ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ 5G, ΤΕΧΝΟΛΟΓΙΕΣ, ΕΦΑΡΜΟΓΕΣ, ΚΑΙ ΒΑΣΙΚΟΙ ΔΕΙΚΤΕΣ ΑΠΟΔΟΣΗΣ

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2024 – 2025

ΚΟΥΡΗ ΜΑΡΙΑ | Α.Μ. 1084526 | up1084526@ac.upatras.gr

ΣΩΚΡΑΤΗΣ ΜΑΝΤΕΣ | Α.Μ. 1093421 | up1093421@ac.upatras.gr

Εισαγωγή

Η παρούσα εργασία υλοποιήθηκε στο πλαίσιο του μαθήματος «Αρχιτεκτονικές 5G, Τεχνολογίες, Εφαρμογές και Βασικοί Δείκτες Απόδοσης».

Η αναφορά συνοδεύεται από το αντίστοιχο **Jupyter Notebook** σε γλώσσα προγραμματισμού **Python**.

Για την υλοποίηση της εργασίας επιλέχθηκαν 5 ημέρες:

- **14 – 12 – 2013**
- **15 – 12 – 2013**
- **16 – 12 – 2013**
- **17 – 12 – 2013**
- **18 – 12 – 2013**

Στόχος είναι ο **εντοπισμός** των περιοχών με την **μεγαλύτερη τηλεπικοινωνιακή δραστηριότητα**, στο παραπάνω χρονικό διάστημα. Η περιοχή ελέγχου περιορίστηκε στο κέντρο του Μιλάνου με συντεταγμένες **45.46°N 9.19°E.**

Θεωρητικό μέρος

Ερώτημα 1ο – Clustering Στις Τηλεπικοινωνίες

Η Μηχανική Μάθηση διαχωρίζεται σε **επιβλεπόμενη** (supervised learning) και **μη επιβλεπόμενη** (unsupervised learning) μηχανική μάθηση [1]. Στην **επιβλεπόμενη** μάθηση το μοντέλο εκπαιδεύεται σε ένα σύνολο δεδομένων το οποίο περιλαμβάνει **ετικέτες** (labels) και ο στόχος είναι είτε η **ταξινόμηση** των δεδομένων στην **σωστή κατηγορία** (classification) είτε η **πρόβλεψη τιμών** (regression) [2]. Αντίθετα, στην **μη επιβλεπόμενη** μάθηση το μοντέλο **δεν περιλαμβάνει** ετικέτες (labels) και το επιθυμητό είναι να **βρεθούν δομές ή μοτίβα** στα δεδομένα για την συσταδοποίησή τους (clustering) ή για να **μειωθούν οι διαστάσεις** τους (dimensionality reduction) [3].

Η τεχνική της **συσταδοποίησης** (clustering) χρησιμοποιείται για να εντοπίσει συστάδες - ομάδες (clusters), μέσα σε ένα σύνολο δεδομένων. Τα δεδομένα ομαδοποιούνται με **κριτήριο** την **ομοιότητά** τους ή ορισμένα **κοινά τους χαρακτηριστικά**, **χωρίς** να υπάρχουν **προκαθορισμένες ετικέτες** (labels) για να χαρακτηριστούν με αυτές. Για να γίνει **ορθή** συσταδοποίηση τα δεδομένα θα πρέπει να έχουν **υψηλό ποσοστό ομοιότητας**, ώστε να ομαδοποιηθούν, και τα δεδομένα **διαφορετικών** συστάδων θα πρέπει να έχουν όσο το δυνατόν **μεγαλύτερη διαφοροποίηση**. Οι αλγόριθμοι συσταδοποίησης μπορούν να βασίζονται σε διαφορετικές τεχνικές, όπως για παράδειγμα:

- **Βασισμένοι σε κέντρα** (k-means): κάθε ομάδα έχει ένα **κεντροειδές** (centroid) και τα δεδομένα τοποθετούνται στην κοντινότερη ομάδα με βάση μια **επιλεγμένη μετρική απόστασης**.
- **Ιεραρχικοί** (Hierarchical Clustering): δημιουργούν μια **ιεραρχική δομή συστάδων**, η οποία μπορεί να προσεγγιστεί είτε με την **συσσωρευτική** μέθοδο (agglomerative) είτε με την **διαιρετική** μέθοδο (divisive). Στην **πρώτη** περίπτωση τα δεδομένα βρίσκονται σε **ξεχωριστές** συστάδες και **σταδιακά** συγχωνεύονται δημιουργώντας μεγαλύτερες ομάδες, ενώ στην **δεύτερη** τα δεδομένα βρίσκονται σε μία **ενιαία** ομάδα και σταδιακά χωρίζονται σε **μικρότερες** συστάδες.
- **Βασισμένοι σε πυκνότητα** (Density-Based Spatial Clustering of Applications with Noise): δημιουργούνται ομάδες βάσει της **πυκνότητας** των δεδομένων.

- **Μοντέλα μείγματος κατανομών** (Gaussian Mixture Models – GMM): αντί να κατατάσσει ένα σημείο σε μία ομάδα, του **αποδίδεται η πιθανότητα** να ανήκει σε **κάθε ομάδα** του συνόλου.

Η τεχνική αυτή έχει πληθώρα εφαρμογών. Μερικά παραδείγματα χρήσης είναι η **ανάλυση πελατών** σε τμήματα **marketing**, η **ανίχνευση ανωμαλιών** και **ύποπτων μοτίβων** σε δεδομένα δικτύου, η **σύσταση περιεχομένων** σε χρήστες εφαρμογών περιεχομένου, αλλά και η **εφαρμογή στον τομέα των τηλεπικοινωνιών**. Στην περίπτωση του τομέα των **τηλεπικοινωνιών**, η τεχνική της συσταδοποίησης (clustering) εφαρμόζεται για **την κατηγοριοποίηση υπηρεσιών** όπως τα SMS, τα δεδομένα (data) και η φωνητική επικοινωνία (voice). Μέσω του clustering, οι πάροχοι μπορούν να αναλύσουν **πρότυπα χρήσης**, να εντοπίσουν ομάδες χρηστών με **παρόμοιες ανάγκες** και να **προσαρμόσουν** τις **υπηρεσίες** τους αναλόγως, βελτιώνοντας την εμπειρία των πελατών.

Πιο συγκεκριμένα, στην περίπτωση των **Δικτύων Πέμπτης Γενιάς** (5g) η τεχνική του clustering συμβάλλει στην **βελτίωση της ποιότητας των υπηρεσιών** (QoS). Όπως αναφέρεται και στην βιβλιογραφία η τεχνική της συσταδοποίησης επιτρέπει την συγκέντρωση χρηστών με παρόμοια μοτίβα χρήσης με στόχο την σωστή κατανομή των πόρων [4, 5, 6]. Παράλληλα, επισημαίνεται πως η **εκθετική αύξηση** των κινητών συσκευών και του όγκου των δεδομένων που ανταλλάσσονται δημιουργεί την **ανάγκη διαχείρισης** και **κατανομής** τους, ώστε να **αποφευχθεί η συμφόρηση** [7]. Έτσι προτείνονται τρόποι συσταδοποίησης των χρηστών, όπου με την ορθή κατανομή των πόρων και των συνδρομητών, αποφεύγεται η συμφόρηση και συνεπώς διατηρείται η υψηλή ποιότητα των υπηρεσιών.

Ακόμη, η διαδικασία της συσταδοποίησης συνεισφέρει στην **μείωση** των **παρεμβολών**. Ο **συντονισμός** της **χρήσης συχνοτήτων** και πόρων εντός κάθε cluster μειώνει σημαντικά τις συνεπακόλουθες παρεμβολές ανάμεσα σε διαφορετικές κυψέλες [8]. Τέλος, βοηθά στη **μείωση** της **κατανάλωσης ενέργειας** [9]. Ομαδοποιώντας τους σταθμούς βάσης σε clusters, το δίκτυο μπορεί να **ελαχιστοποιεί** τους ενεργούς σταθμούς ανά ομάδα όταν η κίνηση είναι χαμηλή, απενεργοποιώντας προσωρινά **περιττές** κυψέλες.

Συμπερασματικά, η εφαρμογή της συσταδοποίησης στην περίπτωση μη επιβλεπόμενης μάθησης μπορεί να αποτελέσει βάση για την εξαγωγή σημαντικών συμπερασμάτων σε ένα ευρύ πεδίο. Ιδιαίτερα στον τομέα των τηλεπικοινωνιών και των δικτύων αποτελεί σημαντικό εργαλείο για τους παρόχους τηλεπικοινωνιακών υπηρεσιών, επιτρέποντας την υψηλή ποιότητα υπηρεσιών, με τις ελάχιστες δυνατές δυσλειτουργίες και με χαμηλό κόστος.

Ερώτημα 2ο – Ανάλυση Του Dataset

Το **Milano Dataset** περιλαμβάνει δεδομένα σχετικά με το **τηλεπικοινωνιακό δίκτυο** στην πόλη Milano, για συγκεκριμένο χρονικό διάστημα. Για την παρούσα εργασία επιλέχθηκαν οι ημερομηνίες **14-12-2013 έως 18-12-2013**, καθώς το αρχείο καταγραφής είναι μεγάλο σε όγκο και περιέχει περισσότερα δεδομένα. Στην αναφορά που συνοδεύει το dataset [10] (documentation) αναφέρεται η **ανάλυση** των δεδομένων - στηλών και η **δομή** των αρχείων. Στο **repository** που παρέχεται στην εκφώνηση [11] παρατίθεται **πρότυπο ανάλυσης** των δεδομένων που περιέχονται στα δοθέντα .txt αρχεία.

Κατόπιν ανάλυσης του dataset (ο κώδικας ανάλυσης εξηγείται σε επόμενα ερωτήματα) προκύπτει ότι τα δεδομένα περιέχουν το **gridID**, το οποίο είναι το αναγνωριστικό του κάθε **cell** στο **10x10 grid** όπου χωρίζεται η πόλη. Για να εντοπιστεί το κάθε κελί στον χάρτη, με βάση το gridID, είναι **απαραίτητος** ο **συσχετισμός** του **gridID** με τις **συντεταγμένες**, πληροφορία η οποία παρέχεται σε ένα **geoJSON** αρχείο [12].

Επιπλέον παρέχεται η πληροφορία **timeInterval**, η οποία περιέχει την **χρονική στιγμή** που ξεκινάει η **καταγραφή** των δεδομένων και εκφράζεται σε **χιλιοστά του δευτερολέπτου**. Ακόμη καταγράφεται ο **κωδικός της χώρας** (country code) προς την οποία ο συνδρομητής **πραγματοποίησε κλήση** (callIn, callOut), **έστειλε μήνυμα** (smsIn, smsOut). Τέλος, καταγράφεται η **δραστηριότητα κίνησης στο διαδίκτυο**, δηλαδή γίνεται αναφορά στον αριθμό των **CDRs** (Call Detail Records), δηλαδή τις καταγραφές των κλήσεων που πραγματοποιούνται μέσα σε μια **συγκεκριμένη γεωγραφική περιοχή**, κατά τη διάρκεια ενός **συγκεκριμένου χρονικού διαστήματος**. Συνεπώς το dataset περιέχει πληροφορίες για την κίνηση του τηλεπικοινωνιακού δικτύου μία συγκεκριμένη χρονική στιγμή, σε ένα συγκεκριμένο cell του grid.

Αξιοποίηση δεδομένων

Για το **clustering** θα αξιοποιηθούν τόσο **χωρικά**, όσο και **χρονικά** δεδομένα. Για **ταυτοποίηση** του κάθε κελιού στον χώρο, θα χρησιμοποιηθεί το αναγνωριστικό του **cell** (gridID). Ακόμη θα χρησιμοποιηθούν οι **συντεταγμένες** του κάθε cell για να προσδιοριστεί η **θέση** τους στο **χάρτη**. Το dataset περιέχει πληροφορίες σχετικά με το **latitude** και το **longitude** των cells. Η πληροφορία αυτή θα επιτρέψει τον **συσχετισμό** και **εντοπισμό γειτονικών περιοχών** με παρόμοια τηλεπικοινωνιακή δραστηριότητα.

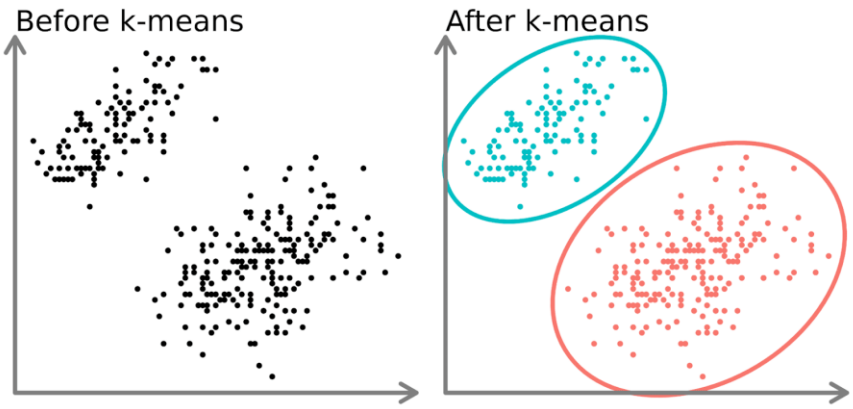
Για ορθό συσχετισμό των cells είναι απαραίτητο το χρονικό πλαίσιο της δραστηριότητα. Πληροφορία για τον χρόνο που εντοπίζεται κίνηση στο δίκτυο παρέχει η μεταβλητή **timeInterval**, η οποία προσδιορίζει το **timestamp** που εντοπίζεται δραστηριότητα. Τέλος, πληροφορίες που χαρακτηρίζουν την **δραστηριότητα**, όπως πλήθος κλήσεων, sms και χρήση δεδομένων θα αξιοποιηθούν ως κριτήριο κατάταξης των cells.

Ερώτημα 3ο – Μέθοδοι Clustering

Η συσταδοποίηση (clustering) αποτελεί μια πολύ βασική τεχνική μη επιβλεπόμενης μηχανικής μάθησης (unsupervised learning). Η διαδικασία αυτή μπορεί να χρησιμοποιηθεί με επιτυχία σε **χωρικά** και **χρονικά** δεδομένα (time series), όπως τα δεδομένα του Milano Dataset. Για την βελτίωση των τηλεπικοινωνιακών υπηρεσιών μπορούν να χρησιμοποιηθούν διάφορες τεχνικές.

K-Means

Αρχικά, ευρέως γνωστή τεχνική που μπορεί να αξιοποιηθεί στο συγκεκριμένο Dataset είναι η k-means [13, 14]. Η συγκεκριμένη μέθοδος συσταδοποίησης χρησιμοποιείται για τη **διαίρεση** ενός **συνόλου** δεδομένων σε **k ομάδες** (clusters), όπου **κάθε δεδομένο** ανήκει στην ομάδα με το **πλησιέστερο κεντροειδές** (centroid). Σε **πρώτη φάση** επιλέγονται **τυχαία k** σημεία από το σύνολο δεδομένων ως **αρχικά κεντροειδή** και **κάθε σημείο** ανατίθεται στο **πλησιέστερο κεντροειδές**, σχηματίζοντας έτσι k ομάδες. Έπειτα υπολογίζονται **εκ νέου** τα **κεντροειδή** κάθε ομάδας, ως η **μέση τιμή** των σημείων της κάθε ομάδας. Η διαδικασία **επαναλαμβάνεται** έως ότου τα κεντροειδή **σταθεροποιηθούν** και **δεν μεταβάλλονται** ή έστω η μεταβολή είναι **αμελητέα**.



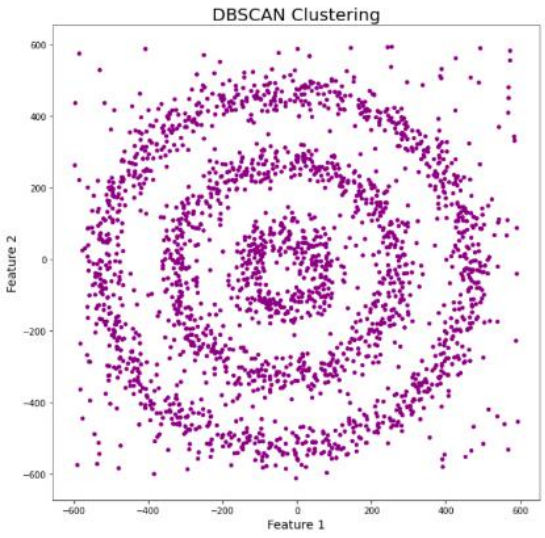
Εικόνα 1: Οπτική αναπαράσταση του αλγορίθμου K-Means

Η τεχνική που περιγράφεται παραπάνω είναι **απλή** και είναι **γρήγορη** στην εφαρμογή της. Ακόμη είναι ιδιαίτερα **αποδοτική** σε **σύνολα μεγάλων δεδομένων**, είτε **χωρικών** είτε **χρονικών**, και μπορεί να αποτελέσει μία **καλή πρακτική διαχωρισμού** των δεδομένων σε συστάδες. Παρόλα αυτά, υπάρχουν κάποιοι **περιορισμοί** στην εφαρμογή της. Για να εφαρμοστεί απαιτεί **προκαθορισμένο αριθμό συστάδων**, ενώ υποτίθεται πως οι **ομάδες** έχουν **παρόμοιο μέγεθος**. Επιπλέον η

συγκεκριμένη μέθοδος απαιτεί να **μην υπάρχουν ανωμαλίες** ή **ανομοιομορφίες** στα δεδομένα, καθώς **επηρεάζεται η απόδοση**, και **δεν** είναι **ανθεκτική** σε **outliers**. Η δομή του Milano Dataset **εξυπηρετεί** την εφαρμογή του k-means clustering. Η γνώση της **γεωμετρίας** του **grid** και του **πλήθους** των **cells** συμβάλλει στον **καθορισμό** του **πλήθους** των συστάδων και στον **εύκολο υπολογισμό** του **κεντροειδούς**.

