπως η γεωργία, η περιβαλλοντική διαχείριση και η αστική ανάπτυξη. Οι παραδοσιακές μέθοδοι ταξινόμησης του κλίματος, βασίζονται σε προκαθορισμένα όρια και χειροκίνητες ερμηνείες. Αν και αυτές οι μέθοδοι ήταν αποτελεσματικές, συχνά δεν έχουν την ευελιξία και την προσαρμοστικότητα που προσφέρουν τα σύγχρονα μοντέλα μηχανικής μάθησης.

2. ΑΠΟΦΑΣΕΙΣ ΠΡΙΝ ΤΗ ΠΡΟΕΡΓΑΣΙΑ

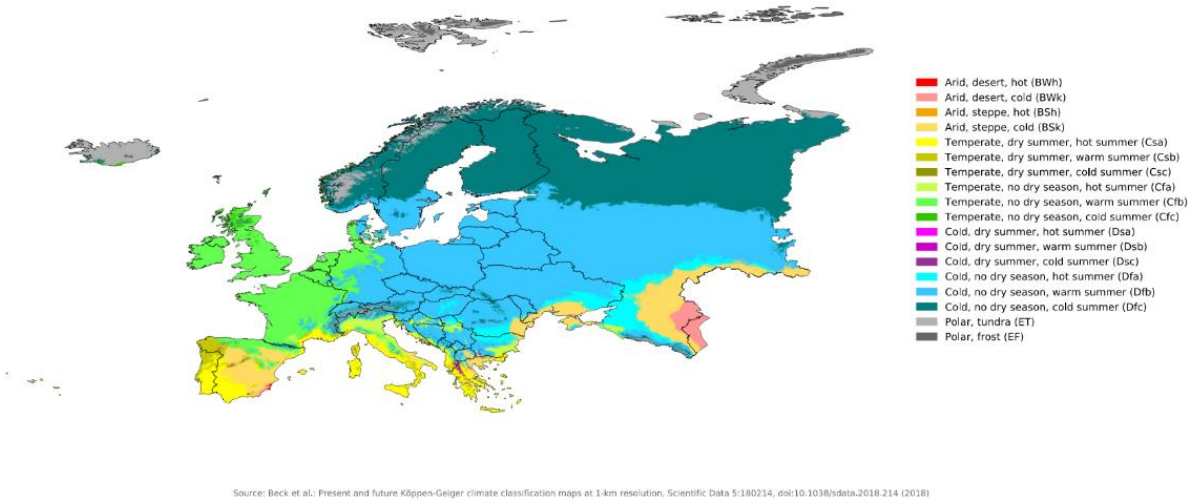
Προτού γίνει οτιδήποτε θα πρέπει να παρθούν ορισμένες αποφάσεις όσο αφορά τη φύση του προβλήματος. Συγκεκριμένα, πρέπει να αποφασιστεί το είδος των τοποθεσιών που θα χρησιμοποιηθούν όσο και των ίδιων των προβλέψεων, να οριστούν οι μετρήσεις που θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου αλλά και να αποφασιστούν οι ημερομηνίες των δεδομένων που θα ληφθούν υπόψιν.

2.1 ΕΠΙΛΟΓΗ ΤΟΥ ΕΙΔΟΥΣ ΤΩΝ ΤΟΠΟΘΕΣΙΩΝ ΠΟΥ ΘΑ ΠΡΟΒΛΕΠΟΝΤΑΙ

Αρχικά, θα πρέπει να οριστεί το είδος των τοποθεσιών που θα προβλέπει το μοντέλο. Οι επιλογές για τα διαφορετικά είδη τοποθεσιών είναι πολλές. Συγκεκριμένα, μπορούν να χρησιμοποιηθούν δεδομένα από όλο τον κόσμο και να προβλέπονται από το μοντέλο οι χώρες. Αυτό είναι ιδιαίτερα δύσκολο, καθώς ο αριθμός των χωρών είναι μεγάλος και χώρα με χώρα δεν διαφέρει έντονα κλιματικά (π.χ. δύο γειτονικές χώρες). Εναλλακτικά, το μοντέλο θα μπορούσε να προβλέπει πόλεις. Αυτό, όπως προηγουμένως, θα ήταν πολύ δύσκολο, εκτός αν χρησιμοποιούνταν λίγες πόλεις με μεγάλες διαφορές ως προς το κλίμα τους. Άλλος ένας τρόπος θα ήταν να χρησιμοποιηθούν για τη πρόβλεψη οι γεωγραφικές συντεταγμένες, έτσι ώστε το μοντέλο να προβλέπει κατά προσέγγιση τη τοποθεσία με βάση τις συντεταγμένες και βάση αυτών να υπολογίζονται τα σφάλματα.

Αυτό που επιλέγεται, τελικά, είναι να χρησιμοποιηθούν αποκλειστικά πόλεις της Ευρώπης, χωρίζοντάς τες σε διαφορετικές κατηγορίες με βάση το κλίμα τους.

Συγκεκριμένα, θα πρέπει να χωρίσουμε την Ευρώπη σε διαφορετικές κλιματικές ζώνες. Σύμφωνα με τη ταξινόμηση Κεππεν [1] η Ευρώπη χωρίζεται στις εξής κλιματικές ζώνες:



Εικόνα 1: *Φωτογραφία της Ευρώπης με τις διαφορετικές κλιματικές ζώνες σύμφωνα με τη κλιματική ταξινόμηση Κέππεν [2]*

Συγκεκριμένα, η Ευρώπη, σύμφωνα με τη ταξινόμηση Κέππεν μπορεί να χωριστεί σε αυτές τις διαφορετικές κλιματικές κατηγορίες:

Code	Description	Group	Precipitation Type	Level of Heat
Af	Tropical rainforest climate	Tropical	Rainforest	
Am	Tropical monsoon climate	Tropical	Monsoon	
As	Tropical dry savanna climate	Tropical	Savanna, Dry	
Aw	Tropical savanna, wet	Tropical	Savanna, Wet	
BSh	Hot semi-arid (steppe) climate	Arid	Steppe	Hot
BSk	Cold semi-arid (steppe) climate	Arid	Steppe	Cold
BWh	Hot deserts climate	Arid	Desert	Hot
BWk	Cold desert climate	Arid	Desert	Cold
Cfa	Humid subtropical climate	Temperate	Without dry season	Hot summer
Cfb	Temperate oceanic climate	Temperate	Without dry season	Warm summer
Cfc	Subpolar oceanic climate	Temperate	Without dry season	Cold summer
Csa	Hot-summer Mediterranean climate	Temperate	Dry summer	Hot summer
Csb	Warm-summer Mediterranean climate	Temperate	Dry summer	Warm summer
Csc	Cool-summer Mediterranean climate	Temperate	Dry summer	Cold summer
Cwa	Monsoon-influenced humid subtropical climate	Temperate	Dry winter	Hot summer
Cwb	Subtropical highland climate or temperate oceanic climate with dry winters	Temperate	Dry winter	Warm summer
Cwc	Cold subtropical highland climate or subpolar oceanic climate with dry winters	Temperate	Dry winter	Cold summer
Dfa	Hot-summer humid continental climate	Cold (continental)	Without dry season	Hot summer
Dfb	Warm-summer humid continental climate	Cold (continental)	Without dry season	Warm summer
Dfc	Subarctic climate	Cold (continental)	Without dry season	Cold summer
Dfd	Extremely cold subarctic climate	Cold (continental)	Without dry season	Very cold winter
Dsa	Hot, dry-summer continental climate	Cold (continental)	Dry summer	Hot summer
Dsb	Warm, dry-summer continental climate	Cold (continental)	Dry summer	Warm summer
Dsc	Dry-summer subarctic climate	Cold (continental)	Dry summer	Cold summer
Dwa	Monsoon-influenced hot-summer humid continental climate	Cold (continental)	Dry winter	Hot summer
Dwb	Monsoon-influenced warm-summer humid continental climate	Cold (continental)	Dry winter	Warm summer
Dwc	Monsoon-influenced subarctic climate	Cold (continental)	Dry winter	Cold summer
Dwd	Monsoon-influenced extremely cold subarctic climate	Cold (continental)	Dry winter	Very cold winter
EF	Ice cap climate	Polar	Ice cap	
ET	Tundra	Polar	Tundra	

Εικόνα 2: *Λίστα των διαφορετικών κλιματικών ζωνών σύμφωνα με τη κλιματική ταξινόμηση Κέππεν [3]*

Παρόλα αυτά, οι κατηγορίες αυτές θεωρούνται πολύ συγκεκριμένες και θα δυσκολευόταν το μοντέλο να κάνει ακριβείς προβλέψεις με τόσες επιλογές. Έτσι, θα πρέπει να απλοποιηθεί η κατηγοριοποίηση, ορίζοντας νέες κατηγορίες που περιλαμβάνουν τις κλιματικές ζώνες που συναντιούνται στην Ευρώπη συχνότερα. Συγκεκριμένα, οι νέες κλιματικές κατηγορίες θα είναι οι εξής:

- Μεσογειακό Κλίμα (Mediterranean Climate)
    - Ζεστά, ξηρά καλοκαίρια
    - Ήπιοι, υγροί χειμώνες
    - Συναντιέται στη Νότια Ευρώπη.
- Ωκεάνιο Κλίμα (Oceanic Climate)
    - Ήπιες θερμοκρασίες
    - Υψηλή υγρασία
    - Βροχοπτώσεις καθ’ όλη τη διάρκεια της χρονιάς
    - Συναντιέται στη Δυτική και Βορειοδυτική Ευρώπη
- Ηπειρωτικό Κλίμα (Continental Climate)
    - Ζεστά καλοκαίρια
    - Κρύοι χειμώνες
    - Συναντιέται στη Κεντρική και στην Ανατολική Ευρώπη.

Αυτές οι κατηγορίες είναι οι συνηθέστερες της Ευρώπης και κάθε Ευρωπαϊκή πόλη θα πρέπει να ενταχθεί σε μία από αυτές, σύμφωνα με τις γεωγραφικές της συντεταγμένες.

2.2 ΣΚΟΠΟΣ ΤΟΥ ΜΟΝΤΕΛΟΥ

Αφού αποφασίστηκε ο τρόπος κατηγοριοποίησης των Ευρωπαϊκών πόλεων σύμφωνα με τη γεωγραφική τους θέση, που αντιστοιχίζεται σε διαφορετικές κλιματικές ζώνες, είναι ξεκάθαρος πλέον ο σκοπός του μοντέλου που θα δημιουργηθεί.

Σκοπός του μοντέλου, λοιπόν, είναι να εκπαιδευτεί με καιρικά δεδομένα διάφορων πόλεων, από όλη την Ευρώπη, και τελικά να είναι ικανό να προβλέψει από άγνωστα καιρικά δεδομένα ποια είναι η κλιματική ζώνη από την οποία προέρχονται αυτά τα δεδομένα.

Το πρόβλημα που καλείται να επιλύσει το μοντέλο είναι ένα classification πρόβλημα το οποίο επιλύεται με ανάλυση δεδομένων και μηχανική μάθηση.

2.3 ΕΠΙΛΟΓΗ ΠΟΛΕΩΝ

Η επιλογή των πόλεων για το δίκτυο είναι πολύ σημαντική, καθώς σύμφωνα με τα δεδομένα αυτών των πόλεων θα γίνονται οι προβλέψεις. Για την ακρίβεια του δικτύου, θα πρέπει να επιλεγούν πόλεις οι οποίες θα παρουσιάζουν ξεκάθαρα μοτίβα που βασίζονται στη κλιματική ζώνη κάθε πόλης, καθώς σε αυτά θα βασιστούν οι προβλέψεις του μοντέλου. Για παράδειγμα, θα πρέπει να αποφευχθεί η επιλογή κάποιας πόλης η οποία βρίσκεται ανάμεσα σε δύο κλιματικές ζώνες και παρουσιάζει καιρικά φαινόμενα και των δύο, αυτών, ζωνών.

2.4 ΕΠΙΛΟΓΗ ΣΗΜΑΝΤΙΚΩΝ ΜΕΤΡΗΣΕΩΝ

Για την ορθή εκπαίδευση του μοντέλου, με σκοπό τις ακριβείς προβλέψεις, θα πρέπει να χρησιμοποιηθούν μετεωρολογικές μετρήσεις που θεωρούνται σημαντικές και αναγκαίες για τον ορισμό της κλιματικής ζώνης μιας τοποθεσίας. Αρχικά, οι ελάχιστες, μέγιστες και μέσες θερμοκρασίες θεωρούνται άκρως σημαντικές. Ακόμη, κάποιες άλλες σημαντικές ατμοσφαιρικές μετρήσεις είναι ο υετός (πιθανότητα βροχής), η ατμοσφαιρική πίεση και η ένταση/κατεύθυνση του ανέμου.

Αυτές είναι οι μετρήσεις που θεωρούνται απαραίτητες για τον ορθό χαρακτηρισμό μιας κλιματικής ζώνης.

2.5 ΕΠΙΛΟΓΗ ΕΝΑΡΞΗΣ ΗΜΕΡΟΜΗΝΙΑΣ

Για το δίκτυο που θα δημιουργηθεί θα πρέπει να αποφασιστεί ποια θα είναι η ημερομηνία έναρξης της συλλογής των δεδομένων. Συγκεκριμένα, δύο είναι τα πράγματα που θα πρέπει να ληφθούν υπόψιν: η διαθεσιμότητα μαζί με τη συνέπεια των δεδομένων και η σχετικότητα με τη κλιματική αλλαγή. Αναλυτικότερα:

- Διαθεσιμότητα και συνέπεια δεδομένων: η συλλογή μετεωρολογικών δεδομένων τυποποιήθηκε και έγινε αξιόπιστη τα τελευταία χρόνια με αποτέλεσμα να υπάρχουν πλέον δεδομένα συνεπή και υψηλής ποιότητας που μπορούν να χρησιμοποιηθούν με αξιοπιστία.
- Σε σχέση με τη κλιματική αλλαγή: είναι γνωστό πως η κλιματική αλλαγή είναι ένα από τα σημαντικότερα προβλήματα των τελευταίων δεκαετιών. Τα φαινόμενα που οφείλονται στη κλιματική αλλαγή έγιναν κυρίως αισθητά τα τέλη της δεκαετίας του ’90 με την έναρξη του 21^{ου} αιώνα και η επιστήμη της μετεωρολογίας επηρεάστηκε σε μεγάλο βαθμό.

Έτσι, αποφασίζεται ότι μία καλή ημερομηνία έναρξης για τη συλλογή των δεδομένων είναι το 2000. Το 2000 ήταν ήδη τυποποιημένα τα μετεωρολογικά δεδομένα, άρα μπορούν να χρησιμοποιηθούν με αξιοπιστία. Επίσης, 20 χρόνια είναι αρκετά για την εκπαίδευση του μοντέλου που θα αναπτυχθεί. Τέλος, θα ληφθούν υπόψιν πρόσφατα δεδομένα, όπου η κλιματική αλλαγή είναι παρούσα, με αποτέλεσμα να προκύπτουν ακριβείς προβλέψεις για παροντικά και μελλοντικά δεδομένα.

3. ΠΡΟΕΡΓΑΣΙΑ

Αφού πάρθηκαν οι κατάλληλες αποφάσεις για την εργασία, μπορεί να γίνει η προεργασία. Η προεργασία αφορά την εύρεση ενός κατάλληλου dataset για το πρόβλημα που καλείται να αντιμετωπίσει το μοντέλο, αλλά και κάποιες άλλες λεπτομέρειες.