DBSCAN

Μία ακόμη τεχνική που μπορεί να αξιοποιηθεί για την συσταδοποίηση **χωροχρονικών** δεδομένων είναι η **DBSCAN** [15]. Η συγκεκριμένη μέθοδος βασίζεται στην **πυκνότητα** των δεδομένων. Για να εφαρμοστεί πρέπει να οριστούν τα μεγέθη **Eps**(ε), δηλαδή η **ακτίνα γειτονιάς για ένα σημείο**, και **MinPts**, όπου είναι το **ελάχιστο απαιτούμενο πλήθος σημείων** για να θεωρηθεί μία **περιοχή** συστάδα. Για την εφαρμογή του τα σημεία **κατηγοριοποιούνται** και ορίζονται σημεία **πυρήνα**, όπου έχουν **τουλάχιστον MinPts** στην **γειτονιά** τους, τα σημεία **ορίου** που ανήκουν σε συστάδα αλλά **δεν** είναι πυρήνας και τα σημεία **θορύβου**, τα οποία **δεν** ανήκουν ούτε σε **πυρήνα**, ούτε σε **όριο**. Για την δημιουργία των συστάδων τα σημεία πυρήνα **συνδέονται με άλλα** σημεία πυρήνα, σε **απόσταση ε**, και δημιουργούν μία συστάδα. Στην συνέχεια τα σημεία **ορίου** ανατίθενται στην **πλησιέστερη συστάδα** και τα σημεία **θορύβου δεν** συμπεριλαμβάνονται.



Εικόνα 2: Οπτική αναπαράσταση του αλγορίθμου DBSCAN

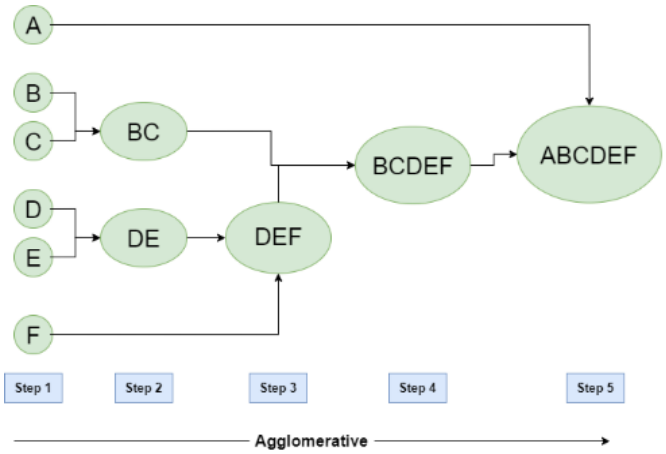
Η συγκεκριμένη τεχνική μπορεί να εφαρμοστεί με **επιτυχία** σε **χωροχρονικά** δεδομένα και **δεν** απαιτεί **προκαθορισμένο** αριθμό **συστάδων ή σχήμα**. Ένα ακόμη πλεονέκτημα είναι η ικανότητα της συγκεκριμένης τεχνικής να αντιμετωπίσει **αποτελεσματικά** τον **θόρυβο** στα δεδομένα. Παρόλα αυτά **εάν δεν επιλεχθούν σωστά MinPts και ε** ή τα δεδομένα έχουν **διαφορετική πυκνότητα** (μη κατανεμημένη) **μειώνεται** η απόδοσή του. Όλα τα παραπάνω καθιστούν την συγκεκριμένη μέθοδο **ικανή** να εφαρμοστεί στο Milano Dataset, αλλά θα

πρέπει να γίνει **προσεκτική επιλογή** των παραμέτρων και πριν την εφαρμογή να ελεγχθεί ότι τα δεδομένα δεν είναι ιδιαίτερα πυκνά σε συγκεκριμένες περιοχές, ώστε να μην μπορεί να τα διαχειριστεί ο αλγόριθμος.

Hierarchical Clustering

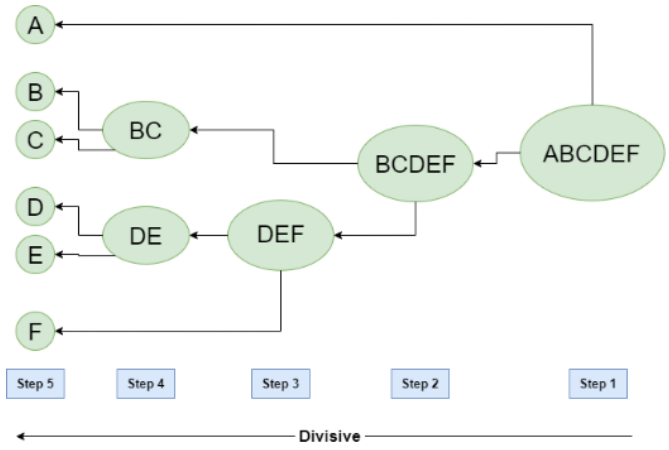
Μία ακόμη διαδεδομένη μέθοδος για την συσταδοποίηση **χρονικών σειρών** και **χωρικών** δεδομένων είναι η **Ιεραρχική** συσταδοποίηση (Hierarchical Clustering) [16]. Η συγκεκριμένη τεχνική δημιουργεί μια **ιεραρχία εμφωλευμένων συστάδων**, η οποία μπορεί να αναπαρασταθεί γραφικά μέσω ενός **δενδρογράμματος**. Μπορεί να αναπτυχθεί με **δύο** τρόπους είτε με την **συσσωρευτική** μέθοδο είτε με την **διαιρετική** μέθοδο.

Στην περίπτωση που ακολουθείται η **συσσωρευτική** μέθοδος κάθε δεδομένο απαρτίζει μία **ξεχωριστή** συστάδα. Σε κάθε βήμα οι **κοντινές** συστάδες **συγχωνεύονται** έως ούτε δημιουργηθεί **μία** μόνο συστάδα ή ένα **συγκεκριμένο πλήθος** συστάδων.



Εικόνα 3: Οπτική αναπαράσταση του αλγορίθμου Hierarchical με συσσωρευτική μέθοδο

Στην **δεύτερη** περίπτωση όλα τα δεδομένα είναι σε μία **ενιαία** συστάδα και σε κάθε βήμα δημιουργούνται **μικρότερες** συστάδες, έως ότου κάθε συστάδα να περιέχει **ένα μόνο δείγμα** ή τον **επιθυμητό** αριθμό.



Εικόνα 4: Οπτική αναπαράσταση του αλγορίθμου Hierarchical με διαιρετική μέθοδο

Ένα **κρίσιμο** στοιχείο αυτών των αλγορίθμων είναι η επιλογή του **κριτηρίου σύνδεσης**, το οποίο **καθορίζει** πώς υπολογίζεται η **απόσταση** ή η **ομοιότητα** μεταξύ των συστάδων κατά τη διαδικασία της συγχώνευσης. Τα **βασικά κριτήρια** σύνδεσης είναι:

1. η **μονή σύνδεση**, όπου η απόσταση ορίζεται ως η μικρότερη μεταξύ οποιουδήποτε ζεύγους σημείων από διαφορετικές συστάδες
2. η **μέση σύνδεση**, όπου λαμβάνεται ο μέσος όρος όλων των αποστάσεων μεταξύ σημείων των συστάδων
3. η **πλήρης σύνδεση**, όπου χρησιμοποιείται η μέγιστη απόσταση μεταξύ ζευγών σημείων
4. η **απόσταση μεταξύ κεντροειδών** (centroid linkage), όπου υπολογίζεται ως η απόσταση μεταξύ των κεντροειδών τους, δηλαδή των μέσων όρων των σημείων κάθε συστάδας

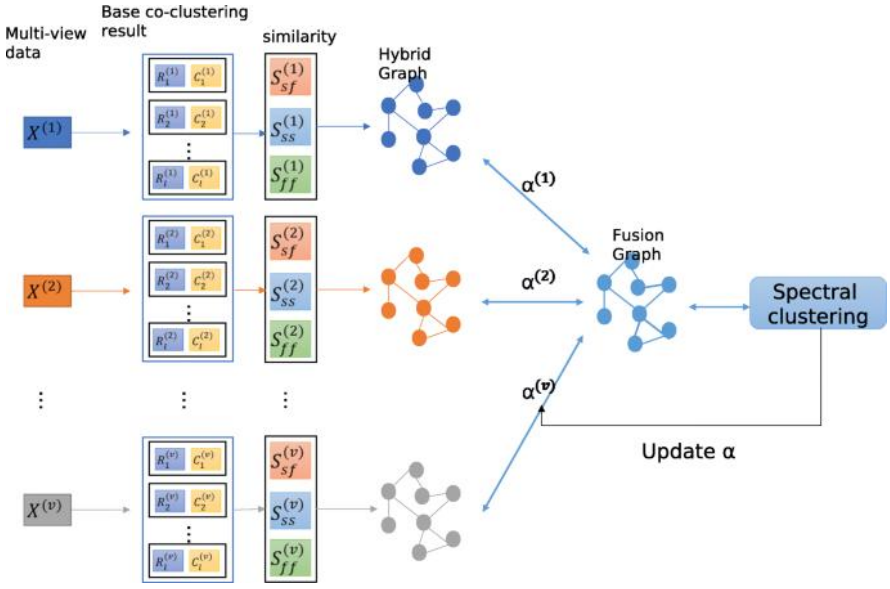
Η μέθοδος αυτή δημιουργεί μία **ιεραρχική δομή** [17], η οποία επιτρέπει την **οπτική αναπαράσταση** και την καλύτερη **κατανόηση** των δεδομένων. Ακόμη **δεν** απαιτεί **προκαθορισμένο** αριθμό συστάδων, το οποίο την καθιστά πιο **εύκολη** και ταυτόχρονα μπορεί να χρησιμοποιηθεί σε **πολλά είδη δεδομένων** χωρίς περιορισμό (π.χ. εικόνες, έγγραφα, γενετικά δεδομένα, κ.ά.). Ορισμένα **αρνητικά** σημεία της συγκεκριμένης συσταδοποίησης είναι ότι απαιτούνται **ισχυροί υπολογιστικοί πόροι** και οι **ερμηνεία** των αποτελεσμάτων είναι ιδιαίτερα **δύσκολη** σε **μεγάλα**

δεδομένα. Η ιεραρχική συσταδοποίηση **μπορεί** να χρησιμοποιηθεί στο Milano Dataset, καθώς μπορεί να **συσχετίσει γειτονικές περιοχές σε διαφορετικά επίπεδα** και μπορεί να **αναγνωρίσει μοτίβα και τάσεις** σε διαφορετικές χρονικές στιγμές ή διαστήματα.

Multivariate Similarity Clustering

Τέλος, μια **καινοτόμος** μέθοδος για συσταδοποίηση **χρονοσειρών** είναι η **Temporal and Multivariate Similarity Clustering** (TMSC), η οποία έχει σχεδιαστεί για την ανάλυση δεδομένων **τηλεπικοινωνιακών δικτύων πέμπτης γενιάς** [16]. Αποτελεί **κράμα** των τεχνικών **Dynamic Time Warping** (DTW) και **Spectral Clustering** επιτρέποντας την ομαδοποίηση κυψελών με **πολλαπλούς δείκτες απόδοσης** (KPIs). Για την εφαρμογή της συγκεκριμένης τεχνικής απαιτείται **μείωση του θορύβου** κατά την **προεπεξεργασία**. Στην συνέχεια η **DTW**, με **περιορισμένο μήκος παραμόρφωσης**, επιτρέπει τον **υπολογισμό** της **ομοιότητας** μεταξύ των **χρονοσειρών**. Τέλος, αξιοποιείται η **Spectral Clustering** στις **ήδη υπολογισμένες αποστάσεις** για ομαδοποίηση των δεδομένων.

Η συγκεκριμένη τεχνική είναι πολύ **ικανοποιητική** και εμφανίζει **μεγάλη απόδοση**. Μπορεί να εφαρμοστεί και σε **άλλους τομείς**, αλλά είναι ιδιαίτερα αποδοτική στην περίπτωση των δικτύων τηλεπικοινωνιών. Παρόλα αυτά **δεν** είναι σχεδιασμένη για **χωρικά** δεδομένα και λόγω της **χωροχρονικής φύσης** του dataset, προτείνεται άλλη τεχνική, η οποία θα αξιοποιεί και τα χωρικά χαρακτηριστικά.



Εικόνα 5: Οπτική αναπαράσταση της μεθόδου TMSC

Ερώτημα 4ο

Η γνώση των **clusters δραστηριότητας** που εξήχθησαν από την ανάλυση των **τηλεπικοινωνιακών** και **χωρικών** δεδομένων στο clustering, αποτελεί ένα εξαιρετικά χρήσιμο εργαλείο για τον **σχεδιασμό και τη βελτιστοποίηση ενός δικτύου 5G**. Η τεχνολογία 5G, σε αντίθεση με τα προηγούμενα δίκτυα, στηρίζεται έντονα στη **δυναμική και προσαρμοστική διαχείριση πόρων**, στην **υψηλή ευρυζωνικότητα** και στην **υποστήριξη μαζικών ταυτόχρονων συνδέσεων**, γεγονός που καθιστά τον **προγνωστικό σχεδιασμό κρίσιμο**.

Μέσω της **χαρτογράφησης** των περιοχών σε **λειτουργικά clusters** (οικιστικές ζώνες με βραδινή χρήση, εμπορικές περιοχές με έντονη ημερήσια δραστηριότητα, περιοχές με χαμηλή ή σποραδική χρήση), οι πάροχοι μπορούν να σχεδιάσουν ένα **ευφυές και αποδοτικό** δίκτυο με τα εξής **πρακτικά οφέλη**:

- **Δυναμική Κατανομή Πόρων (Network Slicing & Load Balancing):**

  Η γνώση των χρονικών προτύπων δραστηριότητας επιτρέπει στο δίκτυο να αναθέτει **διαφορετικές λωρίδες εύρους (slices)** σε κάθε περιοχή ανάλογα με την ώρα. Για παράδειγμα, οι εμπορικές περιοχές μπορούν να έχουν ενισχυμένη κατανομή φασματικών πόρων στις εργάσιμες ώρες, ενώ οι κατοικημένες περιοχές να ενισχύονται τις βραδινές. Αυτό ελαχιστοποιεί τη σπατάλη πόρων και **βελτιστοποιεί την απόδοση του δικτύου ανά ζώνη και χρονική περίοδο**.

- **Βελτίωση Κάλυψης σε Περιοχές Υψηλής Χρήσης:**

  Τα clusters που παρουσιάζουν **συστηματικά υψηλή χρήση** (σε SMS, κλήσεις ή δεδομένα) μπορούν να σηματοδοτήσουν περιοχές όπου απαιτείται εγκατάσταση επιπλέον **μικροκεραιών (small cells), beamforming ή Massive MIMO τεχνικών** για να αυξηθεί η φασματική αποδοτικότητα και η χωρητικότητα της κυψέλης. Αυτό οδηγεί σε **εστιασμένη επένδυση**, μειώνοντας το κόστος κάλυψης σε περιοχές όπου πραγματικά χρειάζεται.

- **Διαχείριση Συμφόρησης (Congestion Management):**

  Οι πληροφορίες για **ώρες αιχμής** ανά grid επιτρέπουν τη **διάγνωση και πρόληψη συμφόρησης**. Για παράδειγμα, αν εντοπίζονται περιοχές όπου παρατηρείται spike στην κίνηση τα Σαββατοκύριακα ή τις νύχτες, τότε μπορούν να ενεργοποιούνται **προσωρινές ενισχύσεις**

ισχύος ή να **επανακατανέμονται χρήστες** σε γειτονικά δίκτυα με λιγότερη φόρτιση.

- **Τοποθέτηση Υποδομών και Edge Servers:**

Η ανάλυση clustering μπορεί να οδηγήσει στον εντοπισμό περιοχών με **χαμηλό latency ανάγκες** (π.χ. εμπορικά hubs με πολλές συσκευές IoT ή streaming). Εκεί μπορεί να τοποθετηθεί **edge υπολογιστική υποδομή**, βελτιώνοντας σημαντικά την απόκριση των υπηρεσιών.

- **Σχεδιασμός για Energy Efficiency:**

Σε περιοχές με **χαμηλή ή ακανόνιστη χρήση**, το δίκτυο μπορεί να εφαρμόζει **energy-saving τεχνικές** (π.χ. sleep modes για κεραιοσυστήματα), μειώνοντας την κατανάλωση χωρίς να επηρεάζει την ποιότητα υπηρεσίας.

Υλοποιητικό μέρος

Ερώτημα 5ο – Προεπεξεργασία Δεδομένων

Το Milano Dataset περιλαμβάνει δεδομένα σχετικά με το τηλεπικοινωνιακό δίκτυο στην πόλη Milano, για **συγκεκριμένο** χρονικό διάστημα. Για την παρούσα εργασία επιλέχθηκαν οι ημερομηνίες:

- **14 – 12 – 2013**
- **15 – 12 – 2013**
- **16 – 12 – 2013**
- **17 – 12 – 2013**
- **18 – 12 – 2013**

Οι τιμές του dataset αναλύονται στο **2ο ερώτημα** του θεωρητικού μέρους.

Για την ανάλυση των δεδομένων αρχικά εισάγονται οι απαραίτητες βιβλιοθήκες και δημιουργείται ένα dataframe, ώστε να καταχωρηθούν τα **δεδομένα της ημέρας** και άλλο ένα για να διαχωριστούν τα **δεδομένα με βάση την ώρα καταγραφής**. Η συγκεκριμένη ενέργεια πραγματοποιείται για να διευκολύνει την κατανόηση της δομής των δεδομένων.

Στην συνέχεια, τα δεδομένα υποβάλλονται σε **προεπεξεργασία**. Η προεπεξεργασία των δεδομένων είναι κρίσιμη διαδικασία που επηρεάζει την **ακρίβεια** των μοντέλων μηχανικής μάθησης.

Αρχικά επιλέγονται οι στήλες:

```
#data columns
col_list = ['gridID', 'timeInterval', 'countryCode', 'smsIn',
'smsOut', 'callIn', 'callOut', 'internet']
```

οι οποίες περιέχουν τα δεδομένα **δραστηριότητας** που θα αξιοποιηθούν. Επιπλέον γίνεται **μετατροπή** της χρονικής σήμανσης **timeInterval** από **milliseconds** σε **ημερομηνία/ώρα**, από **UTC** σε **τοπική ώρα** (Κεντρική Ευρώπη – CET) και **δεν** συμπεριλαμβάνεται η **πληροφορία ζώνης ώρας.** Όσες στήλες κρίνεται ότι είναι **μη αξιοποιήσιμες αφαιρούνται**.

```
#remove useless columns
read_data.drop(columns=['timeInterval', 'countryCode'],
inplace=True)
```

Τα δεδομένα **δραστηριότητας** ομαδοποιούνται με βάση την **ημέρα** και στην συνέχεια με βάση την **ώρα** που παρατηρείται δραστηριότητα στην **επιλεγμένη περιοχή**. Η **ημερήσια** δραστηριότητα **αποκαλύπτει** εάν είναι ή όχι **έντονη** η δραστηριότητα σε κάθε περιοχή **καθ' όλη την διάρκεια των ημερών**, ενώ η **ωριαία** δημιουργεί **μοτίβα** δραστηριότητας **κατά τη διάρκεια της ημέρας**. Η συγκεκριμένη ομαδοποίηση προτείνεται και στο **repository** που συνοδεύει τα δεδομένα.

```
#daily groupby
read_data_daily = read_data.groupby(['gridID',
pd.Grouper(key='startTime', freq='D')]).sum()
    dailyGridActivity = pd.concat([dailyGridActivity,
read_data_daily])
    dailyGridActivity = dailyGridActivity.groupby(['gridID',
'startTime']).sum()

#hourly groupby
read_data_hourly = read_data.groupby(['gridID',
pd.Grouper(key='startTime', freq='H')]).sum()
    hourlyGridActivity = pd.concat([hourlyGridActivity,
read_data_hourly])
    hourlyGridActivity = hourlyGridActivity.groupby(['gridID',
'startTime']).sum()
```

Τα δεδομένα **συνενώνονται** σε δύο πίνακες, στον **dailyGridActivity**, για την **ημερήσια** ανάλυση, και στον **hourlyGridActivity**, για την **ωριαία** ανάλυση.

Επιπλέον, υπολογίζεται η **συνολική** δραστηριότητα για **κάθε gridID** (cell) **αθροίζοντας** όλες τις **ημερήσιες τιμές**. Για καλύτερη ανάλυση της συνολικής δραστηριότητας και για αξιοποίηση των δεδομένων στην δημιουργία στατιστικών **προστίθενται** νέες στήλες **σύνοψης sms** και **call** αντί **για smsIn και smsOut, CallIn και CallOut** στους πίνακες δραστηριότητας, ώστε να αναλυθεί η **συνολική επικοινωνία** ανά τύπο.

Linear Interpolation

Στην συνέχεια, γίνεται **συμπλήρωση τιμών** στις στήλες που περιέχουν **μηδενικά** και **κενά** σε μορφή **string 'NaN'**, με την μέθοδο **Linear Interpolation.** Έτσι, ορίζονται οι στήλες, στις οποίες θα γίνει συμπλήρωση των τιμών.

```
#columns to clean
columns_to_clean = ['smsIn', 'smsOut', 'callIn', 'callOut',
'internet', 'sms', 'call']
```

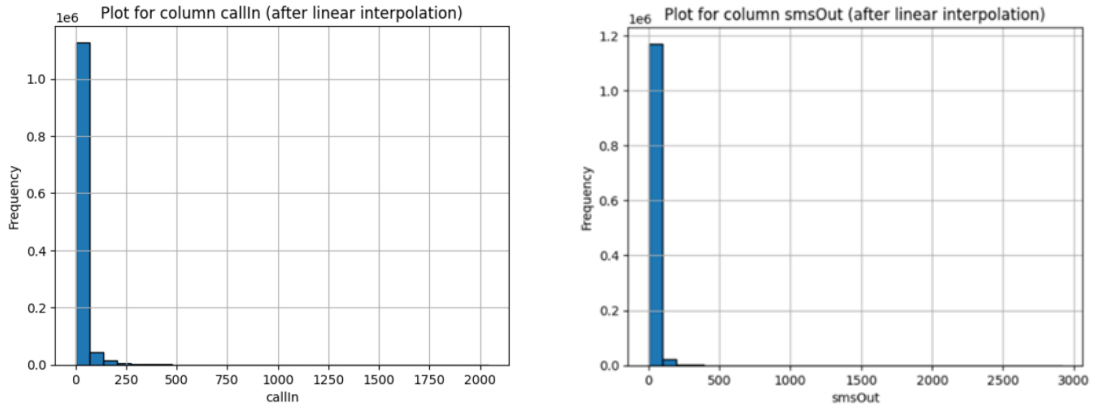
Γίνεται **ταξινόμηση** των εγγραφών με **χρονολογική σειρά** και έπειτα **συμπληρώνονται** με linear interpolation οι 'NaN' τιμές, δηλαδή υπολογίζονται οι τιμές **ανάμεσα σε υπάρχουσες**.

```
#sort by time values
 if 'datetime' in df.columns:
      df = df.sort_values(by='datetime')

#linear interpolation
  df[columns_to_clean] =
df[columns_to_clean].interpolate(method='linear')
```

Τέλος, αν έχουν **απομείνει** κενές τιμές χρησιμοποιεί την **επόμενη διαθέσιμη** τιμή και την **προηγούμενη**.

```
#forward/backward fill for NaN lvalues left
  df[columns_to_clean] =
df[columns_to_clean].fillna(method='bfill').fillna(method='ffill'
)
```

Ενδεικτικά ακολουθούν **ορισμένες** γραφικές παραστάσεις, αποτέλεσμα του linear interpolation.



Εικόνα 6: Histograms για τα features callIn, smsOut μετά το interpolation

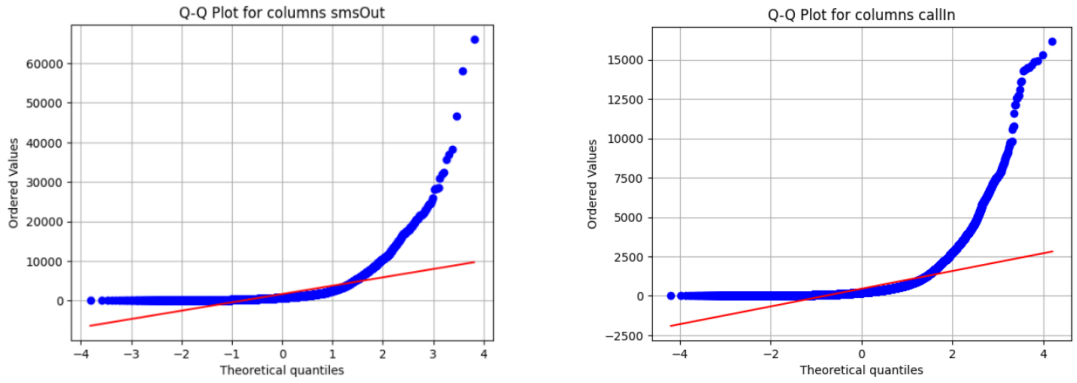
Έλεγχος κανονικής κατανομής

Γίνεται **έλεγχος** εάν τα δεδομένα **ακολουθούν κανονική κατανομή** για τις στήλες με **αριθμητικά** δεδομένα δραστηριότητας σε **πολλαπλά DataFrames**. Ο συγκεκριμένος έλεγχος είναι **απαραίτητος** σε περιπτώσεις στατιστικής ανάλυσης και σε τεχνικές μηχανικής μάθησης, που υποθέτουν κανονική κατανομή.

Ο έλεγχος γίνεται με τις συναρτήσεις **Kolmogorov-Smirnov test** και **Zscore**, ενώ η οπτικοποίηση γίνεται με την μέθοδο **Q-Q plot.**

```
# Standardization z-score
standardized = zscore(col_data)

# Kolmogorov-Smirnov Test
stat, p = kstest(standardized, 'norm')
if p > 0.05:
   print(f"{col}: Normal Distribution (p = {p:.4f})")
else:
   print(f"{col}: Non-Normal Distribution (p = {p:.4f})")

# Q-Q Plot
probplot(col_data, dist="norm", plot=plt)
plt.title(f"Q-Q Plot for columns {col}")
```

Ενδεικτικά ακολουθούν εκτυπώσεις των Q-Q Plot για τις τιμές **callIn** και **smsOut**.



Εικόνα 7: Q-Q Plots για τα features callIn, smsOut

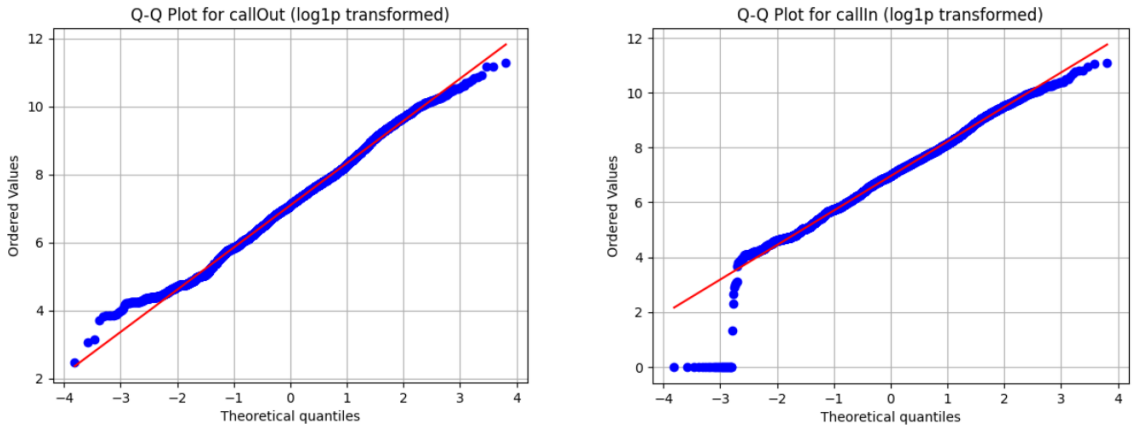
Βελτίωση ποιότητας κατανομής

Για την **βελτίωση** της ποιότητας της κατανομής και για τον περιορισμό των ακραίων τιμών επιλέγεται η εφαρμογή της μεθόδου **log1p** και γίνεται οπτικοποίηση των αποτελεσμάτων με Q-Q Plot.

```
#log1p for each column
for i, df in enumerate(dfs):
  print(f"\nDataFrame: {df_names[i]}")
  for col in columns_to_transform:
    df[f'{col}_log'] = np.log1p(df[col])

#Q-Q plots
for col in columns_to_transform:
```

```
    plt.figure()
    stats.probplot(df[f'{col}_log'].dropna(), dist="norm",
plot=plt)
    plt.title(f"Q-Q Plot for {col} (log1p transformed)")
    plt.grid(True)
    plt.show()
```

Ενδεικτικά ακολουθούν εκτυπώσεις των Q-Q Plot για τις τιμές **callIn** και **smsOut** με εφαρμογή της μεθόδου **log1p**.



Εικόνα 8: Q-Q Plots για τα features callIn, smsOut μετά την βελτίωση ποιότητας της κατανομής

Capping IQR – Επεξεργασία Outliers

Για την επεξεργασία των **ακραίων τιμών** (outliers) εφαρμόζεται η μέθοδος **Capping**. Το capping με βάση το IQR **περιορίζει τις ακραίες τιμές** (outliers) εντός **καθορισμένων ορίων** για **κάθε στήλη**. Υπολογίζονται τα **Q1, Q3** και η διαφορά **IQR** ώστε να εντοπιστούν οι τιμές **εκτός** εύρους. Οι τιμές **κάτω** από το **κατώτερο** ή **πάνω** από το **ανώτερο** όριο **αντικαθίστανται** με αυτά τα **όρια**. Έτσι μειώνεται η επίδραση των outliers, ειδικά σε ευαίσθητους αλγόριθμους όπως ο K-Means.

Έτσι ορίζεται:

$$\boldsymbol{IQR = Q3 - Q1},$$

όπου **Q1** είναι το **πρώτο τεταρτημόριο** και **Q3** το **τρίτο τεταρτημόριο**.

Ορίζει το όριο **κάτω** από το οποίο μια τιμή θεωρείται outlier:

$$\boldsymbol{Lower\ Bound = Q1 - 1.5 \times IQR}$$

Ορίζει το όριο **πάνω** από το οποίο μια τιμή θεωρείται outlier:

$$\boldsymbol{Upper\ Bound = Q3 + 1.5 \times IQR}$$

Η **τελική αντικατάσταση** των ακραίων τιμών δίνεται από τον εξής τύπο:

$$Xi = \begin{cases} \text{Lower Bound}, & \text{αν } Xi < \text{Lower Bound} \\ Xi, & \text{αν Lower Bound} \leq Xi \leq \text{Upper Bound} \\ \text{Upper Bound}, & \text{αν } Xi > \text{Upper Bound} \end{cases}$$

Ακολουθεί το απόσπασμα αντίστοιχου κώδικα για τον υπολογισμό του IQR:

```
#find IQR
      Q1 = np.percentile(df[column].dropna(), 25)
      Q3 = np.percentile(df[column].dropna(), 75)
      IQR = Q3 - Q1
      lower_bound = Q1 - 1.5 * IQR
      upper_bound = Q3 + 1.5 * IQR
```

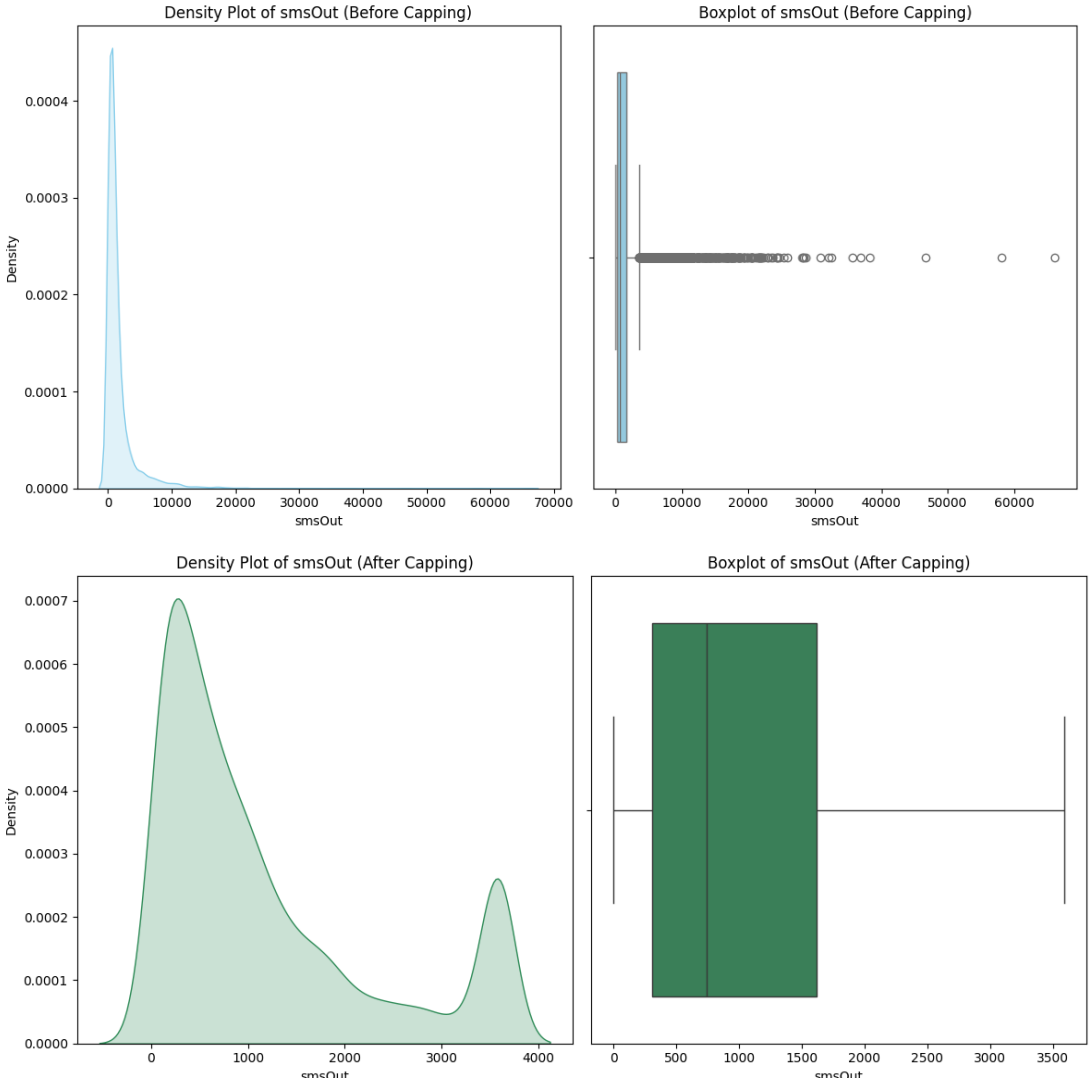
Τέλος γίνεται αντικατάσταση των τιμών που **ξεπερνούν** το **άνω άκρο**, και των τιμών που βρίσκονται **υπό του κάτω άκρου**.

```
# Capping
      df[column] = np.where(df[column] < lower_bound,
lower_bound, df[column])
      df[column] = np.where(df[column] > upper_bound,
upper_bound, df[column])
```

Αντίστοιχα ακολουθούν **ενδεικτικά** ορισμένες **BoxPlot** και **DensityPlot πριν** και **μετά** το capping για το feature **SmsOut**, για την οπτικοποίηση των τιμών.

```
# Density Plot andBoxplot after capping
      plt.figure(figsize=(12, 6))

      # Density after
      plt.subplot(1, 2, 1)
```

```
        sns.kdeplot(df[column].dropna(), fill=True,
color="seagreen")
        plt.title(f'Density Plot of {column} (After Capping)')
        plt.xlabel(column)

        # Boxplot after
        plt.subplot(1, 2, 2)
        sns.boxplot(x=df[column], color="seagreen")
        plt.title(f'Boxplot of {column} (After Capping)')
        plt.xlabel(column)
        plt.tight_layout()
        plt.show()
```



Εικόνα 9: Density plot για smsOut πριν το Capping, Box plot για smsOut μετά το Capping

Ερώτημα 6ο – Επιλογή Χαρακτηριστικών Για Clustering

Για την ομαδοποίηση των κυψελών του δικτύου μέσω μεθόδων clustering, είναι κρίσιμο να επιλεγούν **κατάλληλα χαρακτηριστικά** που να αποτυπώνουν **ουσιαστικά πρότυπα χρήσης**. Η επιλογή έγινε με γνώμονα τη **συνδυασμένη ανάλυση χωρικών και χρονικών χαρακτηριστικών**, ώστε να προκύπτουν clusters **ερμηνεύσιμα**, που αντανακλούν **πραγματικές διαφοροποιήσεις** στη συμπεριφορά των χρηστών.