3.1 ΕΠΙΛΟΓΗ ΚΑΤΑΛΛΗΛΟΥ DATASET

Η επιλογή ενός κατάλληλου dataset είναι άκρως σημαντική, καθώς από αυτό θα κριθεί σε μεγάλο βαθμό η ακρίβεια και η αξιοπιστία των προβλέψεων του μοντέλου.

Αρχικά, το dataset θα πρέπει να περιέχει δεδομένα για αρκετές Ευρωπαϊκές πόλεις, οι οποίες να βρίσκονται σε όλα τα γεωγραφικά μέρη της, ώστε να μπορούν να κατηγοριοποιηθούν στις διαφορετικές κλιματικές υποκατηγορίες που ορίστηκαν. Οι

πόλεις καλό θα ήταν να έχουν ορισμένες και τις γεωγραφικές τους συντεταγμένες, ώστε να μπορέσει να γίνει αντιστοίχιση με τον τύπο της κλιματικής ζώνης τους.

Επίσης, θα πρέπει να υπάρχει πληθώρα ποιοτικών και αξιόπιστων δεδομένων από το 2000 και μετέπειτα. Ιδανικά, θα πρέπει να υπάρχουν δεδομένα για όλες τις εποχές του χρόνου.

Τέλος, θα πρέπει το dataset να περιέχει τις μετεωρολογικές μετρικές που θεωρήθηκαν σημαντικές (π.χ. ελάχιστη-μέγιστη θερμοκρασία κλπ).

Το dataset που επιλέγεται, τελικά, είναι το «The Weather Dataset», το οποίο δημιουργήθηκε από τον χρήστη @guillemservera και είναι δημοσιοποιημένο στη γνωστή πλατφόρμα «Kaggle» [4]. Το συγκεκριμένο dataset ενημερώνεται με νέα δεδομένα κάθε εβδομάδα, με σκοπό να είναι όσο το δυνατόν πιο ανανεωμένο γίνεται. Το dataset αυτό πληροί όλες τις προϋποθέσεις που τέθηκαν και θα μπορέσει να εκπαιδεύσει το μοντέλο και να κάνει αξιόπιστες προβλέψεις.

4. ΑΝΑΛΥΣΗ ΔΕΔΟΜΕΝΩΝ

Το πρώτο βήμα που πρέπει να γίνει είναι η ανάλυση των δεδομένων. Συγκεκριμένα, θα πρέπει να αναλυθεί το dataset που επιλέχτηκε, ώστε να βρεθούν πιθανά λάθη και να διορθωθούν και να διαλεχθούν τα χρήσιμα και απαραίτητα δεδομένα που θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου και τη δοκιμή του.

4.1 ΕΞΟΡΥΞΗ ΕΠΙΘΥΜΗΤΩΝ ΠΟΛΕΩΝ

Το dataset που επιλέχτηκε έχει μετεωρολογικά δεδομένα για χώρες και πόλεις όλου του κόσμου. Οι επιθυμητές πόλεις για το μοντέλο που θα χρησιμοποιηθεί είναι όλες οι Ευρωπαϊκές πόλεις, οι οποίες όμως να μπορούν να κατηγοριοποιηθούν, με βάση τις γεωγραφικές τους συντεταγμένες, στις τρεις κλιματικές ζώνες που ορίστηκαν (μεσογειακή, ωκεάνια, ηπειρωτική). Δηλαδή, μια πόλη η οποία βρίσκεται σε ένα γεωγραφικό σημείο το οποίο βρίσκεται ανάμεσα σε δύο από αυτές τις ζώνες και η οποία, ως αποτέλεσμα, παρουσιάζει μετεωρολογικά φαινόμενα και των δύο ζωνών, θα πρέπει να απορρίπτεται. Διαφορετικά, το μοντέλο που θα εκπαιδευτεί δε θα καταλήγει σε ακριβείς προβλέψεις, διότι θα υπάρχουν δεδομένα που δεν μπορούν εύκολα να κατηγοριοποιηθούν σε μία από τις τρεις επιλογές. Το πρόβλημα είναι classification φύσης, για αυτό απαιτούνται ξεκάθαρες κατηγοριοποιήσεις.

Το πρώτο βήμα είναι να εξορυχθούν όλες οι Ευρωπαϊκές χώρες από το σύνολο των χωρών του database, χρησιμοποιώντας το γνώρισμα – feature - «continent» (Ήπειρος).

Στη συνέχεια, πρέπει να εξορυχθούν όλες οι Ευρωπαϊκές πόλεις από το σύνολο των πόλεων του database, χρησιμοποιώντας τις χώρες που εξορύχθηκαν προηγουμένως.

4.2 ΑΝΤΙΣΤΟΙΧΙΣΗ ΤΩΝ ΠΟΛΕΩΝ ΜΕ ΤΗ ΚΛΙΜΑΤΙΚΗ ΖΩΝΗ ΠΟΥ ΑΝΗΚΟΥΝ

Σε αυτό το σημείο θα πρέπει να αντιστοιχηθούν οι πόλεις που εξορύχθηκαν με τις τρεις διαφορετικές κατηγορίες που ορίστηκαν για τις τρεις διαφορετικές κλιματικές ζώνες (μεσογειακή, ωκεάνια, ηπειρωτική).

Για να γίνει αυτή η αντιστοίχιση θα χρησιμοποιηθούν οι γεωγραφικές συντεταγμένες των πόλεων. Το database που επιλέχτηκε έχει διαθέσιμες αυτές τις

συντεταγμένες για κάθε πόλη. Συγκεκριμένα, κάθε πόλη έχει μία τιμή «latitude» (γεωγραφικό πλάτος) και μία τιμή «longitude» (γεωγραφικό μήκος).

Χρησιμοποιώντας έναν κατάλληλο χάρτη Κέππεν [5], ο οποίος περιέχει τις διαφορετικές κλιματικές ζώνες και την αντιστοίχισή τους με γεωγραφικές συντεταγμένες, θα μπορέσει να γίνει η αντιστοίχιση των επιθυμητών πόλεων με τη κλιματική τους κατηγορία.

Υπάρχουν χάρτες Κέππεν διαφορετικής ακρίβειας που μπορούν να χρησιμοποιηθούν. Η διαφορά είναι ότι όσο μεγαλύτερη είναι η ακρίβεια ενός χάρτη, τόσο περισσότερη ώρα διαρκεί η διαδικασία αντιστοίχισης των γεωγραφικών συντεταγμένων με τις κλιματικές ζώνες, αλλά συγχρόνως, θα δημιουργηθούν δεδομένα τα οποία θα οδηγήσουν σε ακριβέστερες προβλέψεις. Η επιλογή ακρίβειας χάρτη Κέππεν έγγυται στο αν υπάρχει η επιθυμία για πιο γρήγορη ή για πιο ακριβής αντιστοίχιση. Στη προκειμένη περίπτωση, καθώς το πρόβλημα αφορά εκπαίδευση μοντέλου, επιθυμείται πιο ακριβής αντιστοίχιση, για αυτό θα χρησιμοποιηθεί ο ακριβέστερος Κέππεν χάρτης από το σύνολο.

Για να μπορέσει να διαβαστεί το format του χάρτη (GeoTIFF) θα πρέπει να χρησιμοποιηθεί η βιβλιοθήκη «rosterio» [6]. Με τη χρήση αυτής της βιβλιοθήκης γίνεται η αντιστοίχιση των γεωμετρικών συντεταγμένων των πόλεων με τις κλιματικές τους ζώνες.

Η αντιστοίχιση επιστρέφει τους κωδικούς των διαφορετικών κλιματικών τύπων σύμφωνα με το μοντέλο Κέππεν. Κάθε ένας από αυτούς τους κωδικούς θα πρέπει να αντιστοιχιθεί σε μία από τις τρεις κατηγορίες κλιμάτων που επιλέχτηκαν (μεσογειακή, ωκεανική, ηπειρωτική). Συγκεκριμένα, σύμφωνα με το μοντέλο Κέππεν, η αντιστοίχιση θα είναι η εξής:

Κλιματική Κατηγορία	**Κωδικοί Μοντέλου Κέππεν**
Μεσογειακό Κλίμα	Csa, Csb
Ωκεάνιο Κλίμα	Cfb, Cfc
Ηπειρωτικό Κλίμα	Dfa, Dfb, Dfc, Dfd, Dwc

Οι υπόλοιποι κωδικοί του Μοντέλου Κέππεν δεν παρατηρούνται στην Ευρώπη, ή δε μπορούν να κατηγοριοποιηθούν στις παραπάνω υποκατηγορίες, οπότε δε θα χρησιμοποιηθούν για τα δεδομένα του μοντέλου.

Η κλιματική κατηγορία κάθε πόλης αποθηκεύεται στο γνώρισμα «climate_type».

4.3 ΔΗΜΙΟΥΡΓΙΑ TRAIN ΚΑΙ TEST DATASET

Αρχικά, θα δημιουργηθεί ένα dataset με όλο το σύνολο των καιρικών δεδομένων για τις πόλεις που έχουν επιλεχτεί. Σε αυτό θα πρέπει να γίνουν μερικές αλλαγές, όπως η αφαίρεση των γνωρισμάτων (στηλών) που δεν χρειάζονται για το μοντέλο.

Συγκεκριμένα, θα αφαιρεθούν οι εξής στήλες από το dataset:

- «station_id»: αναφέρεται στον σταθμό που έγινε η μέτρηση των μετεωρολογικών τιμών – δε χρησιμεύει το μοντέλο αυτή η πληροφορία
- «sunshine_total_min»: είναι τα λεπτά της ημέρας που είχε ηλιοφάνεια μία πόλη. Αυτό το γνώρισμα αφαιρείται, καθώς ελάχιστες πόλεις έχουν τιμές για αυτό το γνώρισμα και θα είχε κακή επίδραση στο μοντέλο

- «snow_depth_mm»: είναι το πάχος του χιονιού σε περίπτωση χιονόπτωσης. Αφαιρείται για τον ίδιο λόγο όπως το προηγούμενο γνώρισμα – δεν υπάρχουν αρκετά διαθέσιμα δεδομένα.
- «peak_wind_gust_kmh»: μέγιστη ταχύτητα του ανέμου – επίσης δεν υπάρχουν αρκετά διαθέσιμα δεδομένα.

Για τα γνωρίσματα που αφαιρούνται λόγω της έλλειψης δεδομένων θα μπορούσε να γίνει imputation – να γεμίσουν δηλαδή οι κενές τιμές με κάποια μέθοδο π.χ. median value (μέση τιμή) της στήλης, αλλά εκτιμήθηκε ότι τα δεδομένα είναι τόσο λίγα για τα συγκεκριμένα γνωρίσματα, που δε θα βελτιωνόταν η κατάσταση. Θα χρησιμοποιηθούν τέτοιες μέθοδοι στη συνέχεια, για τα υπόλοιπα γνωρίσματα του dataset.

Στη συνέχεια, προστίθεται το γνώρισμα «climate_type» στο dataset που περιέχει τη κλιματική κατηγορία κάθε πόλης και θα αποτελεί το target.

Η χρονιά έναρξης των δεδομένων, όπως προαναφέρθηκε, είναι το 2000. Όλα τα δεδομένα που έχουν παλιότερη ημερομηνία θα αφαιρεθούν από το dataset.

Ακόμη, θα πρέπει να αφαιρεθούν όσες γραμμές θεωρούνται ελλιπείς. Από τις οκτώ μετεωρολογικές μετρήσεις θεωρείται ότι μία γραμμή είναι ελλιπής όταν έχει τουλάχιστον 4 κενές τιμές και θα αφαιρείται.

Το συνολικό dataset είναι πλέον έτοιμο. Τα γνωρίσματά του είναι τα εξής:

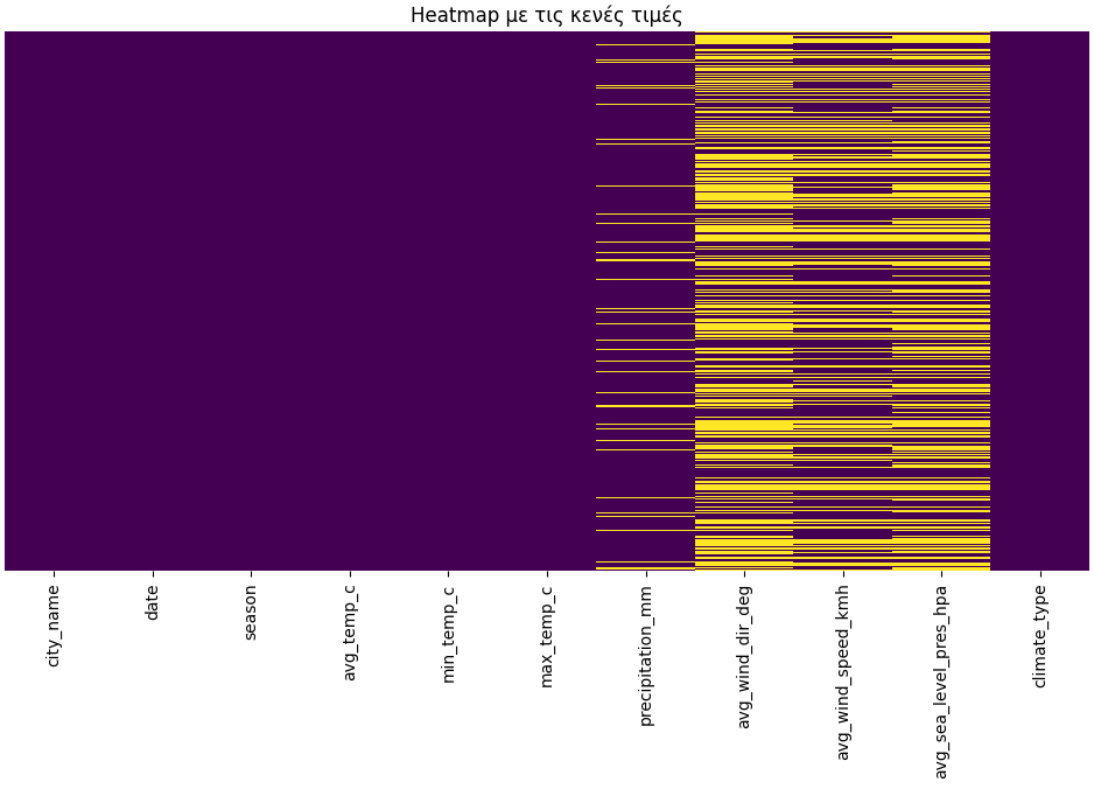
- «city_name»: είναι το όνομα της κάθε πόλης.
- «date»: είναι η ημερομηνία που έγιναν οι μετρήσεις. Δε θα χρησιμοποιηθεί άμεσα στην εκπαίδευση, αλλά μπορούν να δημιουργηθούν νέα γνωρίσματα με βάση αυτό, με σκοπό τις καλύτερες προβλέψεις. Αυτό θα γίνει σε κάποιο βήμα, αργότερα.
- «season»: η εποχή που έγινε η μέτρηση
- «avg_temp»: μέσος όρος θερμοκρασίας της ημέρας
- «min_temp»: η ελάχιστη θερμοκρασία της ημέρας
- «max_temp»: η μέγιστη θερμοκρασία της ημέρας
- «precipitation_mm»: χιλιοστά υετού που έπεσαν
- «avg_wind_dir_deg»: μέσος όρος της γωνίας του ανέμου τη συγκεκριμένη ημέρα (0° βόρειος άνεμος, 90° ανατολικός άνεμος, 180° νότιος άνεμος, 270° ανατολικός άνεμος)
- «avg_wind_speed_kmh»: μέση ταχύτητα του ανέμου
- «avg_sea_level_pres_hpa»: μέση ατμοσφαιρική πίεση (μετρημένη σε hPa)
- climate_type»: μία από τις τρεις κλιματικές κατηγορίες (μεσογειακή, ωκεάνια, ηπειρωτική)

Αυτό που μένει τώρα είναι να χωριστεί το συνολικό dataset στα δύο (train και test dataset). Το train dataset θα χρησιμοποιηθεί για να εκπαιδευτεί το μοντέλο, ενώ το test dataset θα περιέχει τα άγνωστα δεδομένα που θα χρησιμοποιηθούν για να γίνει εκτίμηση του πόσο καλό είναι το μοντέλο.