Επιλέχθηκαν μεταβλητές που προκύπτουν από τις **ημερήσιες και ωριαίες τιμές δραστηριότητας**, όπως ο **συνολικός αριθμός εξερχόμενων και εισερχόμενων SMS και κλήσεων** (sms, call), ο **συνολικός όγκος δεδομένων από χρήση Internet**, **η μέση τιμή και τυπική απόκλιση** ανά κυψέλη, η **ώρα** της ημέρας (hours) και η **ημέρα** της εβδομάδας (weekdayFlag) για τη μελέτη **μοτίβων** καθημερινής και εβδομαδιαίας χρήσης.

Αυτά τα χαρακτηριστικά βοηθούν στον εντοπισμό διαφορών μεταξύ κυψελών με **εντατική** ημερήσια δραστηριότητα (π.χ. εμπορικά κέντρα), **βραδινή** ή **περιοδική** χρήση (π.χ. οικιστικές περιοχές), ή **χαμηλής κινητικότητας** (π.χ. πάρκα, απομακρυσμένες περιοχές).

```
dailyGridActivity.reset_index(inplace=True)
hourlyGridActivity.reset_index(inplace=True)

hourlyGridActivity['weekdayFlag'] =
hourlyGridActivity['startTime'].dt.dayofweek
hourlyGridActivity['hours'] =
hourlyGridActivity['startTime'].dt.hour
dailyGridActivity['weekdayFlag'] =
dailyGridActivity['startTime'].dt.dayofweek
dailyGridActivity['hours'] =
dailyGridActivity['startTime'].dt.hour
```

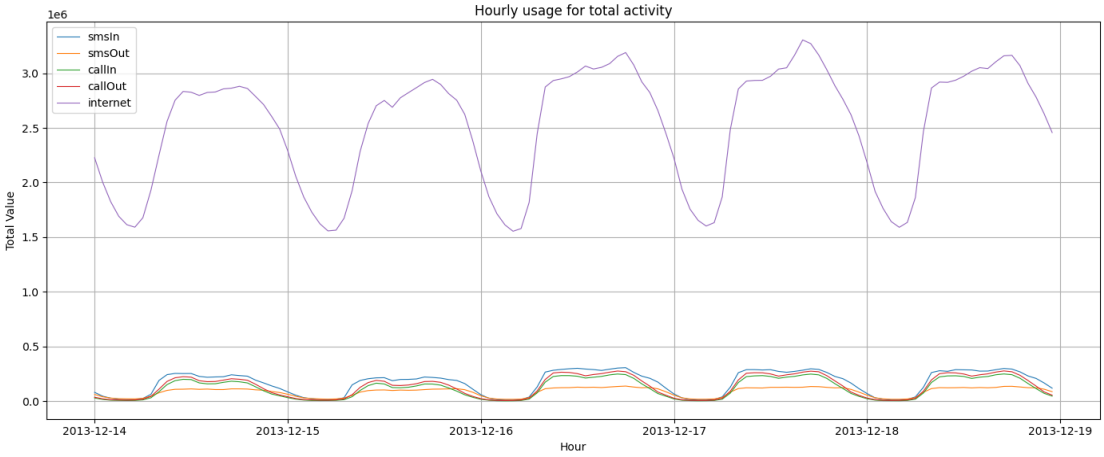
Αρχικά, γίνεται **ομαδοποίηση** (groupby) των δεδομένων με βάση την **startTime**, και υπολογίζεται το **άθροισμα** των τιμών για **κάθε κατηγορία δραστηριότητας** (εισερχόμενα/εξερχόμενα SMS και κλήσεις, και χρήση internet) **ανά ημέρα**. Η **πρώτη** οπτικοποίηση δημιουργεί ένα **γραμμικό διάγραμμα** όπου στον **οριζόντιο άξονα** εμφανίζεται η **ημερομηνία** και στον **κάθετο** ο **συνολικός όγκος** κάθε δραστηριότητας, επιτρέποντας έτσι την παρακολούθηση **μακροχρόνιων τάσεων** και εντοπισμό περιόδων με υψηλή ή χαμηλή χρήση.

```
#group daily data based on startTime
daily_features_trend =
dailyGridActivity.groupby('startTime')[['smsIn', 'smsOut',
'callIn', 'callOut', 'internet']].sum()

plt.figure(figsize=(14, 6))
for column in daily_features_trend.columns:
    plt.plot(daily_features_trend.index,
daily_features_trend[column], label=column)

plt.title('Daily Usage - all activities')
plt.xlabel('Date')
plt.ylabel('Total Value')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

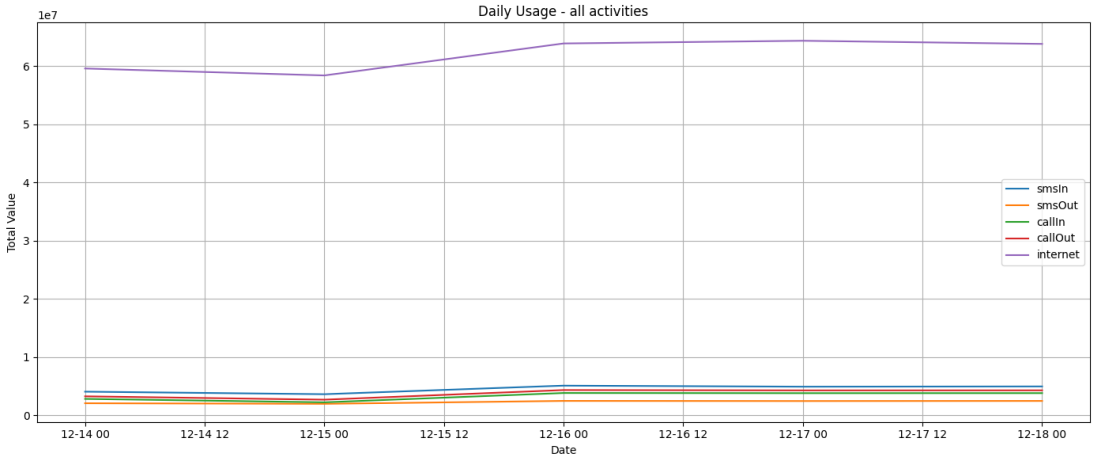
Στη συνέχεια, εφαρμόζεται **αντίστοιχη λογική** και στο **ωριαίο** dataset, προσφέροντας **λεπτομερέστερη ανάλυση** ανά ώρα. Το γράφημα που προκύπτει εμφανίζει τη **μεταβολή** της χρήσης **εντός της ημέρας**, επιτρέποντας την παρατήρηση **συνηθειών** όπως ώρες αιχμής. Η απεικόνιση αυτή βοηθά στον εντοπισμό μοτίβων ή ανωμαλιών στη χρήση, προσφέροντας χρήσιμη πληροφορία για την επιλογή κατάλληλων χαρακτηριστικών που θα χρησιμοποιηθούν στις μεθόδους clustering.

```
#group hourly data based on startTime
hourly_features_trend =
hourlyGridActivity.groupby('startTime')[['smsIn', 'smsOut',
'callIn', 'callOut', 'internet']].sum()

plt.figure(figsize=(14, 6))
for column in hourly_features_trend.columns:
    plt.plot(hourly_features_trend.index,
hourly_features_trend[column], label=column, linewidth=0.8)

plt.title('Hourly usage for total activity')
plt.xlabel('Hour')
plt.ylabel('Total Value')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

Ακολουθούν αντίστοιχες **ενδεικτικές** γραφικές παραστάσεις.



Εικόνα 10: Μεταβολή της δραστηριότητας ανά ώρα της ημέρας



Εικόνα 11: Μεταβολή της δραστηριότητας ανά ημέρα

Η διαδικασία **εξαγωγής χαρακτηριστικών** (feature engineering), που ακολουθεί, αποσκοπεί στην **περιγραφή** της κινητής δραστηριότητας ανά γεωγραφική μονάδα (gridID), με βάση τα ημερήσια και ωριαία δεδομένα που έχουν **ομαδοποιηθεί κατάλληλα** στα dataframes **dailyGridActivity** και **hourlyGridActivity**.

Αρχικά, προσδιορίζονται οι **αριθμητικές στήλες** του **ημερήσιου πίνακα** για μελλοντική χρήση στους υπολογισμούς. Έπειτα, γίνεται **διαχωρισμός** των δεδομένων ανά **τύπο ημέρας**, δηλαδή σε **καθημερινές** (WD – weekday) και **Σαββατοκύριακα** (WE – weekend), κάτι που είναι κρίσιμο για την κατανόηση της **διακύμανσης** της ανθρώπινης δραστηριότητας κατά τη διάρκεια της εβδομάδας.

```
numeric_cols =
dailyGridActivity.select_dtypes(include=np.number).columns.tolist
()

#seperate days
hourlyGridActivity_WE =
hourlyGridActivity[hourlyGridActivity['weekdayFlag'].isin([5,6])]
hourlyGridActivity_WD =
hourlyGridActivity[~hourlyGridActivity['weekdayFlag'].isin([5,6])
]
dailyGridActivity_WE =
dailyGridActivity[dailyGridActivity['weekdayFlag'].isin([5,6])]
dailyGridActivity_WD =
dailyGridActivity[~dailyGridActivity['weekdayFlag'].isin([5,6])]

# create gridActivity DataFrame
gridActivity = pd.DataFrame()
```

Στη συνέχεια, κατασκευάζεται ένα **νέο dataframe** με όνομα **gridActivity**, το οποίο θα περιέχει **όλα τα νέα χαρακτηριστικά**. Για **κάθε υποσύνολο** (WD και WE) και **κάθε τύπο δραστηριότητας** (sms, call, internet), ορίζεται **group** κατά **gridID** και **hours**, ώστε να υπολογιστεί ο **μέσος όρος δραστηριότητας ανά ώρα**. Από αυτό εξάγονται για κάθε grid οι **μέγιστες** (Max), **ελάχιστες** (Min) και **μέσες** (Avg) τιμές της **ωριαίας** δραστηριότητας. Αυτά τα χαρακτηριστικά καταγράφουν την **ενδοημερήσια μεταβλητότητα**, επιτρέποντας την κατηγοριοποίηση περιοχών σε **ζωντανές** (με έντονη διακύμανση ώρα με την ώρα) ή πιο **σταθερές** περιοχές.

```
#Hourly Statistics

for df_in, prefix in [(hourlyGridActivity_WE, "WE"),
(hourlyGridActivity_WD, "WD")]:

    df = df_in.reset_index()

    for col in ['sms', 'call', 'internet']:

        grouped = df.groupby(['gridID','hours'])[[col]].mean()

        grouped.reset_index(level=1, drop=True, inplace=True)

        max_ = grouped.groupby('gridID')[[col]].max()

        min_ = grouped.groupby('gridID')[[col]].min()
```

```
        avg_ = grouped.groupby('gridID')[[col]].mean()

        max_.columns = [f'hourly{col}Max_{prefix}']

        min_.columns = [f'hourly{col}Min_{prefix}']

        avg_.columns = [f'hourly{col}Avg_{prefix}']

        gridActivity = pd.concat([gridActivity, max_, min_,
avg_], axis=1)
```

Έπειτα, ακολουθεί ο **διαχωρισμός** μεταξύ **ημερήσιας** και **νυχτερινής** δραστηριότητας. Ορίζεται το χρονικό διάστημα **08:00–21:59** ως **ημέρα** και **00:00–07:59** ως **νύχτα**. Γίνεται **ομαδοποίηση** και **άθροιση** των τιμών **sms, call και internet για κάθε grid στις δύο αυτές χρονικές περιόδους**, παράγοντας νέα χαρακτηριστικά όπως **totalSmsDay_WD** και **totalInternetNight_WE**. Τα χαρακτηριστικά αυτά είναι ιδιαιτέρως χρήσιμα για να διαπιστωθεί αν μια περιοχή είναι **κατοικημένη**, **εμπορική**, ή **βιομηχανική**, με βάση τις ώρες αιχμής της δραστηριότητας.

```
#Day & Night activity

for    df_in,    prefix    in    [(hourlyGridActivity_WE,    "WE"),
(hourlyGridActivity_WD, "WD")]:

# Daytime 08:00-21:59

      mask_day = (df_in.hours >= 8) & (df_in.hours < 22)

day                                                              =
df_in[mask_day].groupby('gridID')[['sms','call','internet']].sum(
)

day.columns = [f'totalSmsDay_{prefix}', f'totalCallDay_{prefix}',
f'totalInternetDay_{prefix}']

# Night 00:00-07:59

mask_night = (df_in.hours >= 0) & (df_in.hours < 8)

night                                                            =
df_in[mask_night].groupby('gridID')[['sms','call','internet']].su
m()

night.columns             =             [f'totalSmsNight_{prefix}',
f'totalCallNight_{prefix}', f'totalInternetNight_{prefix}']
```

```
gridActivity = pd.concat([gridActivity, day, night], axis=1)
```

Ακόμη, υπολογίζονται επιπλέον **αριθμητικές αναλογίες**, οι οποίες παρέχουν χρήσιμες ενδείξεις για τη **συμπεριφορά** χρήσης. Ειδικότερα, υπολογίζεται η αναλογία **smsIn/smsOut**, **callIn/callOut**, **συνολικών SMS προς συνολικές κλήσεις**, και η αναλογία **χρήσης internet προς το σύνολο των SMS και κλήσεων**. Αυτές οι τιμές είναι ενδεικτικές του τύπου επικοινωνίας που **προτιμάται** σε κάθε περιοχή και μπορούν να συμβάλουν στον διαχωρισμό διαφορετικών μοτίβων χρήσης.

```
df = dailyGridActivity.groupby('gridID')[numeric_cols].sum()

gridActivity['dailySmsIn/dailySmsOut'] = df['smsIn'] /
(df['smsOut'] + 1)

gridActivity['dailyCallIn/dailyCallOut'] = df['callIn'] /
(df['callOut'] + 1)

gridActivity['dailySms/dailyCall'] = (df['smsIn'] + df['smsOut'])
/ (df['callIn'] + df['callOut'] + 1)

gridActivity['dailyInternet/dailySmsCall'] = df['internet'] /
(df['smsIn'] + df['smsOut'] + df['callIn'] + df['callOut'] + 1)
```

Χρησιμοποιείται και η βοηθητική συνάρτηση **daily_stats** για την παραγωγή χαρακτηριστικών **ημερήσιας στατιστικής σύνοψης**. Για κάθε grid και ανάλογα με το αν πρόκειται για **WE** ή **WD**, υπολογίζονται οι **μέγιστες**, **ελάχιστες** και **μέσες ημερήσιες τιμές** των **sms, call, internet.** Αυτό προσθέτει σημαντικό **βάθος**, αναδεικνύοντας περιοχές που παρουσιάζουν είτε **υψηλές αιχμές** είτε **σταθερή συμπεριφορά**. Τα δεδομένα **ανασυντάσσονται εβδομαδιαία** και υπολογίζονται οι **μέσες τιμές** (weeklyAvg), οι **μεταβολές από εβδομάδα σε εβδομάδα** (weeklyChange), καθώς και οι **συνολικές μέγιστες, ελάχιστες και μέσες τιμές δραστηριότητας ανά grid** (weeklyMax, weeklyMin, weeklyAvg2). Το χαρακτηριστικό **weeklyAvgdiff** για κάθε δραστηριότητα λειτουργεί ως δείκτης **δυναμικότητας/μεταβλητότητας** στην περιοχή.

```
#Daily Stats WE/WD

def daily_stats(df, prefix):

    max_ = df.groupby('gridID')[['sms','call','internet']].max()

    min_ = df.groupby('gridID')[['sms','call','internet']].min()
```

```
    avg_ = df.groupby('gridID')[['sms','call','internet']].mean()

    max_.columns   =   [f'smsMax_{prefix}',   f'callMax_{prefix}',
f'internetMax_{prefix}']

    min_.columns   =   [f'smsMin_{prefix}',   f'callMin_{prefix}',
f'internetMin_{prefix}']

    avg_.columns   =   [f'smsAvg_{prefix}',   f'callAvg_{prefix}',
f'internetAvg_{prefix}']

    return pd.concat([max_, min_, avg_], axis=1)



gridActivity = pd.concat([

    gridActivity,

    daily_stats(dailyGridActivity_WE, 'WE'),

    daily_stats(dailyGridActivity_WD, 'WD')

], axis=1)
```

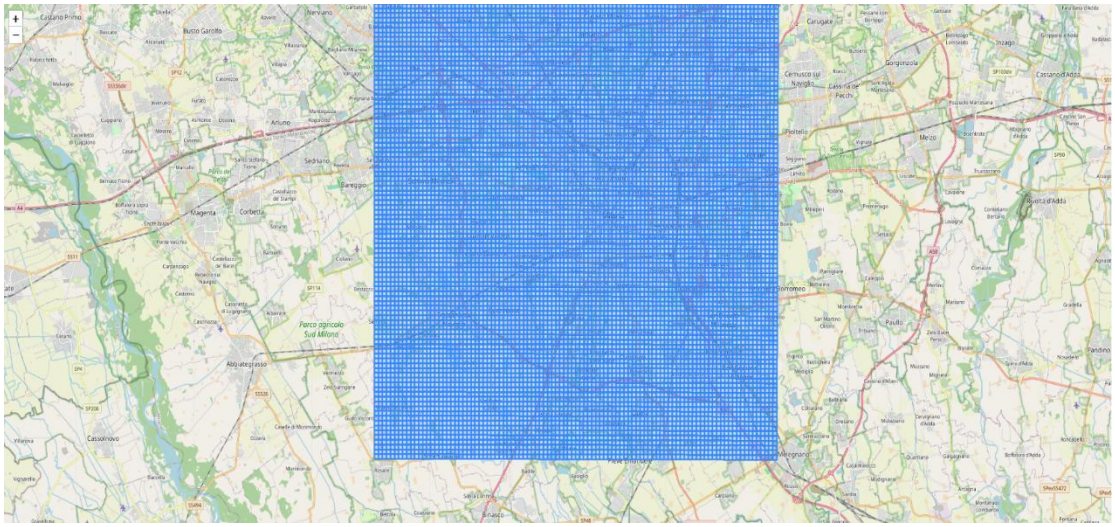
Μετά την **εξαγωγή τον χρονικών χαρακτηριστικών**, γίνεται **εξαγωγή των γεωχωρικών χαρακτηριστικών** από το αρχείο **GeoJSON** που περιέχει την **κατανομή πλέγματος** (grid) στην πόλη του Μιλάνου, και στη **συγχώνευση** αυτών των χαρακτηριστικών με τα δεδομένα δραστηριότητας για κάθε κελί του πλέγματος. Αρχικά, φορτώνεται το GeoJSON αρχείο και αποθηκεύεται σε μορφή λεξικού. Έπειτα, **για κάθε κελί** (feature) **εξάγονται** το **μοναδικό** του **ID**, ο **τύπος γεωμετρίας** και οι **συντεταγμένες του πολυγώνου**. Από αυτές υπολογίζονται το **γεωμετρικό κέντρο** (κεντροειδές), το **εμβαδόν**, η **περίμετρος**, η **γεωμετρική πυκνότητα compactness**(ως δείκτης κυκλικότητας), ο **αριθμός κορυφών** και η **απόσταση** του κελιού από το **ιστορικό κέντρο** του Μιλάνου. Οι υπολογισμοί γίνονται μέσω της βιβλιοθήκης **shapely**, η οποία επιτρέπει τις ακριβείς γεωμετρικές πράξεις.

```
#extract spatial features

grid_data = []
```

```
milan_center = Point(9.19, 45.46)  #Milan historical center(apprx.)

for feature in geojson_data["features"]:

    grid_id = feature["properties"]["cellId"]

    geometry_type = feature["geometry"]["type"]

    coordinates = feature["geometry"]["coordinates"][0]

    #Centroid

    longitudes = [pt[0] for pt in coordinates]

    latitudes = [pt[1] for pt in coordinates]

    center_lon = sum(longitudes) / len(longitudes)

    center_lat = sum(latitudes) / len(latitudes)

    #Polygon object

    polygon = Polygon(coordinates)

    #Feature engineering

    area = polygon.area

    perimeter = polygon.length

    compactness = 4 * np.pi * area / (perimeter ** 2) if perimeter
!= 0 else 0

    n_vertices = len(coordinates)



distance_from_center = polygon.centroid.distance(milan_center)
```

Όλα αυτά τα χαρακτηριστικά **συλλέγονται** στην **λίστα λεξικών** (grid_data) και στο τέλος **μετατρέπονται** σε **pandas DataFrame** (geo_df). Τέλος, όλα τα χαρακτηριστικά αποθηκεύονται σε ένα **merged_df**, το οποίο στη συνέχεια **συγχωνεύεται** με τον πίνακα **gridActivity** βάσει του gridID, ώστε να **εμπλουτιστούν** τα **δεδομένα δραστηριότητας** με **πληροφορίες** για τη **θέση**, το **σχήμα** και την **απόσταση** κάθε κελιού από το κέντρο.

```
grid_data.append({
        'gridID': grid_id,
        'geometry_type': geometry_type,
        'longitude': center_lon,
        'latitude': center_lat,
        'coordinates': coordinates,
        'area': area,
        'perimeter': perimeter,
        'compactness': compactness,
        'n_vertices': n_vertices,
        'distance_from_center': distance_from_center
    })
geo_df = pd.DataFrame(grid_data)
#merge with gridActivity
merged_df = gridActivity.merge(geo_df, on="gridID", how="left")
```



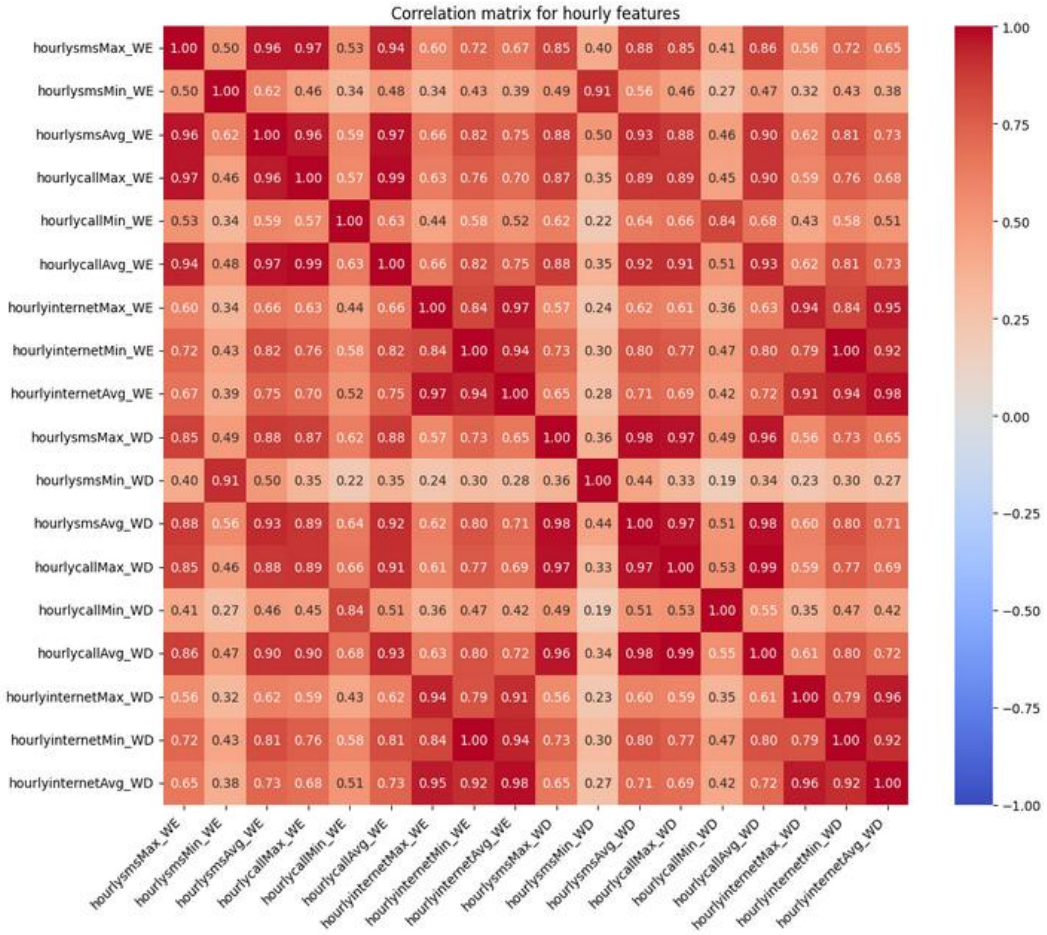
Εικόνα 12: Οπτικοποίηση της περιοχής του κέντρου στην οποία θα γίνει ομαδοποίηση των cells

Statistical Analysis Of Temporal Features

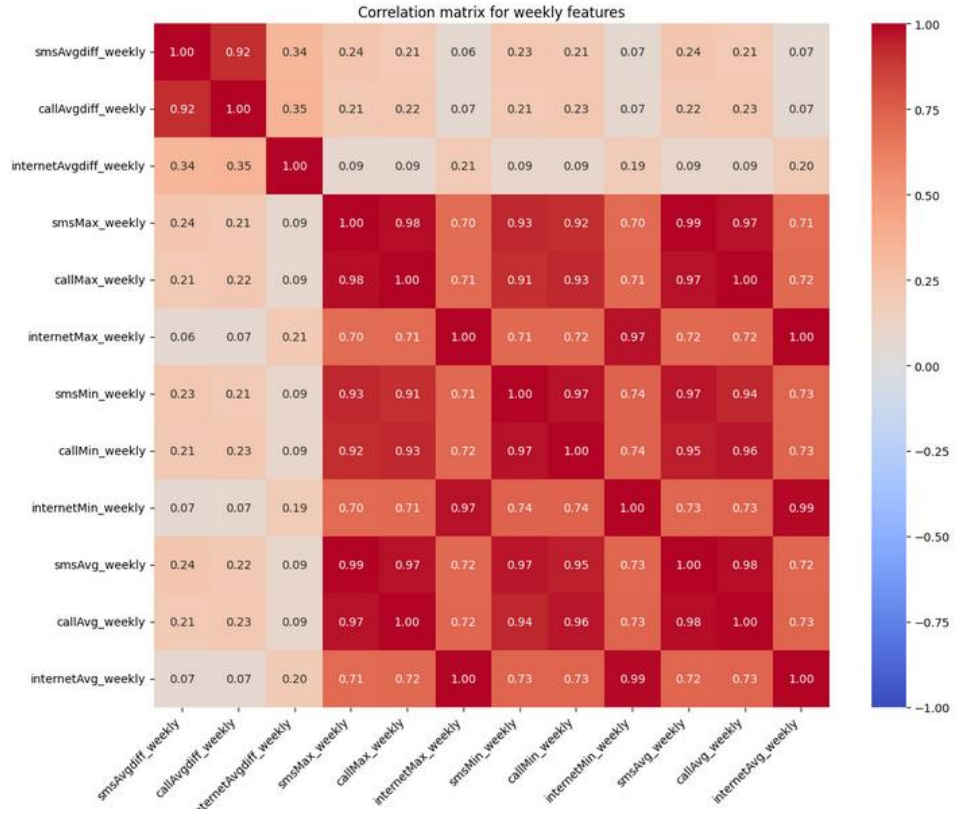
Χρησιμοποιείται η συνάρτηση **plot_feature_correlation**, η οποία δημιουργεί και εμφανίζει το **correlation matrix** μεταξύ των **χαρακτηριστικών** του **gridActivity DataFrame** που **ξεκινούν** με ένα συγκεκριμένο **πρόθεμα** (prefix), όπως "**hourly**", "**weekly**" ή "**daily**". Συγκεκριμένα, επιλέγει όλες τις στήλες του πίνακα df gridActivity που περιέχουν το δοσμένο πρόθεμα, υπολογίζει τον **πίνακα συσχέτισης** και τον απεικονίζει με τη μορφή **heatmap**. Η συνάρτηση καλείται **τρεις φορές** για **διαφορετικές χρονικές κλίμακες**.

```
def plot_feature_correlation(prefix, df=gridActivity):
    selected = df[[col for col in df.columns if prefix in col]]
    corr = selected.corr()

    plt.figure(figsize=(12,10))
    sns.heatmap(corr, annot=True, fmt=".2f", cmap='coolwarm',
vmin=-1, vmax=1)
    plt.title(f"Correlation matrix for {prefix} features")
    plt.xticks(rotation=45, ha='right')
    plt.yticks(rotation=0)
    plt.tight_layout()
    plt.show()
```

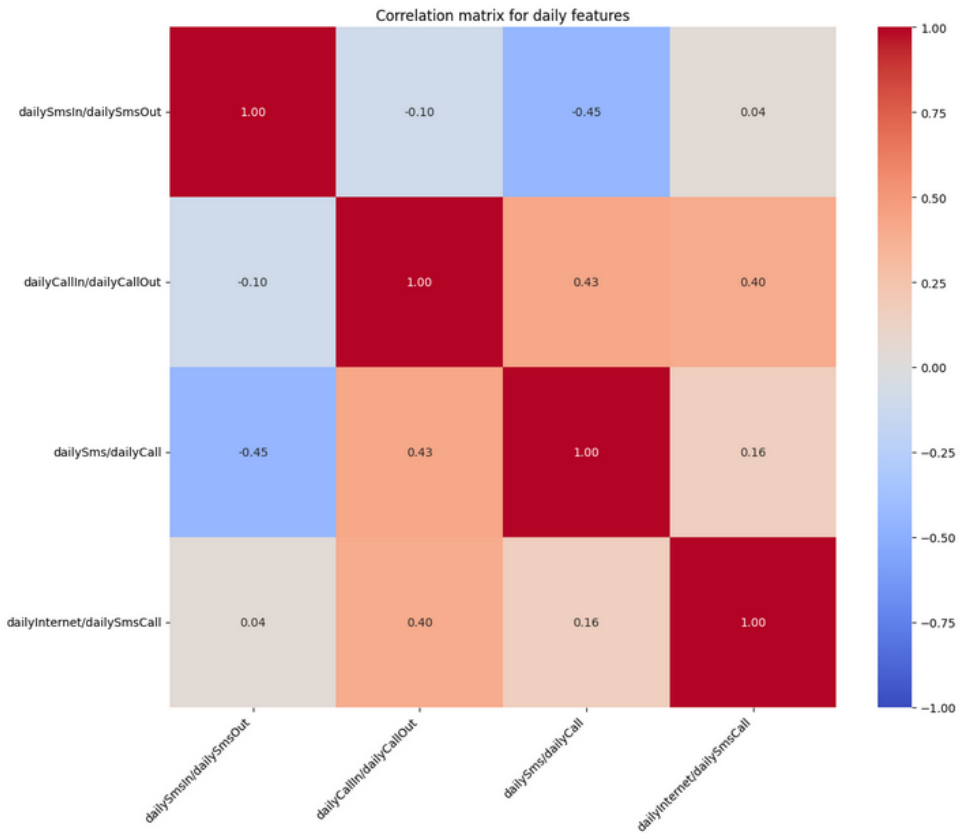
Παρακάτω παρουσιάζονται τα **heatmaps** για τα διάφορα **χρονικά χαρακτηριστικά** του **merged_df**:



Εικόνα 13: Correlation matrix για τα ωριαία χαρακτηριστικά



Εικόνα 14: Correlation matrix για τα εβδομαδιαία χαρακτηριστικά



Εικόνα 15: Correlation matrix για τα ημερήσια χαρακτηριστικά

Με βάση τον **βαθμό συσχέτισης** που προκύπτει από τα heatmaps, **επιλέγονται** συγκεκριμένα **χρονικά χαρακτηριστικά** από τον πίνακα **gridActivity**, τα οποία έχουν εντοπιστεί ως **σημαντικά**. Τα επιλεγμένα χαρακτηριστικά **χωρίζονται** σε **τρεις** κατηγορίες:

- **Μέσες τιμές ανά ώρα** (hourly), ξεχωριστά για **εργάσιμες ημέρες** (WD) και **Σαββατοκύριακα** (WE)
- **Εβδομαδιαία χαρακτηριστικά**, που περιλαμβάνουν τόσο τις **μέσες τιμές** όσο και τις **διαφορές** μεταξύ εργάσιμων και μη εργάσιμων ημερών (Avgdiff_weekly)
- **Ημερήσιες αναλογίες** (daily ratios), που δείχνουν **σχέσεις** μεταξύ εισερχόμενων/εξερχόμενων SMS και κλήσεων, αλλά και τη σχέση μεταξύ internet και άλλων υπηρεσιών.