Ένα συνηθισμένο ποσοστό χωρισμού του dataset σε train και test είναιμ το 80/20, δηλαδή 80% των δεδομένων θα αποτελέσουν το train dataset και το υπόλοιπο 20% το test dataset. Αυτος ο χωρισμός γίνεται τυχαία χρησιμοποιώντας τη συνάρτηση «train_test_split» της βιβλιοθήκης «scikit-learn» [7].

4.4 ΧΕΙΡΙΣΜΟΣ ΚΕΝΩΝ ΤΙΜΩΝ (MISSING VALUES)

Τόσο το train dataset, όσο και το train dataset περιέχουν κενές τιμές. Στο παρακάτω heatmap παρουσιάζονται οι κενές τιμές του train dataset:



Εικόνα 3: *Heatmap plot που παρουσιάζονται οι κενές τιμές του train dataset*

Για να δημιουργηθούν plots (διαγράμματα) χρησιμοποιούνται οι βιβλιοθήκες «Matplotlib» [8] και «seaborn» [9].

Προτού γίνει το fit του μοντέλου θα πρέπει να γίνει μία διαχείριση των τιμών αυτών. Εδώ υπάρχουν δύο επιλογές: η διαγραφή όσων rows (γραμμών) περιέχουν κενές τιμές ή το imputation των άδειων τιμών.

Το imputation μπορεί να οριστεί ως αποκατάσταση δεδομένων. Στην ουσία, είναι η διαδικασία που αντικαθιστούνται οι κενές τιμές από το σύνολο των δεδομένων με εκτιμήσεις. [10]

Για το imputation υπάρχουν πολλές επιλογές και θα εξεταστούν μερικές από αυτές. Αφού υλοποιηθούν οι διαφορετικοί τρόποι αντιμετώπισης των κενών τιμών, στη συνέχεια, θα εξεταστούν τα αποτελέσματα που προκύπτει από κάθε έναν.

4.4.α ΔΙΑΓΡΑΦΗ ΤΩΝ ΓΡΑΜΜΩΝ ΜΕ ΚΕΝΕΣ ΤΙΜΕΣ

Αρχικά, ο πρώτος τρόπος διαχείρισης των κενών τιμών είναι να διαγραφούν όσα rows έχουν κενές τιμές.

Υλοποιώντας αυτόν τον τρόπο διαγράφονται συνολικά 849659 γραμμές από το train dataset και 212323 γραμμές από το test dataset. Οι γραμμές που διαγράφονται είναι περισσότερες από τις γραμμές που παραμένουν καθώς παραμένουν 541096 γραμμές στο train dataset και 135366 γραμμές στο test dataset.

Αυτός ο τρόπος διαχείρισης των κενών τιμών δεν είναι ιδανικός και θα οδηγήσει σε κακές προβλέψεις. Έτσι, δε θα χρησιμοποιηθεί αυτός ο τρόπος για την εκπαίδευση και τη πρόβλεψη.

4.4.β IMPUTATION ΜΕ SIMPLE IMPUTER ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΤΗ ΠΕΡΙΟΔΟ ΤΟΥ ΜΗΝΑ

Ο imputation τρόπος που θα δοκιμαστεί είναι το imputation χρησιμοποιώντας τον Simple Imputer της βιβλιοθήκης «scikit-learn». [11] Με αυτόν θα δοκιμαστούν διάφορες στρατηγικές.

Συγκεκριμένα, θα πρέπει να δημιουργηθεί ένα νέο feature (στήλη) στο train dataset και στο test dataset με το όνομα «month_period». Αυτό το feature δημιουργείται με τη βοήθεια του feature «date» που περιέχει την ημερομηνία της συγκεκριμένης ημέρας. Στην ουσία με βάση τη στήλη «date» στη καινούρια στήλη «month_period» θα χωρίζονται οι μήνες στα τρία (Early, Mid, Late). Οι πρώτες 10 ημέρες του μήνα (1-11) χαρακτηρίζονται ως «Early», οι επόμενες 10 (11-20) ως «Mid» και οι τελευταίες (21-31) ως «Late». Για παράδειγμα, για ένα row, αν στη στήλη «date» υπάρχει η ημερομηνία «15/03/2020», τότε στη στήλη «month_period» που θα δημιουργηθεί θα δοθεί η τιμή «Mid March».

Ως αποτέλεσμα, θα γίνει imputation που θα βασίζεται σε δεδομένα που είναι ίδιας χρονικής περιόδου. Για παράδειγμα, για κενές τιμές που αφορούν μήνες του χειμώνα δε θα χρησιμοποιούνται δεδομένα του καλοκαιριού για να γεμίζουν κλπ, ώστε να υπάρχει μεγάλη ακρίβεια.

Θα πραγματοποιηθεί imputation με τρεις διαφορετικές στρατηγικές. Αυτές είναι οι εξής: [12]

- mean: είναι η average (μέσος όρος) τιμή της στήλης
- median: είναι η middle (μέση) τιμή της στήλης
- most frequent: είναι η πιο συχνά εμφανιζόμενη τιμή της στήλης

Το imputation θα γίνει σε συγκεκριμένες στήλες. Συγκεκριμένα, αυτές οι στήλες είναι οι εξής: «avg_temp_c», «min_temp_c», «max_temp_c», «precipitation_mm», «avg_wind_dir_deg», «avg_wind_speed_kmh» και «avg_sea_level_pres_hpa». Αυτές είναι οι στήλες που μπορεί να περιέχουν κενές τιμές και πρέπει να «γεμίσουν».

Θα γίνει imputation για το train και το test dataset, κάθε ένα και για τις τρεις στρατηγικές.

Υπάρχουν κι άλλα είδη imputation. Μερικά από αυτά είναι το «k-nearest neighbor imputation», το «regression imputation» κ.ά.

Στο επόμενο βήμα θα πρέπει να συγκριθούν τα αποτελέσματα των στρατηγικών που εφαρμόστηκαν για να αποφασιστεί ποια από αυτές είναι η καλύτερη.

5. ΕΚΠΑΙΔΕΥΣΗ ΕΝΟΣ ΒΑΣΙΚΟΥ ΜΟΝΤΕΛΟΥ

Αυτό που πρέπει να γίνει στη συνέχεια είναι να εκπαιδευτεί ένα βασικό μοντέλο πάνω στα datasets που πέρασαν από imputation. Σκοπός είναι να βγούνε τρία διαφορετικά αποτελέσματα (score) για τις τρεις διαφορετικές στρατηγικές του imputation, για να μπορεί να αποφασιστεί ποια από τις τρεις είναι η καλύτερη.

Αρχικά, αυτό που πρέπει να γίνει πριν την εκπαίδευση κάποιου μοντέλου είναι να κωδικοποιηθούν (encoding) οι στήλες των datasets που πέρασαν από imputation, οι

οποίες περιέχουν strings (κείμενο) και να μετατραπούν σε integers (αριθμούς). Αυτό πρέπει να γίνει, καθώς τα περισσότερα μοντέλα έχουν αυτή τη προϋπόθεση. Θα πρέπει να γίνει λοιπόν αυτή η κωδικοποίηση για τις στήλες: «season», «month_period» και «climate_type». Δημιουργούνται, λοιπόν, νέες στήλες με τα ονόματα «season_encoded», «month_period_encoded» και «climate_type_encoded», οι οποίες περιέχουν μόνο integers (αριθμούς).

Τώρα, πρέπει να οριστεί τα datasets που θα εκπαιδεύσουν το μοντέλο και τα datasets που θα το «τεστάρουν». Για την εκπαίδευση θα χρησιμοποιηθούν τα συγκεκριμένα datasets:

- X_train: είναι το train dataset που περιέχει τις εξής στήλες: «avg_temp_c», «min_temp_c», «max_temp_c», «precipitation_mm», «avg_wind_dir_deg», «avg_wind_speed_kmh», «avg_sea_level_pres_hpa», «season_encoded» και «month_period_encoded». Αυτές είναι οι στήλες που θα εκπαιδεύσουν το μοντέλο. Δε χρησιμοποιούνται στήλες όπως οι «city_name» ή «climate_type», καθώς τότε το μοντέλο θα εκπαιδευόταν λανθασμένα και θα είχε ως αποτέλεσμα ψευδή υψηλά score.
- y_train: είναι το train dataset που περιέχει μονάχα τη στήλη στόχου (target), τη στήλη, δηλαδή, που το μοντέλο θα πρέπει να κάνει προβλέψεις.
- X_test: αντίστοιχα το dataset με τις στήλες που υπάρχουν και στο «X_train», αλλά για το test dataset.
- y_test αντίστοιχα το dataset με τη στήλη που υπάρχει και στο «y_train», αλλά για το test dataset.

Αυτή η διαδικασία πρέπει να γίνει για όλα τα διαφορετικά datasets που προέκυψαν με τις διαφορετικές στρατηγικές των imputations. Αφού οριστούν αυτά τα datasets, μπορεί να εκπαιδευτεί το μοντέλο.

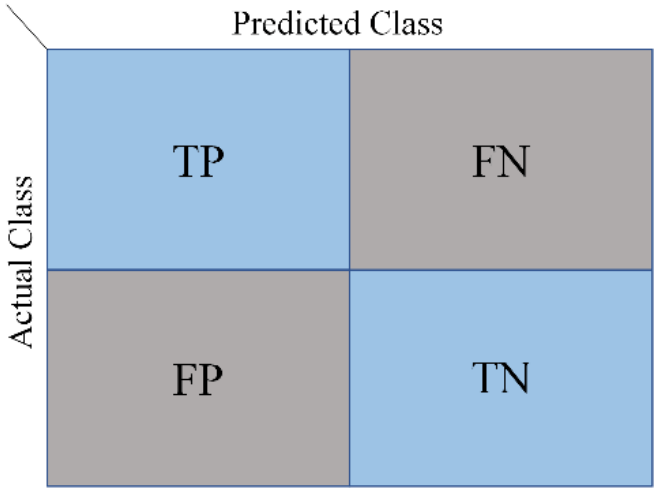
Το βασικότερο μοντέλο για την εκπαίδευση ενός classification προβλήματος (όπως αυτό που επιλύνεται) είναι το «Logistic Regression» (Λογιστική Παλινδρόμηση), για αυτό θα επιλεχτεί αυτό ως το base μοντέλο που θα εκπαιδευτεί. Η λογιστική παλινδρόμηση υπολογίζει τη πιθανότητα να συμβεί ένα γεγονός ή όχι. Για αυτό, συνήθως χρησιμοποιείται για binary προβλήματα (0 ή 1). Παρόλα αυτά, μπορεί να χρησιμοποιηθεί και για περισσότερες μεταβλητές (στη συγκεκριμένη τρεις), για να εκτιμηθεί ποιο dataset είναι καλύτερο. Η υλοποίηση στον κώδικα γίνεται με την LogisticRegression της βιβλιοθήκης «scikit-learn». [13]

Σκοπός σε αυτό το σημείο δεν είναι η επίτευξη ενός «ικανοποιητικού» score, αλλά να βρεθεί η καλύτερη μέθοδος και στρατηγική imputation και να χρησιμοποιηθεί αυτή για τη συνέχεια της επίλυσης του προβλήματος.

Η μετρική που επιλέγεται για τη μέτρηση του score είναι η F1 score. Η μετρική F1 μπορεί να ερμηνευτεί ως ένας αρμονικός μέσος όρος της ακρίβειας και της ανάκλησης. Η τιμή της κυμμαίνεται από το 0 εώς το 1 (0 είναι η χειρότερη περίπτωση και 1 η καλύτερη). Ο τύπος της F1 είναι ο εξής:

$$F1 = \frac{2 * TP}{2 * TP + FP + FN}$$

TP: ο αριθμός των True Positives, FN: ο αριθμός των False Negatives, FP: ο αριθμός των False Positives. [14]



Εικόνα 4: *Ένα τυπικό structure ενός confusion matrix [15]*

Στην παραπάνω εικόνα φαίνεται ένας confusion matrix όπου φαίνονται τα True Positives, False Positives, False Negatives και True Negative values. Στον κώδικα μπορεί να εισαχθεί η μετρική F1 από τη βιβλιοθήκη «scikit-learn».

Εκπαιδεύοντας και τεστάροντας το συγκεκριμένο μοντέλο όσες φορές όσα και τα διαφορετικά imputations, προκύπτουν τα παρακάτω F1 scores:

- Για τη στρατηγική SimpleImputer/mean imputation: $F1\ score = 0.5283$
- Για τη στρατηγική SimpleImputer/median imputation: $F1\ score = 0.5344$
- Για τη στρατηγική SimpleImputer/most frequent imputation: $F1\ score = 0.5403$

	F1 SCORE			Αλγόριθμος	Παράμετροι Αλγόριθμου	Features	Πείραμα
Imputation Strategy	mean	median	most frequent				
	0.5283	0.5344	0.5403	Logistic Regression	max_iter=1000, C=1.0, solver='saga'	Initial Features	baseline

Εικόνα 5: *Πίνακας με το σύνολο των πειραμάτων και τα F1 scores*

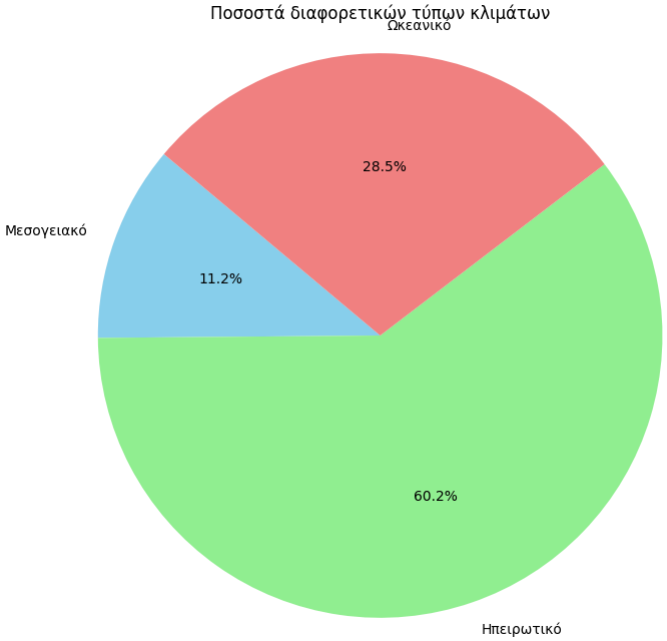
Καταλαβαίνει, λοιπόν, κανείς ότι και με τις τρεις διαφορετικές στρατηγικές το αποτέλεσμα είναι αρκετά παρόμοιο. Ελάχιστα καλύτερη είναι η στρατηγική most frequent, για αυτό το λόγο θα επιλεχτεί για τη δημιουργία μερικών plots.

6. ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΙΚΩΝ ΔΙΑΓΡΑΜΜΑΤΩΝ ΑΠΟ ΤΟ TRAIN DATASET

Θα δημιουργηθούν μερικά διαγράμματα (plots), για να παρουσιαστούν σχηματικά οι διαφορετικές μεταβλητές, οι σχέσεις μεταξύ τους κλπ. Σκοπός είναι να βγουν κάποια συμπεράσματα για το ποιες μεταβλητές συσχετίζονται μεταξύ τους, ποιες μεταβλητές παίζουν μεγαλύτερο ρόλο όσο αφορά το target value (climate_type) κλπ. Για τη δημιουργία των plots θα χρησιμοποιηθεί το train dataset και συγκεκριμένα αυτό που προέκυψε μετά από το imputation με τη στρατηγική most frequent, καθώς αυτή επιλέχτηκε, ως ελάχιστα καλύτερη, στο προηγούμενο βήμα.

6.1 ΠΟΣΟΣΤΑ ΔΙΑΦΟΡΕΤΙΚΩΝ ΤΥΠΩΝ ΚΛΙΜΑΤΩΝ (PIE CHART)

Μπορεί να δημιουργηθεί plot, καθαρά για πληροφοριακούς λόγους, για να γίνει αντιληπτό το σύνολο των δεδομένων για κάθε είδος κλιματικού τύπου. Θα χρησιμοποιηθεί ένα pie chart για τον σκοπό αυτό.



Εικόνα 6: *Pie plot με τα ποσοστά των διαφορετικών κλιματικών τύπων στο dataset*

Εύκολα παρατηρεί κανείς ότι το σύνολο του dataset περίεχει πόλεις που ανήκουν στην ηπειρωτική κλιματική ζώνη. Αυτό είναι λογικό, καθώς η πλειοψηφία των ευρωπαϊκών χωρών βρίσκονται σε αυτή τη ζώνη.

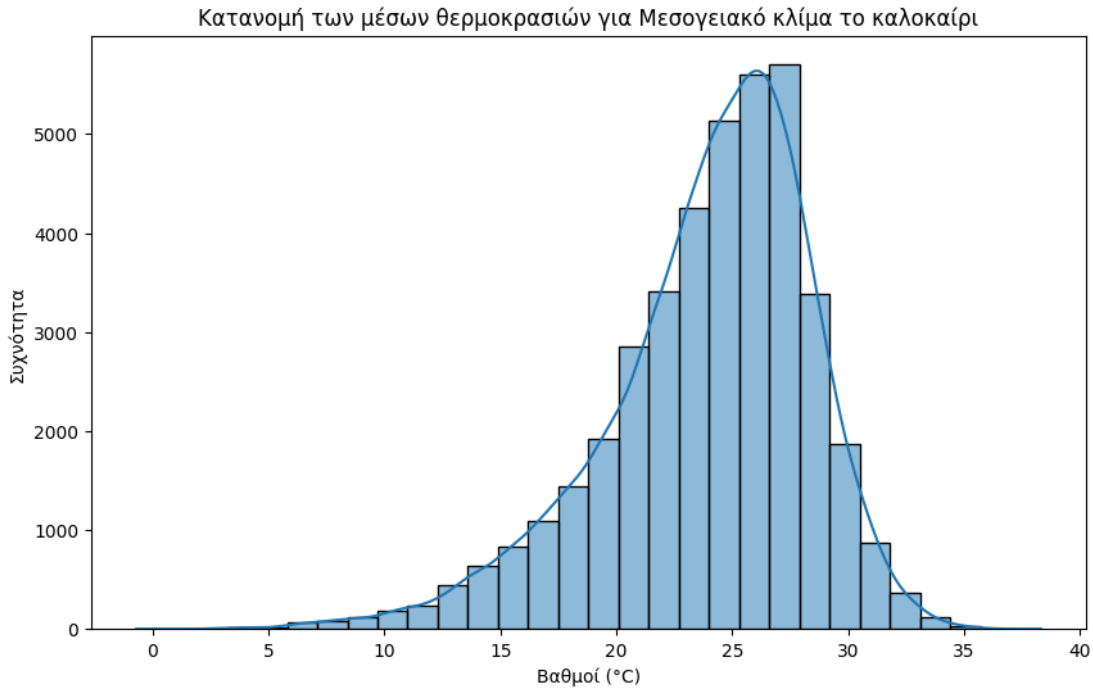
6.2 ΔΙΑΓΡΑΜΜΑ ΚΑΤΑΝΟΜΗΣ ΚΑΠΟΙΟΥ FEATURE (DISTRIBUTION PLOT)

Από ένα distribution plot μπορούν να βγουν αρκετά συμπεράσματα. Συγκεκριμένα, από ένα τέτοιο διάγραμμα μπορεί κάποιος να καταλάβει αν για ένα συγκεκριμένο feature υπάρχουν ακραίες τιμές ή ενδεχομένως λανθασμένες.

Τέτοια διαγράμματα μπορούν να γίνουν για οποιαδήποτε στήλη ενός dataset και να βγουν τα αντίστοιχα συμπεράσματα.

Θα σχεδιαστούν ορισμένα από τα πιθανά distribution plots και θα αναλυθούν μερικά από αυτά:

6.2.α ΚΑΤΑΝΟΜΗ ΤΩΝ ΜΕΣΩΝ ΘΕΡΜΟΚΡΑΣΙΩΝ ΤΟΥ ΚΑΛΟΚΑΙΡΙΟΥ ΣΤΟ ΜΕΣΟΓΕΙΑΚΟ ΚΛΙΜΑ

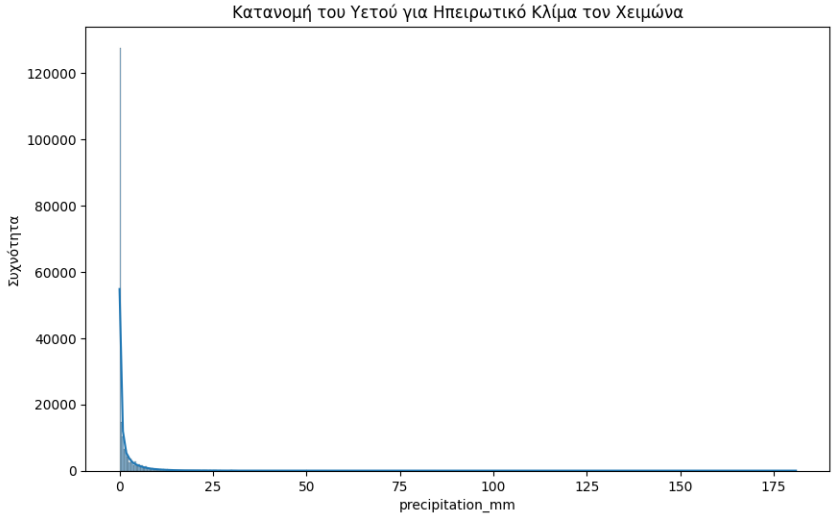


Εικόνα 7: *Distribution plot για τις μέσες θερμοκρασίες που παρουσιάζονται την εποχή του καλοκαιριού στη Μεσογειακή Κλιματική Ζώνη*

Παρατηρώντας αυτό το διάγραμμα μπορούν να προκύψουν ορισμένα συμπεράσματα. Φαίνεται ο μέσος όρος των θερμοκρασιών, οι ακραίες τιμές και γενικότερα φαίνεται αν η κατανομή είναι κεντραρισμένη ή αν εκτείνεται προς κάποια άλλη κατεύθυνση. Συγκεκριμένα, παρατηρεί κανείς ότι υπάρχουν ορισμένες τιμές οι οποίες ενδεχομένως να είναι λανθασμένες ή εξαιρέσεις (π.χ. μέση θερμοκρασία κάτω των 10 βαθμών Κελσίου για την εποχή του καλοκαιριού σε Μεσογειακό κλίμα) οι οποίες στη συνέχεια μπορούν να γίνουν normalized, ώστε το μοντέλο να οδηγείται σε πιο ακριβείς προβλέψεις.

Στη συνέχεια, θα εξεταστεί το distribution plot κάποιου feature που περιείχε αρχικά πολλές κενές τιμές και που έγινε imputed.

6.2.β ΚΑΤΑΝΟΜΗ ΤΟΥ ΜΕΣΟΥ ΥΕΤΟΥ ΓΙΑ ΤΟΥ ΧΕΙΜΩΝΑ ΓΙΑ ΤΟ ΗΠΕΙΡΩΤΙΚΟ ΚΛΙΜΑ



Εικόνα 8: *Distribution plot για τον μέσο υετό για την εποχή του χειμώνα σε ηπειρωτικό κλίμα*

Αυτό που φαίνεται παραπάνω βγάζει νόημα, καθώς εξετάζεται το dataset που προέκυψε από τον τρόπο most frequent για το imputation. Αυτό σημαίνει ότι η πιο συχνά εμφανιζόμενη τιμή ήταν το 0 και είχε ως αποτέλεσμα το παραπάνω.

Έτσι, παρόλο που το dataset που προέκυψε από most frequent στρατηγική imputation σημείωσε καλύτερο score, δε σημαίνει ότι αντιστοιχεί σε καλής ποιότητας dataset.

Στη συνέχεια, θα δοκιμαστούν κι άλλα μοντέλα και τεχνικές για να αποφευχθούν οι ακραίες τιμές και να προκύψουν καλύτερες προβλέψεις.

7. OPTIMIZATION ΤΟΥ DATASET

Προτού δοκιμαστούν κι άλλα μοντέλα, ειδικά για classification προβλήματα, θα γίνει προσπάθεια για περαιτέρω optimization των dataset.

7.1 ΕΛΕΓΧΟΣ ΤΩΝ ΑΚΡΑΙΩΝ ΤΙΜΩΝ (OUTLIERS)

Όπως φάνηκε στα παραπάνω διαγράμματα, ενδέχεται το dataset που εξετάζεται να έχει ακραίες τιμές. Σε αυτό το σημείο θα πρέπει να ελεγχθούν αυτές και να αποφασιστεί αν ο αριθμός τους είναι πολύ μεγάλος, με αποτέλεσμα να επηρεάζονται οι προβλέψεις του μοντέλου.

Αυτός ο έλεγχος θα γίνει στα imputed datasets και των τριών στρατηγικών, για να προκύψουν τα κατάλληλα συμπεράσματα. Αν το ποσοστό των outliers τιμών είναι μεγάλο τότε μια imputation μέθοδος θα χαρακτηριστεί ακατάλληλη.

Για να βρεθεί αυτό το ποσοστό των outliers θα χρησιμοποιηθεί το zscore της βιβλιοθήκης «scipy.stats» [16]. Χρησιμοποιείται το zscore (standard score – τυπική τιμή) για να βρεθεί το ποσοστό των τιμών ενός feature που αποκλίνουν από τον μέσο όρο της στήλης αυτής.

Βρίσκοντας τα zscore για κάθε dataset προκύπτουν τα εξής ποσοστά ακραίων τιμών:

Στρατηγική	**Outliers στο Train**	**Ποσοστό % των Outliers στο Train**	**Outliers στο Test**	**Ποσοστό % των Outliers στο Test**
mean	118316	1.22%	29950	1.23%
median	119439	1.23%	30372	1.25%
most frequent	114683	1.18%	29288	1.20%

Τα ποσοστά των ακραίων τιμών που προέκυψαν θεωρούνται μικρά και δεν χρειάζεται να γίνει κάποια παρέμβαση για τη διαχείρισή τους.

7.2 ΔΗΜΙΟΥΡΓΙΑ ΝΕΩΝ ΜΕΤΑΒΛΗΤΩΝ (FEATURE ENGINEERING)

Από τις υπάρχουσες μεταβλητές (features) μπορούν να προκύψουν νέες μεταβλητές που θα προέρχονται από πράξεις ή ομαδοποιήσεις μεταξύ των αρχικών μεταβλητών. Ακόμη, μπορούν να δημιουργηθούν binary (0 ή 1) μεταβλητές που να υποδηλώνουν τη πραγματοποίηση ή όχι ενός γεγονότος. Αυτές οι νέες μεταβλητές δημιουργούνται για να βοηθήσουν το μοντέλο να προχωρήσει σε καλύτερες προβλέψεις. Δεν είναι σίγουρο ότι αυτές οι μεταβλητές θα προσφέρουν θετικά, αλλά θα γίνει η προσπάθεια και ύστερα την παρουσίαση των αποτελεσμάτων θα παρθούν οι κατάλληλες αποφάσεις. [17]

7.2.α ΜΕΤΑΒΛΗΤΗ «TEMPERATURE RANGE»

Η πρώτη μεταβλητή που μπορεί να δημιουργηθεί με βάση τις ήδη υπάρχουσες είναι η μεταβλητή «temp_range». Αυτό το feature θα εκφράζει το εύρος της ημερήσιας θερμοκρασίας. Συγκεκριμένα, θα προκύπτει από την εξής πράξη:

$$temp\_range = \max\_temp - \min\_temp$$

7.2.β ΜΕΤΑΒΛΗΤΗ «WIND VECTOR»

Η επόμενη μεταβλητή που μπορεί να δημιουργηθεί είναι η μεταβλητή «wind_vector». Αντί να υπάρχουν δύο features που αφορούν τον αέρα μπορούν να ενωθούν και να προκύψει αυτή η νέα μεταβλητή. Ο τρόπος που θα ενωθούν θα είναι με κάποια τριγωνομετρική συνάρτηση, για να προκύψει ένα διάνυσμα από τα features «avg_wind_deg» και «avg_wind_speed_kmh». Συγκεκριμένα, θα βρεθεί η τιμή x και η τιμή y του αέρα της κάθε γραμμής, με βάση τα δύο features «avg_wind_deg» και «avg_wind_speed_kmh», χρησιμοποιώντας το ημίτονο και συνημίτονο αντίστοιχα της γωνίας «avg_wind_deg» (σε radians) – πολλαπλασιασμένα με το μέτρο «avg_wind_speed_kmh».

Για τη μετατροπή των μοιρών των γωνιών σε radians, αλλά και για τον πολλαπλασιασμό χρησιμοποιείται η βιβλιοθήκη «numpy». [18]

Αναγκαστικά, οι τιμές x και y της κάθε γραμμής θα πρέπει να αποθηκευτούν σε ξεχωριστά features, καθώς κάθε feature για να τρέξει σε οποιοδήποτε μοντέλο πρέπει να είναι αριθμός και το «wind_vector» θα ήταν ένα διάνυσμα της μορφής [x, y] και δε θα μπορούσε να χρησιμοποιηθεί στα μοντέλα. Έτσι, θα δημιουργηθούν δύο νέα features, το «wind_x» και το «wind_y».

7.2.γ BINARY ΜΕΤΑΒΛΗΤΗ «RAINED»

Μία ακόμα μεταβλητή που μπορεί να δημιουργηθεί είναι η binary μεταβλητή «rained». Η μεταβλητή αυτή θα παίρνει τη τιμή 1 αν τη συγκεκριμένη μέρα έβρεξε ($participation\_mm > 0$) ή τη τιμή 0 εάν δεν έβρεξε ($participation\_mm = 0$). Αυτό ισχύει γιατί αν ο υετός είναι πάνω από τη μηδενική τιμή, σημαίνει ότι έβρεξε τη συγκεκριμένη ημέρα.

Οι μεταβλητές αυτές θα δημιουργηθούν σε όλα τα datasets που πέρασαν από imputation - και με τις τρεις στρατηγικές.

Ενδεχομένως να μπορούσαν να βρεθούν κι άλλες νέες μεταβλητές με βάση τις ήδη υπάρχουσες για να παραχθούν.