Αυτά τα χαρακτηριστικά έχουν επιλεχθεί επειδή **παρουσιάζουν σημαντικές διαφορές ή συσχετίσεις**, ενδεικτικές μοτίβων ανθρώπινης δραστηριότητας. Ο τελικός **πίνακας gridActivity_selected** περιέχει μόνο τις **επιλεγμένες στήλες** και δημιουργείται ως αντίγραφο (.copy()) για ασφαλή επεξεργασία χωρίς να αλλοιωθεί ο αρχικός πίνακας gridActivity.

```
#selected time geatures based on heatmaps
selected_columns = [
    # Hourly (Avg only)
    'hourlysmsAvg_WE', 'hourlycallAvg_WE',
'hourlyinternetAvg_WE',
    'hourlysmsAvg_WD', 'hourlycallAvg_WD',
'hourlyinternetAvg_WD',

    # Weekly (Avg + Avgdiff)
    'smsAvgdiff_weekly', 'callAvgdiff_weekly',
'internetAvgdiff_weekly',
    'smsAvg_weekly', 'callAvg_weekly', 'internetAvg_weekly',

    # Daily ratios
    'dailySmsIn/dailySmsOut', 'dailyCallIn/dailyCallOut',
    'dailySms/dailyCall', 'dailyInternet/dailySmsCall'
]

#df with selected features
gridActivity_selected = gridActivity[selected_columns].copy()
```

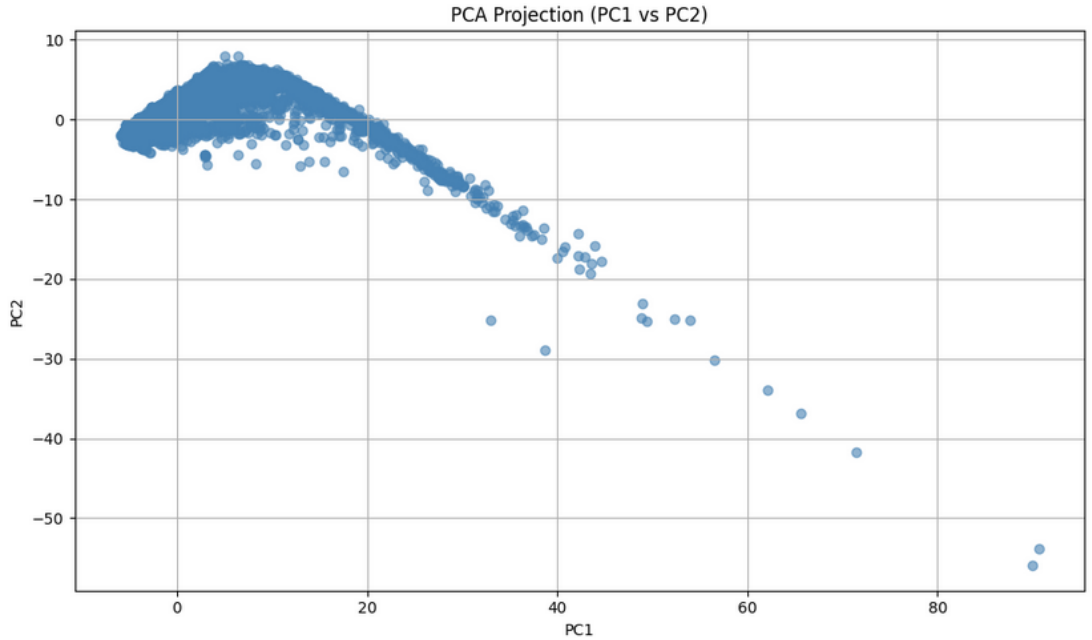
Προκειμένου να **ελεγχθεί** εάν τα χαρακτηριστικά που επιλέχθηκαν **συμβάλλουν περισσότερο στην διακύμανση των δεδομένων** σε ρεαλιστικό υπόβαθρο, εφαρμόζεται ένας **έλεγχος PCA** με **έμφαση** στις **κύριες συνιστώσες**, ο οποίος εκφράζει **πόσο επηρεάζει** το κάθε χαρακτηριστικό τον συγκεκριμένο άξονα (PC), δηλαδή τη διάσταση της νέας μειωμένης αναπαράστασης.

Αρχικά, γίνεται **κανονικοποίηση** όλων των χαρακτηριστικών με χρήση του **StandardScaler**. Στη συνέχεια, με την μέθοδο PCA (n_components=0.95), επιλέγεται αυτόματα ο **ελάχιστος αριθμός κύριων συνιστωσών** (Principal Components) που εξηγούν **τουλάχιστον το 95% της συνολικής μεταβλητότητας** των δεδομένων. Οι τιμές των **loadings** (δηλαδή οι συντελεστές κάθε χαρακτηριστικού σε κάθε κύρια συνιστώσα) αποθηκεύονται σε έναν **πίνακα** με **χαρακτηριστικά ως γραμμές** και **PCs ως στήλες**. Με βάση τα απόλυτα loadings της **πρώτης κύριας συνιστώσας** (PC1), εντοπίζονται τα **10 χαρακτηριστικά** με τη **μεγαλύτερη επίδραση** σε αυτή, ενώ το **overall_contrib** δείχνει το **μέγιστο απόλυτο loading** κάθε χαρακτηριστικού σε **όλες** τις συνιστώσες.

```
#normalization
scaler = StandardScaler()
scaled = scaler.fit_transform(gridActivity)

#PCA keep 95% of variation
pca = PCA(n_components=0.95)
X_pca = pca.fit_transform(scaled)

#compute Loading
loadings = pca.components_.T  # shape: (n_features, n_components)
pca_features = pd.DataFrame(
    loadings,
    index=gridActivity.columns,
    columns=[f'PC{i+1}' for i in range(pca.n_components_)]
)

#top features for first Principal Component
contrib_pc1 =
pca_features['PC1'].abs().sort_values(ascending=False)
print("Top 10 features influencing 1o Principal Component:")
print(contrib_pc1.head(10))

#overall PCA contribution
overall_contrib =
pca_features.abs().max(axis=1).sort_values(ascending=False)
print("\nTop 10 features - total contribution:")
```

```
print(overall_contrib.head(10))
```

Ο άξονας **PC1** συγκεντρώνει τη **μεγαλύτερη διακύμανση**, καθώς τα περισσότερα σημεία έχουν τιμές PC1 **μεταξύ 0 και 30**, ενώ υπάρχει μια ομάδα **outliers** με τιμές PC1 που **ξεπερνούν το 80**. Αυτή η **απότομη αύξηση** δηλώνει ότι σε **λίγα μόνο κελιά** παρατηρούνται έντονα **διαφορετικά μοτίβα** δραστηριότητας, τα οποία κυριαρχούν στην πρώτη συνιστώσα. Ο άξονας **PC2 διαφοροποιεί** τα σημεία πιο **ήπια**, κυρίως στην **περιοχή γύρω από το μηδέν**, υποδηλώνοντας ότι οι **διαφορές** μεταξύ των κελιών είναι **λιγότερο έντονες** σε αυτή τη διάσταση. Το γεγονός ότι η κατανομή έχει υψηλό PC1 και αρνητικό PC2 μπορεί να υποδεικνύει περιοχές με **υψηλή δραστηριότητα αλλά μικρή ποικιλία τύπου** (π.χ. κυριαρχία μόνο ενός τύπου επικοινωνίας). Η κατανομή υπονοεί ύπαρξη υποομάδων στα δεδομένα, ενδεχομένως περιοχές με σταθερό, καθημερινό μοτίβο και άλλες με πιο έντονη, ανομοιογενή χρήση.



Εικόνα 16: Scatter plot των δύο πρώτων κύριων συνιστωσών (PC1 και PC2) από PCA

Τελικά, δημιουργείται η λίστα **final_features**, η οποία περιλαμβάνει όλα τα αρχικά **επιλεγμένα χρονικά χαρακτηριστικά** (selected_columns) μαζί με **τέσσερα επιπλέον**, τα οποία είναι τα ελάχιστα επίπεδα κλήσεων και SMS κατά τις εργάσιμες (WD) και μη εργάσιμες (WE) ημέρες. Η χρήση set εξασφαλίζει ότι **δεν** υπάρχουν **διπλότυπα**. Στη συνέχεια, δημιουργείται το **selected_features_df**, ένας νέος πίνακας που περιέχει **μόνο αυτά τα τελικά επιλεγμένα χαρακτηριστικά** από τον συγχωνευμένο πίνακα **merged_df.**

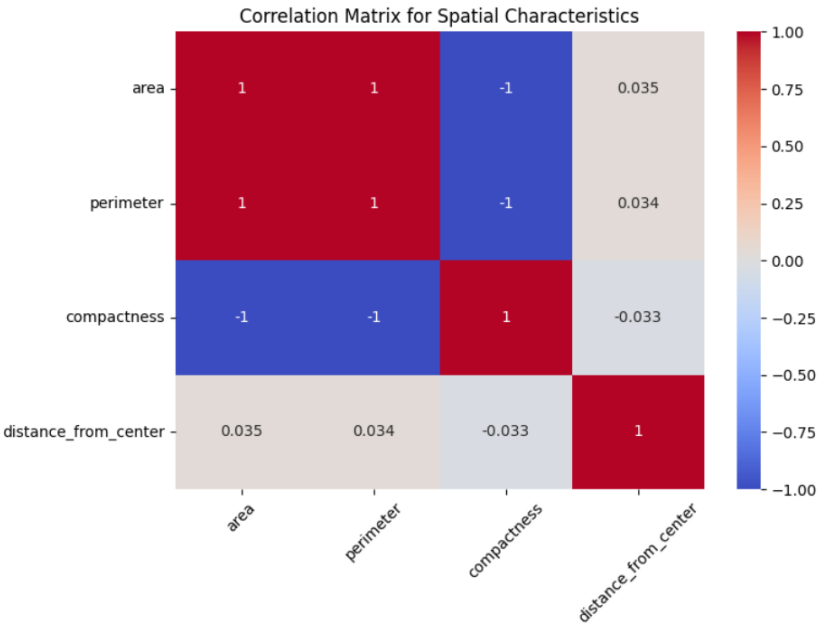
```
final_features = list(set(selected_columns) | {
    'hourlycallMin_WD', 'hourlycallMin_WE', 'hourlysmsMin_WD',
'hourlysmsMin_WE'
})

#final DataFrame with all selected time features
selected_features_df = merged_df[final_features]
```

Statistical Analysis Of Spatial Features

Επιλέγονται **τέσσερα αριθμητικά γεωχωρικά** χαρακτηριστικά από τον πίνακα merged_df, τα οποία είναι:

- το **εμβαδόν** (area)
- η **περίμετρος** (perimeter)
- πόσο **συμπαγές** είναι το γεωμετρικό **πολύγονο** (compactness)
- η **απόσταση** από το **ιστορικό κέντρο** του Μιλάνου (distance_from_center).

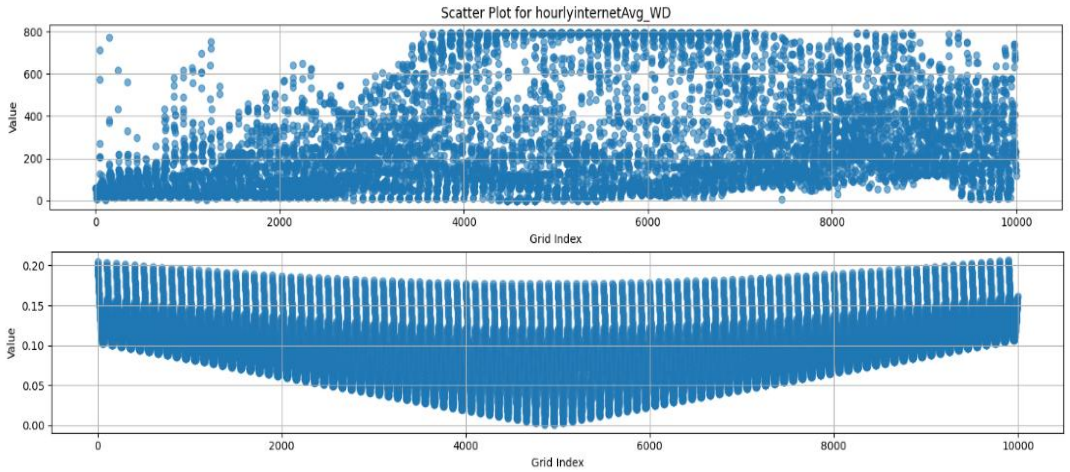
Στη συνέχεια, υπολογίζεται ξανά ο **πίνακας συσχέτισης** (corr_matrix) μεταξύ αυτών των μεταβλητών. Ο πίνακας αυτός επιτρέπει την **αξιολόγηση** του κατά πόσο **σχετίζονται** οι **γεωμετρικές ιδιότητες** μεταξύ τους, κάτι χρήσιμο για την κατανόηση της δομής του πλέγματος.



Εικόνα 17: Correlation matrix για τα χωρικά χαρακτηριστικά

Επιπρόσθετα, γίνεται **οπτικοποίηση** των **χωρικών χαρακτηριστικών** του πίνακα **merged_df** μέσω **scatter plots**, όπου κάθε χαρακτηριστικό εμφανίζεται σε ξεχωριστό υπό-διάγραμμα. Έτσι, γίνεται αντιληπτό ο τρόπος με τον οποίο **μεταβάλλονται** οι τιμές κάθε χαρακτηριστικού σε σχέση με το **index** του grid. Στη συνέχεια, ορίζεται η λίστα **selected_spatial_features** που περιέχει τα **βασικά γεωγραφικά δεδομένα**: area, compactness και distance_from_center. Ακολουθεί ο **συνδυασμός** αυτών με άλλες στήλες όπως hourlycallMin_WD, hourlycallMin_WE, hourlysmsMin_WD, hourlysmsMin_WE, καθώς και τις ήδη επιλεγμένες στήλες από **selected_columns**. Ο συνδυασμός γίνεται με χρήση **συνόλων** (sets) ώστε να αποφευχθούν τα **διπλότυπα**. Το αποτέλεσμα αποθηκεύεται στη λίστα **final_features**, η οποία περιλαμβάνει **όλες τις στήλες** που κρίνονται χρήσιμες για clustering. Τέλος, δημιουργείται ένα νέο DataFrame (selected_features_df) με βάση αυτές τις στήλες, το οποίο είναι έτοιμο να χρησιμοποιηθεί για τεχνικές ομαδοποίησης όπως KMeans, DBSCAN ή ιεραρχική ταξινόμηση.

```
selected_spatial_features = ['area', 'compactness',
'distance_from_center']

#merge all selected features
final_features = list(
    set(selected_columns)
    | {'hourlycallMin_WD', 'hourlycallMin_WE', 'hourlysmsMin_WD',
'hourlysmsMin_WE'}
    | set(selected_spatial_features)
)

#create final DataFrame for clustering
selected_features_df = merged_df[final_features]
```

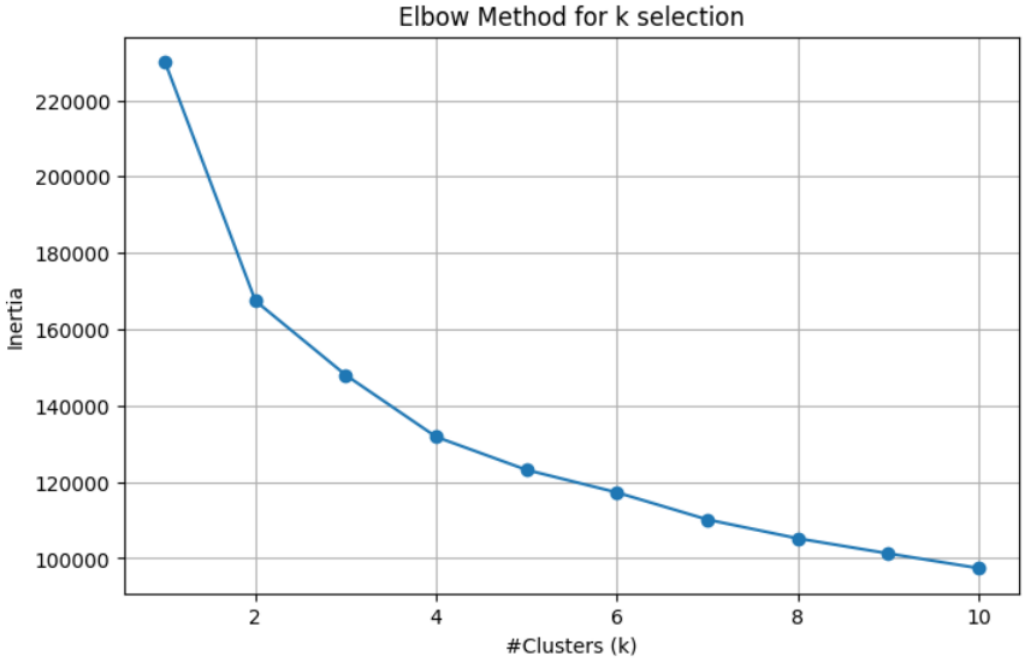


Εικόνα 18: Ενδεικτικά Scatter plots για χρονικά και χωρικά features

Ερώτημα 7ο – Εφαρμογή Μεθόδων Clustering

Μέθοδος K-Means

Πραγματοποιείται ομαδοποίηση (clustering) χωρικών και τηλεπικοινωνιακών δεδομένων με τη μέθοδο **KMeans**, ξεκινώντας με **κανονικοποίηση** των χαρακτηριστικών μέσω του **StandardScaler**, ώστε όλα τα χαρακτηριστικά να συμβάλλουν ισότιμα στον υπολογισμό αποστάσεων. Στη συνέχεια, χρησιμοποιείται η μέθοδος **Elbow** για να εντοπιστεί ο **βέλτιστος** αριθμός clusters (k), εφαρμόζοντας KMeans για τιμές του k από 1 έως 10 και καταγράφοντας την **αδράνεια (inertia)** – δηλαδή το άθροισμα των τετραγώνων αποστάσεων από τα κέντρα των clusters. Η αδράνεια θα σχεδιαστεί σε γράφημα στο οποίο θα εντοπιστεί το σημείο της καμπύλης, που αποτελεί την πιο καθαρή ένδειξη του ιδανικού αριθμού ομάδων. Στο επόμενο στάδιο, καθορίζεται k=4 και πραγματοποιείται ο τελικός υπολογισμός των clusters.

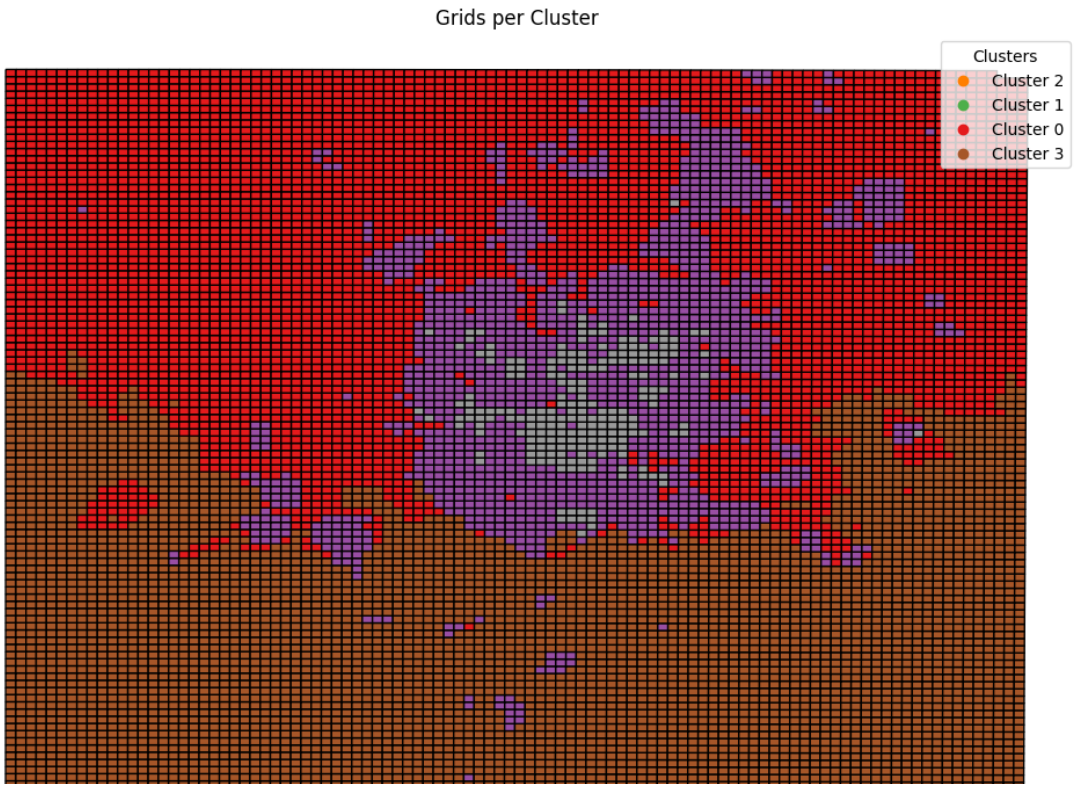


Εικόνα 19: Γράφημα Elbow method για επιλογή του k

Στην συνέχεια, οι ετικέτες των clusters προστίθενται στο **selected_features_df**, ώστε κάθε γραμμή (grid) να φέρει την ανάλογη ταξινόμηση. Παράλληλα, δημιουργείται **γεωχωρικό** DataFrame (geo_gdf) από τις συντεταγμένες των πολυγώνων που αντιπροσωπεύουν τα **grid cells**, χρησιμοποιώντας shapely.geometry.Polygon. Έπειτα, γίνεται **συσχέτιση** (merge) των **γεωχωρικών πολυγώνων** με τα **labels** των

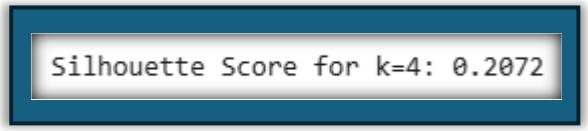
clusters με βάση το gridID, και προκύπτει το **geo_clusters**, που συνδυάζει τη γεωμετρία με την ομάδα στην οποία ανήκει κάθε grid.

Τέλος, γίνεται οπτικοποίηση των ομάδων πάνω στον χάρτη με χρωματική διακριτότητα και κάθε ομάδα έχει το δικό της χρώμα και ετικέτα στον χάρτη. Με τον τρόπο αυτό αποδίδεται **γεωχωρική ερμηνεία των clusters**, διευκολύνοντας την ανάλυση πρότυπων συμπεριφοράς ή τηλεπικοινωνιακής χρήσης ανά περιοχή.



Εικόνα 20: Γεωγραφική αναπαράσταση των clusters με την μέθοδο k-means

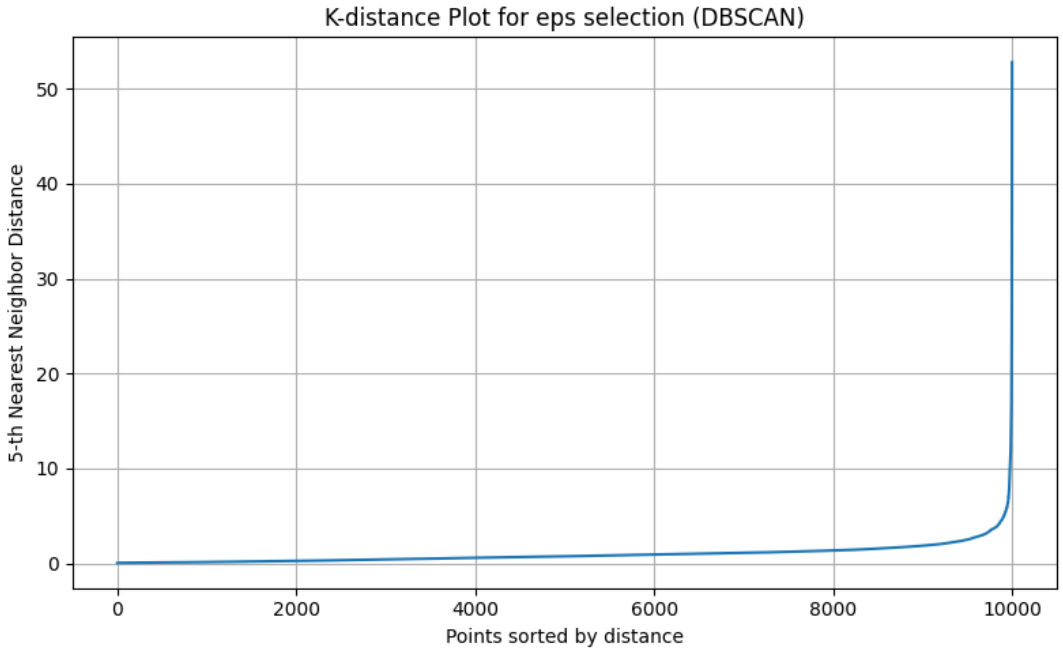
Για την αξιολόγηση την ποιότητα των ομάδων που προέκυψαν από το KMeans clustering αξιοποιείται το **Silhouette Score**, ένας δείκτης που μετρά πόσο καλά διαχωρίζονται οι ομάδες μεταξύ τους και πόσο συμπαγείς είναι. Η τιμή αυτή κυμαίνεται από -1 έως 1, όπου υψηλότερες τιμές δείχνουν καλύτερο διαχωρισμό. Στη συνέχεια, γίνεται **ομαδοποίηση των δεδομένων ανά cluster** και υπολογίζονται οι μέσες τιμές όλων των αριθμητικών χαρακτηριστικών για κάθε ομάδα. Έτσι, είναι εύκολο να γίνει εξαγωγή των **χρήσιμων στατιστικών προφίλ** για κάθε cluster.



Η τιμή **Silhouette Score = 0.2072** για το **K-Means clustering με k = 4** υποδηλώνει ότι ο διαχωρισμός των δεδομένων σε τέσσερα clusters είναι **μέτριας ποιότητας**. Ο Silhouette Score μετρά την **ομοιογένεια** και **διαχωρισιμότητα** των ομάδων, δηλαδή πόσο κοντά είναι κάθε σημείο στο δικό του cluster σε σύγκριση με τα γειτονικά clusters. Η τιμή 0.2072 είναι πάνω από το μηδέν, που σημαίνει ότι υπάρχει κάποια δομή στα δεδομένα, ωστόσο η ομαδοποίηση **δεν είναι ιδιαίτερα καθαρή**. Ειδικά στο πλαίσιο ενός dataset internet activity, όπου υπάρχουν περιοχές με πολύ **διαφορετικά προφίλ χρήσης** (αστικές-αγροτικές, υψηλή-χαμηλή δραστηριότητα), είναι πιθανό οι ομάδες να **επικαλύπτονται** ή τα σημεία να βρίσκονται κοντά στα **όρια** μεταξύ ομάδων. Αυτό εξηγεί τη μέτρια τιμή του score. Επίσης, το K-Means υποθέτει ότι οι συστάδες είναι **σφαιρικές** και **ισομεγέθεις**, κάτι που **σπάνια ισχύει στα δεδομένα πραγματικού κόσμου** όπως αυτά του διαδικτύου.

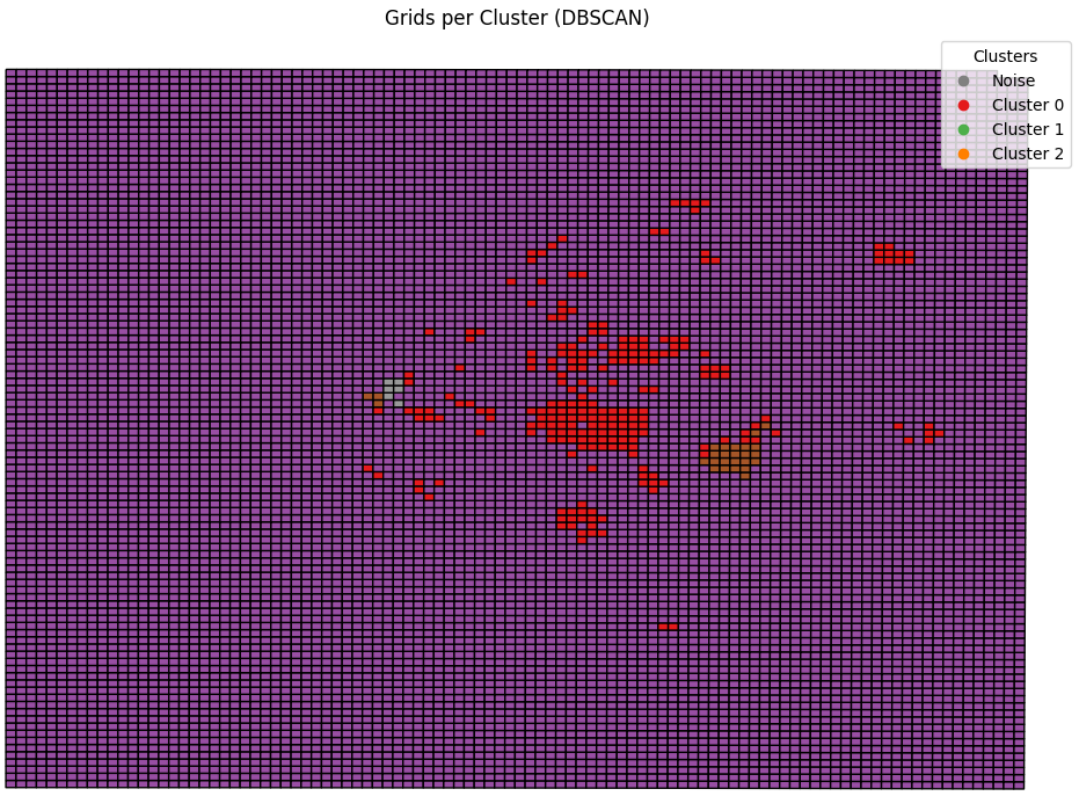
Μέθοδος DBSACN

Χρησιμοποιείται η μέθοδος **NearestNeighbors** για να δημιουργηθεί αρχικά ένα **k-distance γράφημα**, στο οποίο σχεδιάζεται η **απόσταση** κάθε σημείου από τον **5ο κοντινότερο γείτονά** του. Ο 5ος γείτονας δίνει πιο **αξιόπιστη εκτίμηση τοπικής πυκνότητας** από τους πρώτους, καθώς **ελαχιστοποιεί** την επίδραση **θορύβου** και **απομονωμένων σημείων** κατά την επιλογή της ακτίνας eps. Από αυτές τις αποστάσεις, επιλέγεται για κάθε σημείο η **πέμπτη μικρότερη απόσταση** και κατασκευάζεται ένα γράφημα όπου τα σημεία ταξινομούνται κατά **αύξουσα** σειρά. Το γράφημα αυτό βοηθά στην επιλογή της τιμής eps, που θα είναι η **μέγιστη ακτίνα απόστασης** για να θεωρηθούν σημεία γειτονικά. Το σημείο όπου παρατηρείται απότομη αλλαγή στη γραμμή, υποδηλώνει καλή τιμή για eps, που εδώ επιλέγεται ως **eps = 3**.



Εικόνα 21: K - distance γράφημα για την επιλογή της παραμέτρου eps

Στη συνέχεια, εφαρμόζεται ο DBSCAN για **eps = 3** και **min_samples = 5**, και κάθε σημείο **ταξινομείται** σε κάποιο cluster ή χαρακτηρίζεται ως **θόρυβος** αν δεν πληροί τις προϋποθέσεις. Ο αριθμός των συστάδων υπολογίζεται αφαιρώντας την ετικέτα -1, που αναπαριστά τα σημεία εκτός οποιασδήποτε ομάδας.



Εικόνα 22: Γεωγραφική αναπαράσταση των clusters με την μέθοδο DBSCAN

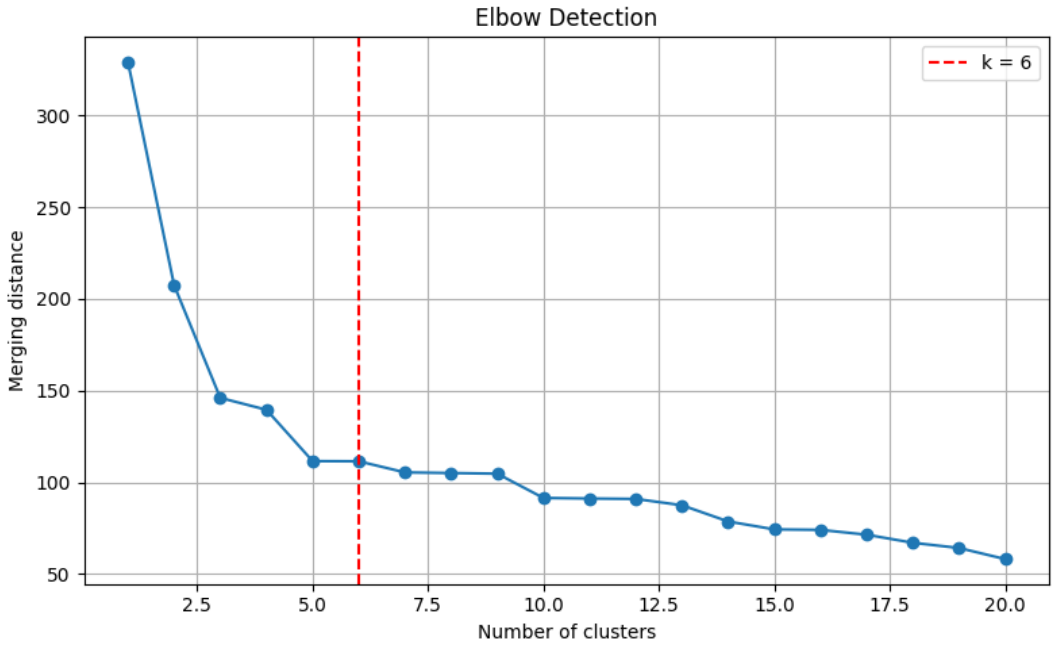
```
Number of clusters (without noise): 3
Silhouette Score: 0.5106
```

Αυτή η τιμή **Silhouette Score** θεωρείται **καλή**, καθώς υποδηλώνει ότι τα σημεία είναι, κατά μέσο όρο, **πιο κοντά** στο δικό τους cluster απ' ό,τι σε **άλλα**. Δηλαδή, ο διαχωρισμός μεταξύ των ομάδων είναι ικανοποιητικός και **δεν** υπάρχει σημαντική επικάλυψη. Ο DBSCAN έχει καταφέρει να εντοπίσει φυσικές, **πυκνές** συστάδες στα δεδομένα, διατηρώντας ταυτόχρονα μια λογική ποσότητα θορύβου **εκτός** ανάλυσης. Το γεγονός ότι προέκυψαν 3 ομάδες χωρίς να έχει καθοριστεί εκ των προτέρων ο αριθμός τους αποτελεί ένα βασικό πλεονέκτημα του DBSCAN.

Η μέθοδος αυτή είναι πολύ **αποτελεσματική** σε **internet activity** datasets, όπως το **milano dataset**, το οποίο περιέχει **αυθαίρετα** σχήματα και πυκνότητες, σε αντίθεση με το KMeans που απαιτεί σφαιρικές ομάδες και προκαθορισμένο αριθμό clusters.