7.3 ΕΠΑΝΕΞΕΤΑΣΗ ΤΩΝ F1 SCORES

Αφού δημιουργήθηκαν οι νέες μεταβλητές θα ξαναεκπαιδευτεί το βασικό μοντέλο που εκπαιδεύτηκε παραπάνω και θα γίνει η βαθμολογία τους ξανά με βάση το F1 score, για να προκύψει αν βελτιώθηκε ή όχι εξαιτίας των νέων features. Το μοντέλο που χρησιμοποιείται είναι το ίδιο βασικό μοντέλο «Logistic Regression» που χρησιμοποιήθηκε παραπάνω. Κάνοντας την εκπαίδευση και τρέχοντας το μοντέλο για τα νέα datasets που περιέχουν τα νέα features προκύπτουν τα εξής αποτελέσματα:

- Για τη στρατηγική SimpleImputer/mean imputation με τα νέα features: $F1\ score = 0.5302$, δηλαδή αύξηση κατά 0.0019.

- Για τη στρατηγική SimpleImputer/median imputation με τα νέα features: $F1\ score = 0.535$, δηλαδή αύξηση κατά 0.0006.
- Για τη στρατηγική SimpleImputer/most frequent imputation με τα νέα features: $F1\ score = 0.5499$, δηλαδή αύξηση κατά 0.0096.

	F1 SCORE			Αλγόριθμος	Παράμετροι Αλγόριθμου	Features	Πείραμα
Imputation Strategy	mean	median	most frequent				
	0.5283	0.5344	0.5403	Logistic Regression	max_iter=1000, C=1.0, solver='saga'	Initial Features	baseline
	0.5302	0.535	0.5499	"	"	Initial Features + όλα όσα φτιάχτηκαν	Feature Engineering

Εικόνα 9: *Πίνακας με το σύνολο των πειραμάτων και τα F1 scores*

Άρα, με τη δημιουργία των νέων features, παρατηρείται έστω και μία μικρή αύξηση στα F1 scores και των τριών στρατηγικών.

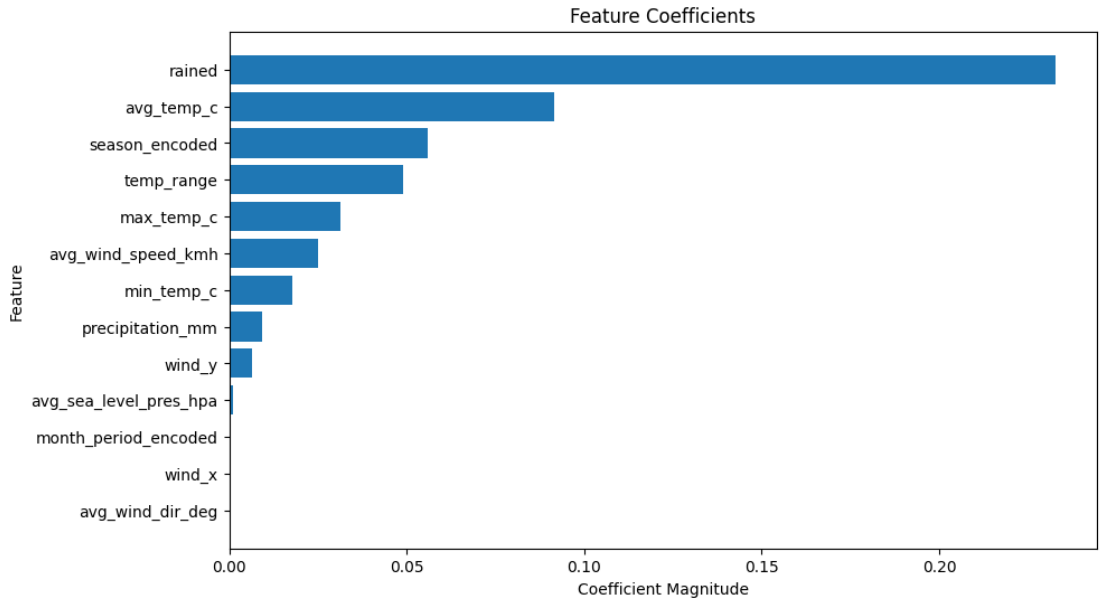
Τα features που δημιουργήθηκαν, δε βελτίωσαν σε μεγάλο βαθμό τις προβλέψεις του βασικού μοντέλου. Παρόλα αυτά, θα κρατηθούν, καθώς στη συνέχεια θα γίνει Feature Importance όπου θα εξεταστεί η σημαντικότητα κάθε μεταβλητής.

7.4 FEATURE IMPORTANCE

Με το feature importance θα βρεθούν ποιες μεταβλητές δε συνεισφέρουν στο μοντέλο με σκοπό να αποκλειστούν από την εκπαίδευση.

Αρχικά θα γίνει ένα διάγραμμα με το feature importance για το συγκεκριμένο βασικό μοντέλο που δοκιμάστηκε. Στη συνέχεια θα βρεθούν τα σημαντικότερα features με μεθόδους feature selection.

Το διάγραμμα θα γίνει για το train dataset για τον most frequent imputation τρόπο και με τα νέα features υπόψιν. Το διάγραμμα είναι το παρακάτω:



Εικόνα 10: *Feature Importance plot για το μοντέλο «Logistic Regression»*

Στο μοντέλο «Logistic Regression» το feature importance μεταφράζεται σε «Feature Coefficients», αλλά το νόημα είναι παρόμοιο.

Όπως φαίνεται και από το διάγραμμα, το binary feature «rained» που δημιουργήθηκε στο προηγούμενο βήμα παίζει σημαντικό ρόλο για το μοντέλο

«Logistic Regression». Ακολουθούν σε συνεισφορά τα features «avg_temp_c», «season_encoded» και ακολουθεί το επίσης νέο feature «temp_range». Έτσι, το feature engineering, που έγινε στο προηγούμενο βήμα, χαρακτηρίζεται επιτυχημένο.

Παρατηρείται εύκολα ότι τα features «avg_wind_dir_deg», «wind_x», «month_period_encoded» και «avg_sea_level_pres_hpa» δε βοηθάνε το μοντέλο καθόλου στις προβλέψεις.

Παρόλα αυτά, κάθε μοντέλο λειτουργεί διαφορετικά και δεν μπορεί να βασιστεί η συνέχεια της επίλυσης σε αυτό το διάγραμμα, καθώς αυτό το διάγραμμα αφορά το μοντέλο «Logistic Regression».

Θα πρέπει, λοιπόν, να χρησιμοποιηθεί μια γενική αντιμετώπιση για την εύρεση των σημαντικών features, ώστε τα features που δε συνεισφέρουν θετικά να αποκλειστούν. Αυτή η αντιμετώπιση που θα ακολουθήσει, ονομάζεται feature selection.

7.5 FEATURE SELECTION

Η feature selection (επιλογή μεταβλητών) επιλέγει με αυτόματο τρόπο τις καλύτερες μεταβλητές ενός dataset και χρησιμοποιεί αυτές για την εκπαίδευση και το τεστάρισμα, αντί για όλο το σύνολο. [19]

Η επιλογή μεταβλητών μπορεί να γίνει με πολλές μεθόδους. Μία από τις κλασσικότερες μεθόδους που θα δοκιμαστεί είναι η «Recursive Feature Elimination».

7.5.α FEATURE SELECTION ΜΕ RECURSIVE FEATURE ELIMINATION (RFE)

Αυτή η μέθοδος αφαιρεί τις λιγότερο σημαντικές μεταβλητές ενός dataset μέχρι να φτάσει τον αριθμό των επιθυμητών features. Ο επιθυμητός αριθμός των τελικών features απαιτείται να οριστεί από τον χρήστη στις παραμέτρους της μεθόδου – αυτή είναι και η αδυναμία αυτής της μεθόδου.

Για να χρησιμοποιηθεί η μέθοδος «Recursive Feature Elimination» θα χρησιμοποιηθεί η RFE από την βιβλιοθήκη «scikit-learn». [20] Επίσης, θα χρειαστεί και ένας εκτιμητής (estimator) για να τρέξει η μέθοδος RFE. Θα χρησιμοποιηθεί ο RandomForestClassifier από την βιβλιοθήκη «scikit-learn» και θα επεξηγηθεί η λειτουργία του στη συνέχεια. [21]

Αρχικά, ο RandomForestClassifier είναι ένας αλγόριθμος μηχανικής μάθησης που χρησιμοποιείται για προβλήματα ταξινόμησης. Συγκεκριμένα, δημιουργεί πολλά δέντρα αποφάσεων (decision trees) και συμπυκνώνει τα αποτελέσματα για να οδηγηθεί σε ακριβείς προβλέψεις.

Ο RandomForestClassifier χρειάζεται στη προκειμένη περίπτωση, γιατί θα το χρησιμοποιήσει η Recursive Feature Elimination για να αποφασίσει ποια features είναι χρήσιμα και ποια όχι.

Για τον RandomForestClassifier θα χρησιμοποιηθούν οι εξής παράμετροι: «n_estimators=100, random_state=42». Για την RFE θα χρησιμοποιηθούν οι εξής παράμετροι: «n_features_to_select=8, step=1». Αυτό σημαίνει ότι η επιλογή για τον τελικό αριθμό επιθυμητών features είναι 8 (επιλέγεται τυχαία – με το feature importance που προέκυψε στο παραπάνω διάγραμμα υπόψιν). Άρα στο τέλος, θα μείνουν τα οκτώ features που έχουν τη καλύτερη συνεισφορά, σύμφωνα πάντα με τον RandomForestClassifier. Θα τρέξει η μέθοδος RFE για 100 χιλιάδες γραμμές για κάθε

dataset και όχι για όλες, καθώς ο αριθμός όλων των γραμμών είναι πολύ μεγάλος, και το σφάλμα που θα προκύψει δε θεωρείται σημαντικό.

Αφού τρέξει η RFE με τη βοήθεια του RandomForestClassifier θα γίνει ξανά εκπαίδευση και τεστάρισμα των dataset, για να προκύψουν τα νέα F1 scores.

Για κάθε στρατηγική imputation θεωρητικά μπορεί να προκύψουν 8 διαφορετικά βέλτιστα features, καθώς η RFE θα τρέξει για κάθε μία ξεχωριστά.

Παρόλα αυτά, και για τις τρεις στρατηγικές προκύπτουν τα ίδια 8 «σημαντικότερα» features. Συγκεκριμένα, αυτά τα features είναι τα εξής:: «avg_temp_c», «min_temp_c», «max_temp_c», «avg_sea_level_pres_hpa», «temp_range», «avg_wind_speed_kmh», «wind_x» και «wind_y». Δηλαδή τα features που απορρίφθηκαν είναι τα «precipitation_mm», «avg_wind_dir_deg», «month_period_encoded», «season_encoded» και «rained»

Τα νέα F1 scores των datasets είναι τα εξής:

- Για τη στρατηγική SimpleImputer/mean imputation με τα 8 καλύτερα features της RFE: $F1\ score = 0.5297$, δηλαδή μείωση κατά 0.0005.
- Για τη στρατηγική SimpleImputer/median imputation με τα 8 καλύτερα features της RFE: $F1\ score = 0.5319$, δηλαδή μείωση κατά 0.0031.
- Για τη στρατηγική SimpleImputer/most frequent imputation με τα 8 καλύτερα features της RFE: $F1\ score = 0.5426$, δηλαδή μείωση κατά 0.0073.

	F1 SCORE			Αλγόριθμος	Παράμετροι Αλγόριθμου	Features	Πείραμα	Παράμετροι Πειράματος
Imputation Strategy	mean	median	most frequent					
	0.5283	0.5344	0.5403	Logistic Regression	max_iter=1000, C=1.0, solver='saga'	Initial Features	baseline	
	0.5302	0.535	0.5499	"	"	Initial Features + όλα όσα φτιάχτηκαν	Feature Engineering	
	0.5297	0.5319	0.5426	"	"	Τα 8 καλύτερα features μετά την RFE	Recursive Feature Elimination (RFE)	n_estimators=100, random_state=42, n_features_to_select=8, step=1

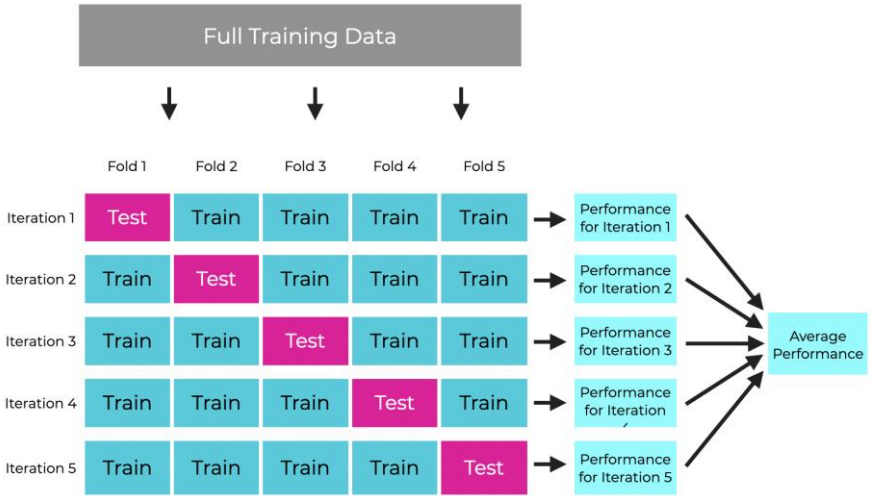
Εικόνα 9: *Πίνακας με το σύνολο των πειραμάτων και τα F1 scores*

Άρα, με τη μέθοδο Recursive Feature Elimination, χρησιμοποιώντας το βασικό μοντέλο, προέκυψαν χειρότερα αποτελέσματα. Ενδεχομένως ευθύνεται το μοντέλο που χρησιμοποιείται, καθώς η Recursive Feature Elimination απευθύνεται, κυρίως, σε αλγόριθμους που είναι φτιαγμένοι για classification προβλήματα. Σε επόμενο βήμα θα δοκιμαστούν τέτοια μοντέλα, με σκοπό να αυξηθεί το F1 score.

Προτού δοκιμαστούν, όμως, νέα μοντέλα θα δοκιμαστεί κι άλλη μία μέθοδος για feature selection, η «Recursive Feature Elimination with Cross Validation».

7.5.β FEATURE SELECTION ΜΕ RECURSIVE FEATURE ELIMINATION WITH CROSS VALIDATION (RFECV)

Η μέθοδος αυτή είναι μία πιο προηγμένη μορφή της προηγούμενης μεθόδου RFE που εξετάστηκε. Η διαφορά είναι ότι σε αυτή τη μέθοδο γίνεται cross-validation. Συγκεκριμένα, κάθε dataset χωρίζεται σε training και validation sets πολλές φορές (folds) για διαφορετικό αριθμό και συνδυασμό features. Μετά, συγκρίνονται μεταξύ τους τα αποτελέσματα του cross-validation για να αποφαστεί ποιος αριθμός features και συνδυασμός παράγει τα καλύτερα αποτελέσματα, με σκοπό την επίτευξη καλύτερων προβλέψεων από τα μοντέλα που θα εκπαιδευτούν.



Εικόνα 10: Διάγραμμα ενός παραδείγματος cross-validation με 5 folds [22]

Το πλεονέκτημα της είναι ότι ο χρήστης δε χρειάζεται να πάρει την απόφαση για το πόσα θα είναι τα τελικά features, καθώς αποφασίζει μόνη της τον optimal αριθμό.