Επιπλέον Μέθοδος - Hierarchical Clustering

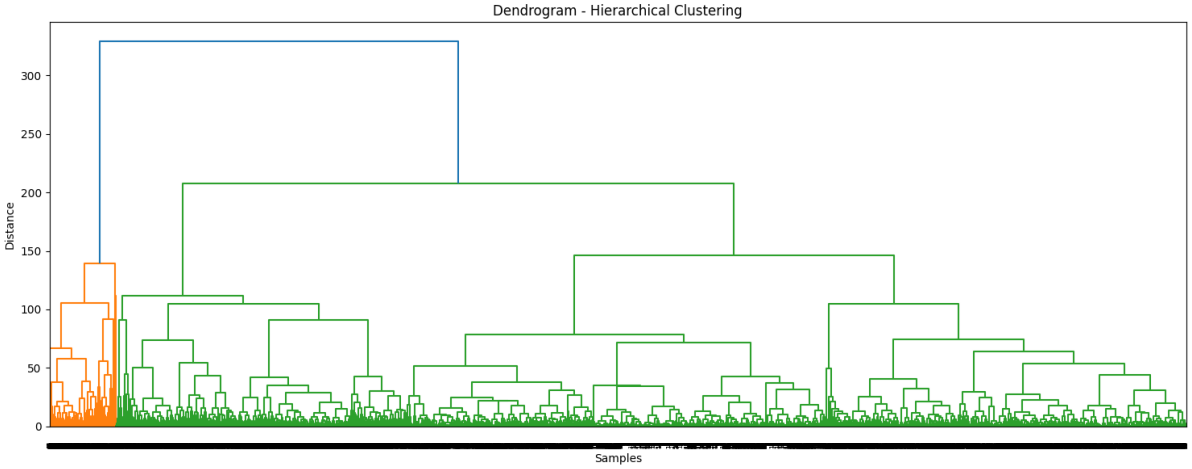
Αξιοποιήθηκε και **τρίτη** μέθοδος συσταδοποίησης, ώστε να υπάρχει **καλύτερο δείγμα σύγκρισης των αποτελεσμάτων** των μεθόδων συσταδοποίησης με βάση την έρευνα για την αποδοτικότητα τους που πραγματοποιήθηκε στο **θεωρητικό μέρος**. Χρησιμοποιείται η συνδετική μέθοδος **Ward linkage**, που ελαχιστοποιεί τη διακύμανση μέσα σε κάθε cluster καθώς ενώνονται τα σημεία. Αρχικά δημιουργείται ο **linkage πίνακας**, ο οποίος περιγράφει τη σειρά με την οποία συγχωνεύονται τα σημεία ή οι ομάδες στο clustering. Έπειτα, εξετάζονται τα τελευταία 20 βήματα συγχώνευσης από το linkage matrix, καθώς σε αυτά γίνονται οι πιο σημαντικές συγχωνεύσεις (δηλαδή αυτές με τις μεγαλύτερες αποστάσεις). Οι αποστάσεις αυτές αντιστρέφονται για να βρίσκονται σε αύξουσα σειρά και υπολογίζονται οι διαφορές (np.diff) μεταξύ τους, ώστε να βρεθεί το σημείο όπου η διαφορά της απόστασης είναι η μεγαλύτερη. Το σημείο αυτό θεωρείται η προτεινόμενη τιμή για k, που αποτυπώνεται γραφικά με κόκκινη διακεκομμένη γραμμή στο plot.



Εικόνα 23: Γράφημα Elbow method για επιλογή του k

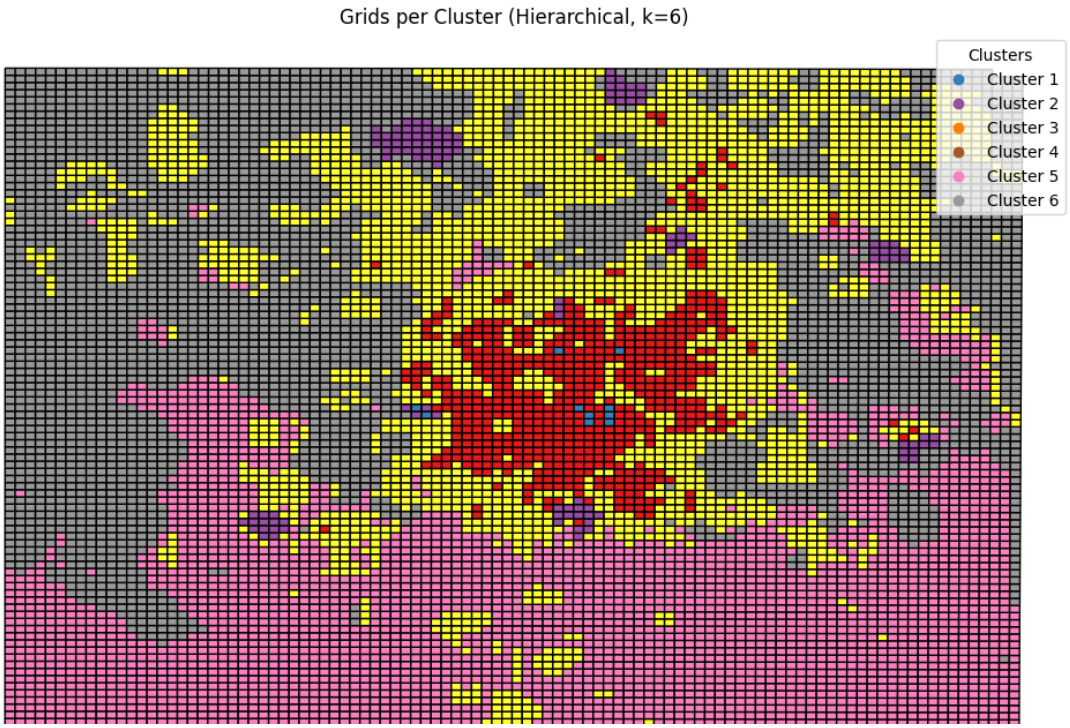
Στη συνέχεια, πραγματοποιείται πλήρης ιεραρχική ανάλυση με απεικόνιση του **δενδρογράμματος (dendrogram)**, που δείχνει τη δομή συγχώνευσης των σημείων, αποκόπτοντας στα πρώτα 20 επίπεδα για ευκρίνεια. Το dendrogram επιτρέπει οπτική αναγνώριση πιθανών φυσικών διαχωρισμών στα δεδομένα. Αμέσως μετά, εφαρμόζεται η συνάρτηση fcluster για να αποδώσει ετικέτες συστάδων (labels) με βάση την

επιλεγμένη τιμή k = 6, με χρήση του maxclust criterion, που κόβει το δενδρόγραμμα σε k ομάδες.



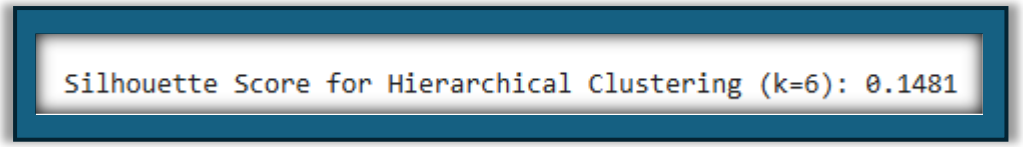
Εικόνα 24: Απεικόνιση δενδροδιαγράμματος για k=6

Παρατηρούνται τρεις βασικές σκάλες (διαφορετικά χρώματα) που υποδεικνύουν πιθανούς διαχωρισμούς σε 3 κύρια clusters, καθώς οι μεγαλύτερες συγχωνεύσεις συμβαίνουν σε μεγάλες αποστάσεις. Η μακρινή απόσταση της μπλε γραμμής δείχνει ότι η τελική ένωση ήταν ανάμεσα σε ήδη διαφορετικές, καλά διαχωρισμένες ομάδες, υποστηρίζοντας την ύπαρξη ξεκάθαρων συστάδων.



Εικόνα 25: Γεωγραφική αναπαράσταση των clusters με την μέθοδο Hierarchical

‘



Η τιμή **Silhouette Score = 0.1481** για την ιεραρχική ομαδοποίηση με **k = 6**, όταν εφαρμόζεται στο **internet activity dataset**, υποδηλώνει ότι οι ομάδες που σχηματίστηκαν **δεν έχουν σαφή πυκνότητα** ή δεν είναι καλά διαχωρισμένες. Τα δεδομένα δραστηριότητας στο διαδίκτυο (όπως διάρκεια κλήσεων, SMS, ή χωρικά χαρακτηριστικά) συχνά εμφανίζουν **ανομοιόμορφη κατανομή**, με περιοχές υψηλής πυκνότητας (π.χ. αστικά κέντρα) και άλλες αραιές (π.χ. αγροτικές περιοχές). Η ιεραρχική μέθοδος, ειδικά με Ward linkage, προσπαθεί να ελαχιστοποιήσει τη συνολική διακύμανση, αλλά δεν λαμβάνει υπόψη τοπικές πυκνότητες. Αυτό σημαίνει ότι μπορεί να ενώσει σημεία με παρόμοια μέση τιμή, αλλά διαφορετική πυκνότητα, δημιουργώντας **ασαφή όρια**. Έτσι, το χαμηλό score δείχνει ότι τα clusters **δεν αντανακλούν τις φυσικές συστάδες πυκνότητας** του dataset. Πιο κατάλληλοι μπορεί να είναι αλγόριθμοι όπως DBSCAN, που βασίζονται απευθείας στην εντοπισμένη πυκνότητα, αναγνωρίζοντας outliers και αυθαίρετα σχήματα.

Ερώτημα 8ο – Χαρακτηρισμός Clusters

Ο χαρακτηρισμός των clusters που προέκυψαν από τις μεθόδους ομαδοποίησης **K-Means, DBSCAN και Hierarchical Clustering** αποκαλύπτει σημαντικές **χωροχρονικές διαφορές στη συμπεριφορά χρήσης τηλεπικοινωνιακών υπηρεσιών** στα διάφορα grids της περιοχής μελέτης. Αντλώντας πληροφορίες από τα **στατιστικά χαρακτηριστικά κάθε cluster**, όπως η **μέση δραστηριότητα ανά ώρα**, οι **διαφορές μεταξύ καθημερινών και Σαββατοκύριακων**, η **σχέση ημερήσιας/νυχτερινής χρήσης** και η **κυρίαρχη μορφή επικοινωνίας** (SMS, κλήσεις, internet), μπορούμε να εξάγουμε λειτουργικές ερμηνείες για τον χαρακτήρα των περιοχών που αντιπροσωπεύει κάθε ομάδα.

Ένα βασικό cluster παρουσιάζει **υψηλή δραστηριότητα κατά τις ώρες ημέρας (08:00–21:59)**, κυρίως σε **κλήσεις** και χρήση **internet**, με έμφαση στις **εργάσιμες ημέρες**. Τα σημεία που ανήκουν σε αυτή την ομάδα εντοπίζονται συχνά **κοντά στο κέντρο της πόλης**, όπως δείχνει η μικρή απόστασή τους από το milan_center, και εμφανίζουν σχετικά υψηλό δείκτη compactness, υποδηλώνοντας πιο δομημένες αστικές περιοχές. Αυτά τα χαρακτηριστικά συνάδουν με περιοχές **εμπορικής ή επαγγελματικής φύσης**, όπου η δραστηριότητα είναι συγκεντρωμένη κατά τις εργάσιμες ώρες και μειώνεται σημαντικά τη νύχτα και τα Σαββατοκύριακα. Εδώ μπορεί να ανήκουν κεντρικά γραφεία, καταστήματα, υπηρεσίες και γενικά περιοχές υψηλής προσπελασιμότητας, όπου συγκεντρώνεται πληθυσμός για εργασία ή εξυπηρέτηση.

Ένα δεύτερο cluster διαφοροποιείται εμφανώς, παρουσιάζοντας **εντονότερη δραστηριότητα τις βραδινές ώρες (00:00–07:59)**, ειδικά τα Σαββατοκύριακα, και αυξημένη χρήση internet και SMS σε σχέση με τις φωνητικές κλήσεις. Τα σημεία του cluster αυτού εντοπίζονται κυρίως **σε περιοχές απομακρυσμένες από το κέντρο**, με μέτρια έως μεγάλη χωρική επιφάνεια. Η κατανομή της δραστηριότητας υποδεικνύει **κατοικίες ή οικιστικά προάστια**, όπου οι χρήστες δραστηριοποιούνται κυρίως εκτός ωραρίου εργασίας. Οι τιμές των δεικτών totalInternetNight, totalSmsDay, καθώς και η αναλογία dailyInternet/dailySmsCall ενισχύουν την υπόθεση ότι πρόκειται για ζώνες όπου κυριαρχεί η ψυχαγωγική και κοινωνική χρήση του τηλεπικοινωνιακού δικτύου.

Ένα τρίτο cluster περιλαμβάνει περιοχές με **πολύ χαμηλή ή σποραδική χρήση**, χωρίς σαφή χρονικά μοτίβα. Αυτές οι περιοχές παρουσιάζουν μικρές ή μηδενικές τιμές σε όλους τους δείκτες και δεν διακρίνονται από έντονη δραστηριότητα ούτε την ημέρα ούτε τη νύχτα. Συχνά βρίσκονται

στην περιφέρεια του αστικού ιστού και εμφανίζουν αυξημένο αριθμό κορυφών (vertices), δηλαδή πιο περίπλοκες πολυγωνικές γεωμετρίες, που ενδεχομένως υποδηλώνουν αγροτικές περιοχές, φυσικές εκτάσεις ή μη κατοικημένες ζώνες. Ερμηνεύονται λοιπόν ως **περιοχές χωρίς σταθερή πληθυσμιακή δραστηριότητα**, πιθανώς εγκαταστάσεις, αποθήκες, ή εκτάσεις εκτός αστικής κάλυψης.

Τέλος, σε περιπτώσεις clustering με DBSCAN παρατηρείται και η ύπαρξη θορύβου (cluster = -1), δηλαδή περιοχών που δεν ανήκουν σε κανένα cluster. Οι περιοχές αυτές χαρακτηρίζονται είτε από **εξαιρετικά υψηλές τιμές**, είτε από **μηδενική δραστηριότητα**, και μπορούν να ερμηνευτούν ως **τεχνικά outliers ή εξειδικευμένες περιοχές**, όπως αεροδρόμια, σταθμοί ή ενδεχομένως περιοχές όπου η μέτρηση παρουσίασε σφάλματα.

Συνολικά, η ανάλυση και ερμηνεία των clusters αναδεικνύει την **χωρική και χρονική πολυπλοκότητα της ανθρώπινης δραστηριότητας**. Η ενσωμάτωση χρονικών, τηλεπικοινωνιακών και χωρικών χαρακτηριστικών προσφέρει μια πλούσια εικόνα της δυναμικής κάθε περιοχής. Μέσω των clustering αποτελεσμάτων, μπορούμε να κατηγοριοποιήσουμε τα grids σε **λειτουργικούς τύπους περιοχών**, όπως **εμπορικές/επαγγελματικές ζώνες, οικιστικά προάστια, περιθωριακές περιοχές χαμηλής χρήσης**, και εξαιρέσεις. Αυτές οι πληροφορίες είναι κρίσιμες για εφαρμογές σε **αστικό σχεδιασμό, κατανομή υποδομών, διαχείριση δικτύων και έξυπνη πόλη (smart city analytics)**.

Βιβλιογραφία

[1] https://dl.acm.org/doi/pdf/10.1145/331499.331504

[2] https://d1wqtxts1xzle7.cloudfront.net/97762947/IJSC_Paper_4_946-952-libre.pdf?1674612307=&response-content-disposition=inline%3B+filename%3DSupervised_Machine_Learning_Approaches_A.pdf&Expires=1741643450&Signature=EKjPuFp-3MsC3WqrcXvF7NCGZvP2vJ8i5CuQ-e2Nfh4qzxv-bSKLIOT5RZzroZqGwCAaVaxl2BsU8kwryAmRpIhlz39XDHJzhxS~wg66l6EtDFvG4sjm2yGudrD-gLIO6h-KTKGPPkGII5WbvOC7lFB6CA4t5BldoLTcb1-a6ae-f3AOsJgh5ZgZPpWl~LfUfQ-4gLDtqbVulSryxPgM6gcGUbz6~baBwsxGOXKW~vTCI6cIIlSIbK2NJUr-ZfyuuSIttK~C3HvOl1QfHjzqgOOnjCM0UqCqKUs6V84ekozWPkfWYmaVlMNkrsovYhDBl4OuIaiI0H8KCYLNfPAfqg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[3] https://www.researchgate.net/profile/Aqib-Ali-6/publication/368983958_An_Unsupervised_Machine_Learning_Algorithms_Comprehensive_Review/links/643c783f1b8d044c632ba4ab/An-Unsupervised-Machine-Learning-Algorithms-Comprehensive-Review.pdf

[4] https://www.sciencedirect.com/science/article/abs/pii/S1084804520300138

[5]https://openurl.ebsco.com/EPDB%3Agcd%3A10%3A13007738/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A156725591&crl=c&link_origin=scholar.google.com

[6] https://telematics.upatras.gr/wp-content/uploads/2024/01/1570699183.pdf

[7] https://ieeexplore.ieee.org/abstract/document/9376929

[8]https://www.mdpi.com/1424-8220/23/8/3899#:~:text=algorithms%20and%20jamming%20strategies%2C%20were,the%20signal%20to%20interference%20plus

[9]https://pmc.ncbi.nlm.nih.gov/articles/PMC11598088/#:~:text=%2A%20Proposing%20a%20clustering,19

[10]https://www.nature.com/articles/sdata201555

[11]https://github.com/arunasubbiah/milan-telecom-data-modeling/tree/master

[12] https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/QJWLFU

[13] https://www.mdpi.com/2076-3417/12/3/1203

[14] https://www.geeksforgeeks.org/time-series-clustering-techniques-and-applications/

[15] https://arxiv.org/abs/2403.14798

[16] https://encord.com/blog/data-clustering-intro-methods-applications/

[17] https://users.softlab.ntua.gr/~taslan/comp_th/data/DataMining_02.pdf

[18] https://www.researchgate.net/publication/383161365_Temporal_and_Multivariate_Similarity_Clustering_of_5G_Performance_Data