Για να υλοποιηθεί η RFECV μέθοδος, θα χρησιμοποιηθεί η RFECV της βιβλιοθήκης «scikit-learn». [23] Επίσης, θα χρειαστεί πάλι ο RandomForestClassifier για να «συνεργαστεί» με την RFECV, με παρόμοιο τρόπο όπως με την RFE προηγουμένως. Για τον RandomForestClassifier θα χρησιμοποιηθούν οι ίδιες παράμετροι με πριν. Για την RFECV θα χρησιμοποιθούν οι εξής παράμετροι: «cv=5» και «scoring='accuracy'». Το cv είναι ο αριθμός των folds που θα πραγματοποιηθούν και 5 είναι ένας αρκετά συνηθισμένος αριθμός που χρησιμοποιείται. Όσο αφορά τη παράμετρο scoring, είναι η μετρική που θα χρησιμοποιηθεί για να υπολογιστεί ο καλύτερος αριθμός και συνδυασμός των features. Επιλέγεται η μετρική «accuracy», γιατί είναι η πιο συνηθισμένη για αυτή τη δουλειά.

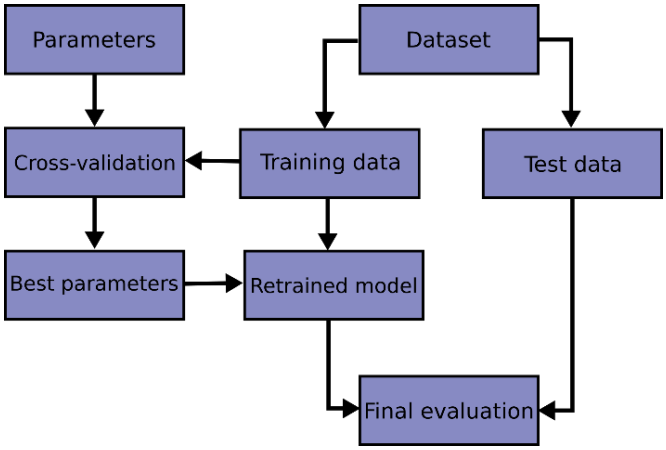
Ο τύπος της accuracy είναι ο εξής:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

Για classifications υπάρχει και ο παρακάτω τύπος που αφορά τις μετρικές TP, TN, FP και FN που εξηγήθηκαν παραπάνω (βλ. Εικόνα 4):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$ [24]

Αυτό που θα προκύψει για κάθε ένα dataset (ένα για κάθε στρατηγική imputation) είναι ο optimal αριθμός και συνδυασμός των μεταβλητών. Ύστερα, θα χρησιμοποιηθούν τα νέα optimized features, για κάθε dataset, για να ξαναυπολογιστούν τα F1 scores.



Εικόνα 11: Διάγραμμα που δείχνει πως λειτουργεί μία μέθοδος cross-validation [25]

Η μέθοδος RFECV, όπως και η RCE, θα χρησιμοποιήσει 100 χιλιάδες γραμμές από κάθε dataset και όχι όλο το σύνολο. Παρακάτω θα παρουσιαστούν τα αποτελέσματα που προέκυψαν από την μέθοδο RFECV.

Τα features που απορρίφθηκαν είναι τα εξής:

- Για τη στρατηγική mean και median: «month_period_encoded»
- Για τη στρατηγική most_frequent: «month_period_encoded» και «rained»

Παρατηρείται, δηλαδή, ότι η RFECV θεωρεί σχεδόν όλα τα features σημαντικά και κάνει eliminate ελάχιστα. Τα νέα F1 scores που προκύπτουν είναι τα εξής:

- Για τη στρατηγική SimpleImputer/mean imputation με τα καλύτερα features της RFECV: $F1\ score = 0.5299$.
- Για τη στρατηγική SimpleImputer/median imputation με τα καλύτερα features της RFECV: $F1\ score = 0.5342$.
- Για τη στρατηγική SimpleImputer/most frequent imputation με τα καλύτερα features της RFECV: $F1\ score = 0.5439$.

	F1 SCORE			Αλγόριθμος	Παράμετροι Αλγόριθμου	Features	Πείραμα	Παράμετροι Πειράματος
Imputation Strategy	mean	median	most frequent					
	0.5283	0.5344	0.5403	Logistic Regression	max_iter=1000, C=1.0, solver='saga'	Initial Features	baseline	
	0.5302	0.535	0.5499	"	"	Initial Features + όλα όσα φτιάχτηκαν	Feature Engineering	
	0.5297	0.5319	0.5426	"	"	Τα 8 καλύτερα features μετά την RFE	Recursive Feature Elimination (RFE)	n_estimators=100, random_state=42, n_features_to_select=8, step=1
	0.5299	0.5342	0.5439	"	"	Τα καλύτερα features μετά την RFECV	Recursive Feature Elimination with Cross Validation (RFECV)	n_estimators=100, random_state=42, step=1, cv=5, scoring='accuracy'

Εικόνα 12: *Πίνακας με το σύνολο των πειραμάτων και τα F1 scores*

Τα νέα F1 scores είναι ελάχιστα καλύτερα από τα F1 scores που προέκυψαν μετά το τρέξιμο της RFE και ελάχιστα χειρότερα από την εκπαίδευση του μοντέλου με όλα τα features. Όπως αναφέρθηκε και παραπάνω, αυτό οφείλεται ότι δεν έχει χρησιμοποιηθεί ακόμα κάποιο μοντέλο που εξειδικεύεται σε classification προβλήματα. Αυτό είναι το επόμενο βήμα.

8. ΔΟΚΙΜΗ ΔΙΑΦΟΡΕΤΙΚΩΝ ΜΟΝΤΕΛΩΝ

Θα δοκιμαστούν νέα μοντέλα, τα οποία ειδικεύονται σε προβλήματα ταξινόμησης, με σκοπό τη βελτίωση των F1 scores και κατ' επέκταση των προβλέψεων. Για κάθε μοντέλο θα εξεταστούν όλες οι στρατηγικές (mean, median, most frequent) αλλά και για κάθε μία από αυτές θα εξεταστούν οι διαφορετικοί συνδυασμοί των features τους. Συγκεκριμένα, θα εξεταστούν με τα αρχικά features των dataset, με όλα τα features (συμπεριλαμβάνοντας και τα features που δημιουργήθηκαν από το feature engineering), με τα 8 καλύτερα features που προέκυψαν από τη μέθοδο «Recursive Feature Elimination» και με τα καλύτερα features που προέκυψαν από τη μέθοδο «Recursive Feature Elimination with Cross Validation» (κάθε στρατηγική στα features που της αντιστοιχούν.

Έτσι, θα δοθεί μια καθαρή εικόνα για το πως επηρεάζονται τα F1 scores και κατ' επέκταση η ακρίβεια των προβλέψεων, χρησιμοποιώντας μοντέλα ειδικευμένα σε προβλήματα ταξινόμησης και διαφορετικούς συνδυασμούς των features που προτάθηκαν παραπάνω.

8.1 K-NEAREST NEIGHBORS CLASSIFIER

Για την βελτιστοποίηση του μοντέλου, μπορεί να χρησιμοποιηθεί ο αλγόριθμος «k-Nearest Neighbors Classifier», ο οποίος αποτελεί τεχνική μηχανικής μάθησης που χρησιμοποιείται για την επίλυση προβλημάτων ταξινόμησης.

Ο τρόπος που λειτουργεί αυτό το μοντέλο είναι να κρατάει το training dataset αποθηκευμένο και για κάθε νέο δεδομένο που εισέρχεται να υπολογίζει την απόσταση από αυτό και το αντίστοιχο στο training dataset. Στη συνέχεια, βρίσκει τους k κοντινότερους γείτονες για το νέο δεδομένο, βασιζόμενο στις αποστάσεις που υπολογίστηκαν. Τη παράμετρο k την ορίζει ο χρήστης αυθαίρετα και είναι το πόσοι γείτονες θα λαμβάνονται υπόψιν.

Η κλάση του νέου δεδομένου προσδιορίζεται με βάση την κλάση που εμφανίζεται συχνότερα μεταξύ των k κοντινότερων γειτόνων (πλειοψηφική ψήφος). Η επιλογή της παραμέτρου k και της μετρικής απόστασης είναι σημαντική για την απόδοση του μοντέλου. [26]

Για να υλοποιηθεί αυτό το μοντέλο θα χρησιμοποιηθεί ο KNeighborsClassifier της βιβλιοθήκης «scikit-learn». [27]

Όπως προαναφέρθηκε, κάθε τεχνική θα δοκιμαστεί στις τρεις διαφορετικές στρατηγικές imputation που πραγματοποιήθηκαν και κάθε μία από αυτές με τους διαφορετικούς συνδυασμούς των features που αναλύθηκαν.

Ο KNeighborsClassifier δέχεται ορισμένες παραμέτρους. Η σημαντικότερη παράμετρος είναι το k που θα οριστεί με τη default τιμή 5. Όσο για τις υπόλοιπες παραμέτρους, θα πάρουν κι αυτές τις default τιμές, οι οποίες φαίνονται στην «Εικόνα 13».

Παρακάτω φαίνονται τα διαφορετικά F1 scores για τις διαφορετικές στρατηγικές και συνδυασμούς set.

	F1 SCORE			Αλγόριθμος	Παράμετροι Αλγόριθμου	Features	Πείραμα
Imputation Strategy	mean	median	most frequent				
	0.6278	0.6183	0.5918	K-nearest Neighbors Classifier	n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None	Initial Features	Δοκιμή του μοντέλου K-nearest Neighbors Classifier με τα αρχικά features
	0.6315	0.6206	0.593	"	"	Initial Features + όλα όσα φτιάχτηκαν	Δοκιμή του μοντέλου K-nearest Neighbors Classifier με τα features που φτιάχτηκαν
	0.6312	0.631	0.5954	"	"	Τα 8 καλύτερα features μετά την RFE	Δοκιμή του μοντέλου K-nearest Neighbors Classifier μετά την RFE
	0.6589	0.6505	0.6166	"	"	Τα καλύτερα features μετά την RFECV	Δοκιμή του μοντέλου K-nearest Neighbors Classifier μετά την RFECV

Εικόνα 13: *Πίνακας με τα διαφορετικά F1 scores για το μοντέλο «KNeighborsClassifier»*

Όπως φαίνεται, υπάρχει σημαντική αύξηση στα F1 scores σε σύγκριση με το βασικό μοντέλο «Logistic Regression» που αναλύθηκε παραπάνω. Αυτό ήταν αναμενόμενο, καθώς όπως προαναφέρθηκε, το «Logistic Regression» είναι ένα μοντέλο γενικής φύσης, ενώ από εδώ και πέρα εξετάζονται τεχνικές ειδικά για classification προβλήματα (όπως στη προκειμένη περίπτωση ο KNeighborsClassifier).

Όσο αφορά τις διαφορετικές στρατηγικές, παρατηρείται ότι τα καλύτερα αποτελέσματα φέρνει η mean, με την median να ακολουθεί και την most frequent να έρχεται τελευταία. Αυτό είναι διαφορετικό από αυτό που προέκυψε με το βασικό μοντέλο, όπου η most frequent ήταν η πιο ακριβής μέθοδος.

Όσο αφορά τον συνδυασμό των features το καλύτερο αποτέλεσμα, όπως και ήταν αναμενόμενο, προέρχεται από τα optimized features που προέκυψαν από τη μέθοδο «Recursive Feature Reduction with Cross Validation», η οποία επέλεξε αυτόματα τον αριθμό και τον συνδυασμό των μεταβλητών, για να προκύψει το καλύτερο δυνατό αποτέλεσμα. Ακολουθεί σε ακρίβεια ο συνδυασμός όλων των μεταβλητών, μαζί και με αυτών που δημιουργήθηκαν με το Feature Engineering. Ελάχιστα χειρότερο αποτέλεσμα φέρνει ο συνδυασμός των οκτώ καλύτερων features που προέκυψαν από την μέθοδο «Recursive Feature Elimination». Από ότι φαίνεται αυτή η μέθοδος δεν ωφέλησε καθόλου, καθώς οδηγεί σε χειρότερα αποτελέσματα. Τελευταίος σε scores είναι ο συνδυασμός των αρχικών features, όπως και είναι λογικό, καθώς δεν έχει γίνει κάποιο optimization.

Θα δοκιμαστούν κι άλλα classification μοντέλα για να βρεθεί το καλύτερο για το συγκεκριμένο πρόβλημα.

8.2 RANDOM FOREST CLASSIFIER

Το επόμενο μοντέλο που θα εξεταστεί είναι το «Random Forest Classifier». Αυτός ο αλγόριθμος εξετάστηκε ήδη, καθώς χρησιμοποιήθηκε για την λειτουργία των μεθόδων RFE και RFECV.

Η λειτουργία του αναφέρθηκε ήδη παραπάνω. Συνοπτικά, αποτελείται από δέντρα αποφάσεων, τα οποία καταλήγουν εν τέλει σε ένα συγκεκριμένο αποτέλεσμα.

Όσο αφορά τις παραμέτρους του, θα χρησιμοποιηθούν οι default τιμές, με τη σημαντικότερη εξ αυτών να είναι η παράμετρος «n_estimators» που ορίζεται ίση με 100. Αυτή η παράμετρος, ουσιαστικά, είναι ο αριθμός των δέντρων που θα φτιαχτούν.

Παρακάτω φαίνονται τα διαφορετικά F1 scores για το μοντέλο αυτό:

	F1 SCORE			Αλγόριθμος	Παράμετροι Αλγόριθμου	Features	Πείραμα
Imputation Strategy	mean	median	most frequent				
	0.6695	0.6642	0.6391	Random Forest Classifier	n_estimators=100, random_state=42	Initial Features	Δοκιμή του μοντέλου Random Forest Classifier με τα αρχικά features
	0.6692	0.6625	0.639	"	"	Initial Features + όλα όσα φτιάχτηκαν	Δοκιμή του μοντέλου Random Forest Classifier με τα features που φτιάχτηκαν
	0.6552	0.6533	0.6281	"	"	Τα 8 καλύτερα features μετά την RFE	Δοκιμή του μοντέλου Random Forest Classifier μετά την RFE
	0.7092	0.7058	0.695	"	"	Τα καλύτερα features μετά την RFECV	Δοκιμή του μοντέλου Random Forest Classifier μετά την RFECV

Εικόνα 14: *Πίνακας με τα διαφορετικά F1 scores για το μοντέλο «RandomForestClassifier»*

Φαίνεται ότι υπάρχει περαιτέρω αύξηση των F1 scores χρησιμοποιώντας αυτό το μοντέλο.

Συγκεκριμένα, και πάλι η καλυτερη στρατηγική είναι η mean, με την median να ακολουθεί και τη most frequent να έρχεται τελευταία.

Όσον αφορά τα features, παρατηρείται κάτι διαφορετικό εδώ από το προηγούμενο μοντέλο. Στη προκειμένη περίπτωση, τα αρχικά features παρουσιάζουν μεγαλύτερη ακρίβεια τόσο από το σύνολο των features (μαζί με αυτά που κατασκευάστηκαν), όσο και από τη μέθοδο RFE. Παρόλα αυτά, όπως αναμενόταν ο συνδυασμός των μεταβλητών που προέκυψαν από τη μέθοδο RFECV είναι και πάλι ο καλύτερος.

8.3 XGBOOST

Ο επόμενος αλγόριθμος που θα αναλυθεί, για να δημιουργηθεί μοντέλο είναι ο xgboost. Συγκεκριμένα, θα χρησιμοποιηθεί ο «XGBClassifier» της βιβλιοθήκης «xgboost» [28]

Κι αυτός ο αλγόριθμος είναι ένας αλγόριθμος μηχανικής μάθησης που χρησιμοποιείται κυρίως για προβλήματα ταξινόμησης. Φημίζεται κυρίως, λόγω της μεγάλης του απόδοσης.

Ο αλγόριθμος αυτός χρησιμοποιεί τη μέθοδο «gradient boosting» για να βελτιώσει την ακρίβεια των προβλέψεών του. [29] Συνδυάζει πολλαπλά δέντρα αποφάσεων και υποστηρίζει παράλληλη επεξεργασία για καλύτερη απόδοση.

Και για αυτόν τον αλγόριθμο θα χρησιμοποιηθούν όλες οι default παράμετροι.

Παρακάτω φαίνονται τα διαφορετικά F1 scores για το μοντέλο αυτό:

	F1 SCORE			Αλγόριθμος	Παράμετροι Αλγόριθμου	Features	Πείραμα
Imputation Strategy	mean	median	most frequent				
	0.6725	0.6682	0.6476	XGBClassifier	default	Initial Features	Δοκιμή του μοντέλου XGBClassifier με τα αρχικά features
	0.6753	0.6701	0.6483	"		Initial Features + όλα όσα φτιάχτηκαν	Δοκιμή του μοντέλου XGBClassifier με τα features που φτιάχτηκαν
	0.6618	0.6606	0.6395	"		Τα 8 καλύτερα features μετά την RFE	Δοκιμή του μοντέλου XGBClassifier μετά την RFE
	0.7165	0.7115	0.7063	"		Τα καλύτερα features μετά την RFECV	Δοκιμή του μοντέλου XGBClassifier μετά την RFECV

Εικόνα 15: *Πίνακας με τα διαφορετικά F1 scores για το μοντέλο «XGBClassifier»*

Όπως φαίνεται, υπάρχει ακόμη περισσότερη βελτίωση των F1 scores για το μοντέλο «XGBClassifier».

Όσο αφορά τη καλύτερη στρατηγική imputation, αυτή είναι και πάλι η mean. Ο καλύτερος συνδυασμός των features είναι και πάλι – με μεγάλη διαφορά – ο συνδυασμός που προέκυψε από τη μέθοδο RFECV. Από ότι φαίνεται και πάλι η μέθοδος RFE κάνει χειρότερες προβλέψεις, ακόμη και από τα initial features, άρα δε θα πρέπει να χρησιμοποιείται σε καμία περίπτωση ο συνδυασμός των features που προέκυψαν από αυτή.

Ενδεχομένως, θα μπορούσαν να εξεταστούν κι άλλα μοντέλα, για να βρεθεί κάποιος καλύτερος αλγόριθμος για το συγκεκριμένο πρόβλημα.

Επίσης, για κάθε ένα από τα μοντέλα που δοκιμάστηκαν θα μπορούσαν να γίνουν εκτενή πειράματα για να βρεθούν τα optimal parameters, για κάθε μοντέλο (hyperparameter tuning). [30] Με αυτό το τρόπο δε θα χρησιμοποιούνταν οι default παράμετροι, αλλά οι παράμετροι που οδηγούν κάθε μοντέλο στις καλύτερες προβλέψεις. Για να γίνει hyperparameter tuning υπάρχουν και αυτόματοι τρόποι.

Αντί να γίνει hyperparameter tuning θα γίνει μία άλλη διαδικασία που συνδυάζει το model selection με το hyperparameter tuning, αυτόματα, με σκοπό να βρεθεί το καλύτερο μοντέλο με τις καλύτερες δυνατές παραμέτρους, ειδικά για το συγκεκριμένο πρόβλημα ταξινόμησης.

9. ΕΥΡΕΣΗ ΤΟΥ ΚΑΛΥΤΕΡΟΥ ΜΟΝΤΕΛΟΥ ΚΑΙ ΤΩΝ ΚΑΛΥΤΕΡΩΝ ΠΑΡΑΜΕΤΡΩΝ ΜΕΣΩ ΤΟΥ TPOTCLASSIFIER

Το «TPOT» (Tree-based Pipeline Optimization Tool) είναι ένα open-source «AutoML» εργαλείο που αυτοματοποιεί την διαδικασία του σχεδιασμού και του optimization των pipelines της μηχανικής μάθησης.

«AutoML» είναι ορισμένα εργαλεία που αυτοματοποιούν τις διαδικασίες για τη παραγωγή μοντέλων μηχανικής μάθησης. [31]

Pipeline στη μηχανική μάθηση είναι μία δομημένη ακολουθία βημάτων επεξεργασίας δεδομένων, που συνήθως περιλαμβάνει την προεπεξεργασία δεδομένων, την εξέταση των features, την εκπαίδευση μοντέλων και την αξιολόγηση μοντέλων. [32]

Για τη λειτουργία του χρησιμοποιεί Γενετικούς Αλγόριθμους, για να βρει, τελικά, ποιο είναι το καλύτερο pipeline για το αντίστοιχο πρόβλημα που καλείται να επιλύσει.

Με τη διαδικασία «TPOTClassifier» γίνεται αυτόματη σχεδίαση των pipelines, χρήση γενετικών αλγόριθμων για να βελτιωθούν αυτά τα pipelines, hyperparameter tuning για να βρεθούν οι βέλτιστες παράμετροι για κάθε pipeline, επιλογή μοντέλου για την εύρεση του καταλληλότερου classification μοντέλου για το πρόβλημα και στο τέλος γίνεται σύγκριση των scores για κάθε pipeline, μαζί με cross-validation, ώστε να προκύψει το καλύτερο pipeline. [33]

Αυτή η διαδικασία θα γίνει μόνο για το dataset της στρατηγικής mean και με τα features που προέκυψαν από τη μέθοδο RFECV, καθώς αυτά έβγαλαν τα καλύτερα αποτελέσματα. Επίσης, δε θα γίνει για ολόκληρο το dataset, αλλά για ένα subset που

αποτελείται από 100 χιλιάδες γραμμές, για να εξοικονομηθεί χρόνος, καθώς η «TPOTClassifier» είναι μία χρονοβόρα μέθοδος.

Το καλύτερο pipeline που θα προκύψει θα χρησιμοποιηθεί τελικά για την εκπαίδευση όλου του συνόλου των δεδομένων για να βρεθούν τελικά το f1 score.

Για να πραγματοποιηθεί αυτή η διαδικασία χρησιμοποιείται το «TPOTClassifier» από τη βιβλιοθήκη «tpot». [34]

Για τις παραμέτρους της διαδικασίας «TPOTClassifier» επιλέγονται κάποιες συνηθισμένες παράμετροι. Συγκεκριμένα:

- generations: ο αριθμός των γενιών της διαδικασίας (πόσες φορές θα τρέξει). Κάθε φορά εξελίσσεται και γίνεται καλύτερη.
- population_size: ο αριθμός των πιθανών pipelines που θα εξεταστούν σε κάθε γενιά.
- verbosity: αφορά τα output μηνύματα που εκτυπώνονται κατά τη διάρκεια της λειτουργίας
- scoring: η μετρική που χρησιμοποιείται για να αξιολογηθεί κάθε pipeline. Στη προκειμένη περίπτωση επιλέγεται η μετρική «accuracy», η οποία αναλύθηκε στα προηγούμενα.
- cv: αφορά τη διαδικασία cross-validation και συγκεκριμένα πόσα folds θα πραγματοποιηθούν.

Αφού τρέξει η διαδικασία, τα αποτελέσματα είναι αυτά: το μοντέλο που επιλέχτηκε είναι το «XGBClassifier» το οποίο δοκιμάστηκε και προηγουμένως. Άρα, η επιλογή για τη δοκιμή του ήταν σωστή. Οι παράμετροι του μοντέλου και το F1 score που προκύπτει φαίνονται στη παρακάτω εικόνα:

	F1 SCORE			Αλγόριθμος	Παράμετροι Αλγόριθμου	Features	Πείραμα	Παράμετροι Πειράματος
Imputation Strategy	mean	median	most frequent					
	0.7251			XGBClassifier ως επιλογή του TPOTClassifier	learning_rate=0.5, max_depth=7, min_child_weight=12, n_estimators=100, n_jobs=1, subsample=0.7500000000000001, verbosity=0	Τα καλύτερα features μετά την RFECV	Δοκιμή του TPOTClassifier	generations=5, population_size=20, verbosity=2, scoring='accuracy', cv=5, random_state=42

Εικόνα 16: *Πίνακας με τις παραμέτρους του μοντέλου «XGBClassifier» και το F1 score που προκύπτουν μετά τη διαδικασία «TPOTClassifier»*

Όσο αφορά τις optimized παραμέτρους που διάλεξε η διαδικασία για το μοντέλο:

- learning_rate: χρησιμοποιείται για την αποφυγή του overfitting (όταν ένα μοντέλο «υπερεκπαιδεύεται» στα γνωστά δεδομένα και δεν έχει καλή απόδοση σε άγνωστα δεδομένα). [35] Μικρότερη τιμή κάνει το μοντέλο πιο «στιβαρό», αλλά μπορεί να χρειάζονται περισσότερα βήματα για να ολοκληρωθεί η διαδικασία. Η τιμή 0.5 που δίνεται θεωρείται σχετικά μεγάλη.
- max_depth: είναι το μέγιστο «βάθος» του κάθε δέντρου αποφάσεων. Μεγαλύτερο βάθος σημαίνει ότι το μοντέλο μπορεί να μάθει πιο πολύπλοκα μοτίβα, αλλά υπάρχει κίνδυνος για overfitting. Η τιμή 7 που δίνεται σημαίνει ότι το μέγιστο βάθος κάθε δέντρου είναι 7 επίπεδα.
- min_child_weight: αυτό είναι το ελάχιστο άθροισμα του βάρους που απαιτείται σε ένα παιδί (απόγονος δέντρου). Εάν το βάρος που εμφανιστεί

σε κάποιο δέντρο είναι μικρότερο από αυτό, τότε δε θα πραγματοποιηθεί split του δέντρου σε παιδιά.

- n_estimators: αυτός είναι ο αριθμός των δέντρων που φτιάχνονται.
- n_jobs: αυτό αφορά το πόσα cores του επεξεργαστή του υπολογιστή θα χρησιμοποιηθούν για την εκτέλεση της διαδικασίας (αλλάζει ανάλογα τη δύναμη του υπολογιστή).
- subsample: αφορά το ποσοστό των δεδομένων που χρησιμοποιούνται για την εκπαίδευση των δέντρων. Εδώ επιλέγεται το 75% των δεδομένων να χρησιμοποιηθεί.
- verbosity: αφορά τα μηνύματα εκτύπωσης κατά τη διάρκεια της λειτουργίας της διαδικασίας.

Τελικά το f1 score που προκύπτει είναι ίσο με $F1\ score = 0.7251$, που είναι το καλύτερο που έχει προκύψει εώς τώρα. Αυτή η τιμή θεωρείται καλή και το μοντέλο κάνει ικανοποιητικές προβλέψεις, σχετικά με τα δεδομένα που του δίνονται.

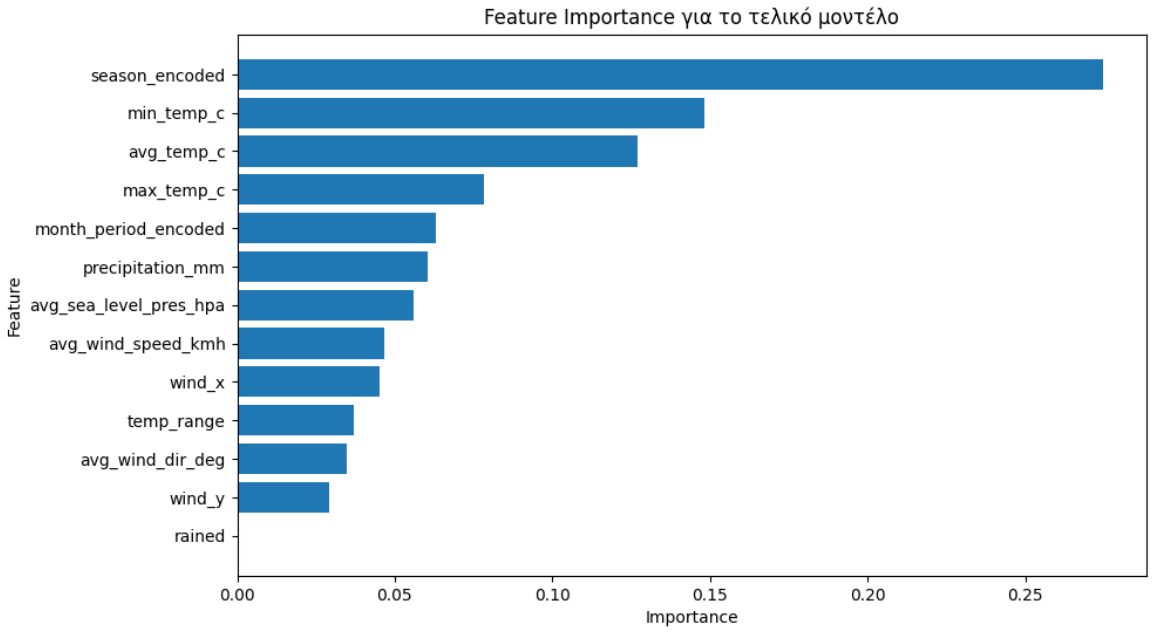
10. ΔΗΜΙΟΥΡΓΙΑ ΤΕΛΙΚΩΝ ΔΙΑΓΡΑΜΜΑΤΩΝ

Θα σχεδιαστούν μερικά ακόμα plots για να παρουσιαστούν τα αποτελέσματα και να μπορορέσουν να βγουν μερικά συμπεράσματα ευκολότερα. Τα νέα plots θα σχεδιαστούν για τα καλύτερα αποτελέσματα που προέκυψαν από το σύνολο των πειραμάτων.

Άρα, τα διαγράμματα θα σχεδιαστούν για το dataset που προέκυψε από τη στρατηγική mean imputation, για τον συνδυασμό των features που προέκυψε από τη μέθοδο «RFECV» και για το μοντέλο «».

10.1 ΔΙΑΓΡΑΜΜΑ FEATURE IMPORTANCE

Όπως είχε δημιουργηθεί το διάγραμμα για το feature importance του βασικού μοντέλου που χρησιμοποιήθηκε, έτσι και για το τελικό, optimized μοντέλο θα γίνει το αντίστοιχο διάγραμμα.



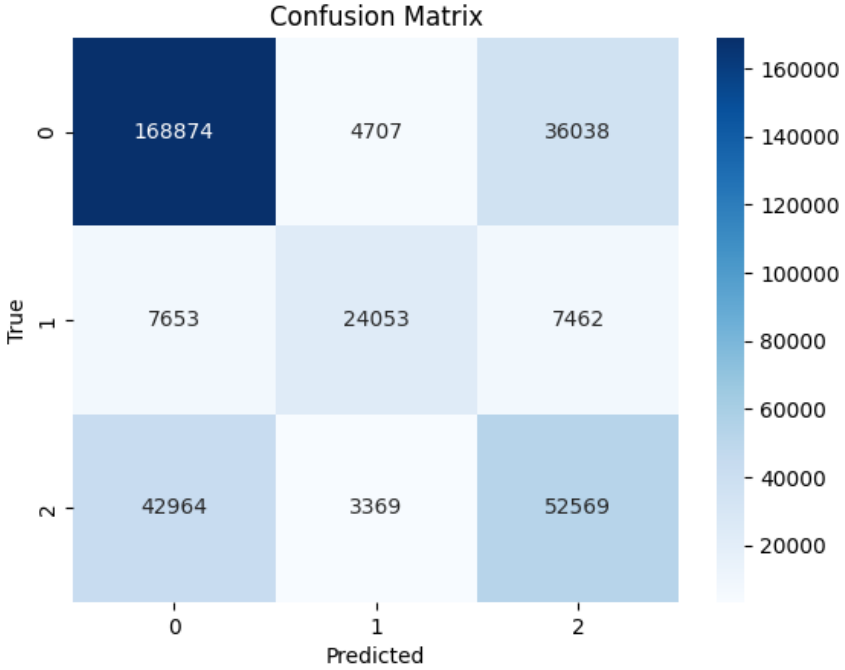
Εικόνα 17: *Feature Importance plot για το μοντέλο «XGBClassifier» με τις optimized παραμέτρους που προέκυψαν από τη διαδικασία «TPOTClassifier»*

Όπως φαίνεται και στο διάγραμμα, το feature που είναι με διαφορά το σημαντικότερο είναι το «season_encoded». Το feature που φτιάχτηκε «rained» εδώ φαίνεται πως δεν οφελεί καθόλου το μοντέλο.

10.2 ΠΙΝΑΚΑΣ ΣΥΓΧΥΣΗΣ (CONFUSION MATRIX)

Στα προηγούμενα έγινε αναφορά σε πίνακες «Confusion Matrix». Ξαναυπενθυμίζεται ότι είναι οι πίνακες όπου εμφανίζονται τα True Positives, True Negatives, False Positives και False Negatives. Από έναν τέτοιο πίνακα μπορούν να βγούνε ορισμένα συμπεράσματα για την ακρίβεια των προβλέψεων για ένα μοντέλο. Σημειώνεται ότι οι κλάσεις αντιστοιχίζονται ως εξής:

- Κλάση 0: Ηπειρωτικό Κλίμα
- Κλάση 1: Μεσογειακό Κλίμα
- Κλάση 2: Ωκεανικό Κλίμα



Εικόνα 18: *Confusion Matrix πίνακας για το μοντέλο μοντέλο «XGBClassifier» με τις optimized παραμέτρους που προέκυψαν από τη διαδικασία «TPOTClassifier»*

Από τον πίνακα σύγχυσης, μπορεί κάποιος να δει στα άγνωστα δεδομένα ποια είναι η σωστή πρόβλεψη και ποια ήταν η πρόβλεψη του μοντέλου. Για να είναι πιο εύκολο στη κατανόηση θα εκτυπωθούν και ορισμένες μετρικές ακρίβειας, για τις διαφορετικές κλάσεις (μεσογειακό, ηπειρωτικό, ωκεάνιο κλίμα).

Οι μετρικές που θα χρησιμοποιηθούν είναι οι εξής: [24]

- Precision: η ακρίβεια αποτελεί τον λόγο των σωστά προβλεπόμενων παρατηρήσεων ως προς το σύνολο των προβλεπόμενων παρατηρήσεων, απαντώντας στην ερώτηση «ποιο ποσοστό των παρατηρήσεων ήταν πραγματικά σωστό;»

$$\text{Precision} = \frac{Relevant\ retrieved\ instances}{All\ retrieved\ instances}$$

- Recall: η ευαισθησία πρόκειται για τον λόγο των σωστά προβλεπόμενων παρατηρήσεων ως προς το σύνολο των παρατηρήσεων στην πραγματική

κλάση απαντώντας στην ερώτηση «ποιο ποσοστό των πραγματικών θετικών αναγνωρίστηκε σωστά;»

$$Recall = \frac{Relevant\ retrieved\ instances}{All\ retrieved\ instances}$$

- F1-score: έχει αναλυθεί ήδη εκτενώς.

Προκύπτουν τα εξής:

Κλάση	Precision	Recall	F1-Score
0 – Ηπειρωτικό	0.77	0.81	0.79
1 – Μεσογειακό	0.75	0.61	0.67
2 - Ωκεανικό	0.55	0.53	0.54

Από αυτές τις μετρικές βγαίνει το συμπέρασμα ότι το μοντέλο δεν αποδίδει τόσο καλά στη κλάση 2 (ωκεάνιο κλίμα), όσο στις άλλες δύο. Αυτό μπορεί να οφείλεται στο ότι ωκεάνιο κλίμα έχουν λιγότερες πόλεις της Ευρώπης, συγκριτικά με τα άλλα δύο κλίματα, άρα ως αποτέλεσμα υπάρχει μικρότερος αριθμός δεδομένων για την εκπαίδευση. Εναλλακτικά, μπορεί να οφείλεται σε περισσότερες χαμένες τιμές για αυτή τη κλιματική ζώνη, με αποτέλεσμα το imputation για αυτή να μην ήταν αρκετά ικανοποιητικό.

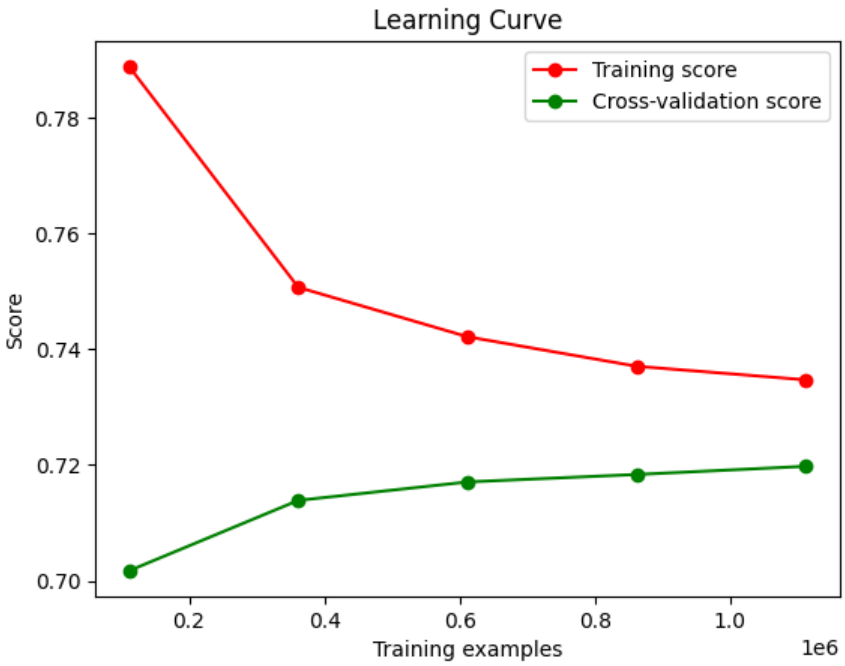
Παρόλα αυτά, το τελικό μοντέλο που αναπτύχθηκε θεωρείται ικανοποιητικό, καθώς κατά κύριο λόγο πραγματοποιεί σωστές προβλέψεις. Ειδικά στη κλάση 0 (ηπειρωτική κλιματική ζώνη) η ακρίβεια είναι αρκετά μεγάλη.

10.3 ΚΑΜΠΥΛΗ ΜΑΘΗΣΗΣ (LEARNING CURVE)

Η ερμηνεία των καμπυλών μάθησης είναι μια κρίσιμη πτυχή της αξιολόγησης και του συντονισμού των μοντέλων στη μηχανική μάθηση. Οι καμπύλες μάθησης παρέχουν μια οπτική αναπαράσταση του τρόπου με τον οποίο η απόδοση ενός μοντέλου εξελίσσεται με διαφορετικές ποσότητες δεδομένων εκπαίδευσης.

Συγκεκριμένα, από τη καμπύλη μάθησης μπορούν να βγουν συμπεράσματα για τη κατανόηση της απόδοσης του μοντέλου, για την υπερπροσαρμογή, την υποπροσαρμογή, την αξιολόγηση της πολυπλοκότητας του μοντέλου κλπ.

Για να γίνει το διάγραμμα της καμπύλης μάθησης θα χρησιμοποιηθεί η «learning_curve» από τη βιβλιοθήκη «scikit-learn». [36]



Εικόνα 19: *Το διάγραμμα της καμπύλης μάθησης για το μοντέλο «XGBClassifier» με τις optimized παραμέτρους που προέκυψαν από τη διαδικασία «TPOTClassifier»*

Η καμπύλη μάθησης δείχνει ότι η προσθήκη περισσότερων δεδομένων βελτιώνει την ικανότητα του μοντέλου να γενικεύει.

Η φθίνουσα διαφορά μεταξύ των αποτελεσμάτων εκπαίδευσης και επικύρωσης υποδηλώνει ότι το μοντέλο επωφελείται από τα πρόσθετα δεδομένα και ότι η υπερπροσαρμογή δεν αποτελεί σημαντικό ζήτημα.

Η συνεπής απόδοση υποστηρίζει περαιτέρω την αξιοπιστία του μοντέλου.

11. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΚΑΤΕΥΘΥΝΣΕΙΣ

Με τελικό F1 score ίσο με 0.7251 σαφώς το μοντέλο θα μπορούσε να φανεί χρήσιμο για πρακτικούς σκοπούς, ειδικά εάν η ισορροπημένη απόδοση είναι ζωτικής σημασίας.

Παρ΄ όλα αυτά, ενώ το 0.7251 είναι μια καλή βαθμολογία, υποδηλώνει επίσης, ότι υπάρχει περιθώριο βελτίωσης. Συγκεκριμένα, η περαιτέρω ρύθμιση του μοντέλου θα οδηγούσε σε καλύτερες επιδόσεις της ταξινόμησης. Θα μπορούσαν να δοκιμαστούν κι άλλες μέθοδοι ή αυτές που δοκιμάστηκαν να έτρεχαν για περισσότερα δεδομένα, καθώς σε κάποια σημεία αναγκαστικά χρησιμοποιήθηκαν λιγότερες γραμμές από το σύνολο, λόγω του χρόνου τρεξίματος.

Μέσω της εύστοχης πρόβλεψης του κλίματος θα μπορούσαν να αναχθούν συμπεράσματα για τα μοτίβα που διέπουν τις διαφορετικές κλιματικές ζώνες. Σταδιακά, θα γινόταν δυνατή μία ολοένα αυξανόμενη μεγέθυνση της ειδίκευσης, μεταβαίνοντας από γενικότερα κλίματα (όπως στη προκειμένη περίπτωση) σε ειδικά κλίματα (όπως αυτά του μοντέλου Κέππεν) και εν τέλει στην πρόβλεψη των πόλεων ή και μικρότερων γεωγραφικών περιοχών.

Μελλοντικά, θα μπορούσε να εξεταστεί το πως συγκρίνονται οι παραδοσιακές προβλέψεις με τα μοντέλα που χρησιμοποιήθηκαν στην έρευνα αυτή αλλά και με

βελτιωμένες εκδόσεις τους. Στο πλαίσιο αυτό θα απαιτούνταν πιο ενδελεχείς στατιστικές μέθοδοι ώστε να αναλυθούν οι αστοχίες του μοντέλου εις βάθος αλλά και να τελεστεί δυνατή η σύγκρθση των αποτελεσμάτων με των φυσικών μοντέλων.

Αναφορές

- [1] Κλιματικές Ζώνες σύμφωνα με το μοντέλο Κέππεν: https://en.wikipedia.org/wiki/Köppen_climate_classification
- [2] Χάρτης της Ευρώπης με τις διαφορετικές κλιματικές ζώνες: https://upload.wikimedia.org/wikipedia/commons/d/d5/Koppen-Geiger_Map_Europe_present.svg
- [3] Λίστα των διαφορετικών κλιματικών ζωνών σύμφωνα με τη κλιματική ταξινόμηση Κέππεν: https://www.mindat.org/climate.php
- [4] “The Weather Dataset”: https://www.kaggle.com/datasets/guillemservera/global-daily-climate-data
- [5] Ο χάρτης Κεππεν που χρησιμοποιείται για την αντιστοίχιση των γεωγραφικών συντεταγμένων των πόλεων με τις κλιματικές ζώνες: https://figshare.com/articles/dataset/Present_and_future_K_ppen-Geiger_climate_classification_maps_at_1-km_resolution/6396959/2
- [6] Βιβλιοθήκη «rosterio» για το διάβασμα του χάρτη Κέππεν: https://rasterio.readthedocs.io/en/stable/
- [7] Βιβλιοθήκη «scikit-learn»: https://scikit-learn.org/stable/
- [8] Βιβλιοθήκη «Matplotlib» για τη δημιουργία των διαγραμμάτων: https://matplotlib.org/
- [9] Βιβλιοθήκη «seaborn» για το visualization των στατιστικών δεδομένων: https://seaborn.pydata.org/
- [10] Πληροφορίες για το imputation: https://en.wikipedia.org/wiki/Imputation_(statistics)
- [11] «SimpleImputer»: https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html
- [12] Πληροφορίες για τις στρατηγικές imputation (mean, median, mode): https://reintech.io/term/mean-median-mode-imputation-software-development
- [13] «LogisticRegression»: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression
- [14] «F1 score»: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html
- [15] Πληροφορίες για το «Confusion Matrix»:

https://www.researchgate.net/figure/Typical-structure-of-confusion-matrix_fig1_363509386
- [16] «zscore»: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.zscore.html
- [17] Πληροφορίες για το «Feature Engineering»: https://en.wikipedia.org/wiki/Feature_engineering
- [18] Βιβλιοθήκη «numpy»: https://numpy.org/
- [19] Πληροφορίες για το «Feature Selection»: https://en.wikipedia.org/wiki/Feature_selection
- [20] «Recursive Feature Elimination» (RFE): https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html
- [21] «RandomForestClassifier»: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- [22] Το σχήμα για τη λειτουργία του «Cross-Validation»: https://www.sharpsightlabs.com/blog/cross-validation-explained/
- [23] «Recursive Feature Elimination with Cross Validation» (RFECV): https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html1
- [24] Πληροφορίες για το «Accuracy», το «Precision» και το «Recall»: https://en.wikipedia.org/wiki/Precision_and_recall
- [25] Διάγραμμα για το «Cross-Validation»: https://scikit-learn.org/stable/modules/cross_validation.html
- [26] Πληροφορίες για το «k-Nearest Neighbors»: https://luis-miguel-code.medium.com/knn-k-nearest-neighbors-and-kneighborsclassifier-what-it-is-how-it-works-and-a-practical-914ec089e467
- [27] «KNeighborsClassifier»: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
- [28] Βιβλιοθήκη «xgboost»: https://xgboost.readthedocs.io/en/stable/python/python_api.html
- [29] Πληροφορίες για το «Gradient Boosting»: https://en.wikipedia.org/wiki/Gradient_boosting
- [30] Πληροφορίες για το «Hyperparameter Tuning»: https://www.geeksforgeeks.org/hyperparameter-tuning/
- [31] Πληροφορίες για τα «AutoML» εργαλεία: https://en.wikipedia.org/wiki/Automated_machine_learning

- [32] Πληροφορίες για το pipeline στη μηχανική μάθηση: https://valohai.com/machine-learning-pipeline/
- [33] Πληροφορίες για το «TPOT» και το «TPOTClassifier»: http://epistasislab.github.io/tpot/using/
- [34] Βιβλιοθήκη «tpot»: http://epistasislab.github.io/tpot/
- [35] Πληροφορίες για το overfitting: https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/
- [36] «learning_curve»: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.learning_curve.html

Ορισμένες Python βιβλιοθήκες που χρησιμοποιήθηκαν και δεν αναφέρθηκαν:

- «pandas» (βιβλιοθήκη για ανάλυση δεδομένων): https://pandas.pydata.org/
- «os» (βιβλιοθήκη για διάβασμα/αποθήκευση αρχείων): https://docs.python.org/3/library/os.html
- «pyarrow.parquet» (βιβλιοθήκη χρήσιμη για το διάβασμα του dataset): https://arrow.apache.org/docs/python/parquet.html
- «calendar» (βιβλιοθήκη για επεξεργασία δεδομένων σε μορφή ημερομηνίας): https://docs.python.org/3/library/calendar.html