Πανεπιστήμιο Πειραιώς

University of Piraeus

Σχολή Τεχνολογιών Πληροφορικής και Επικοινωνιών.

Τμήμα Πληροφορικής

Εργασία μαθήματος

Γεωγραφικά Πληροφοριακά Συστήματα

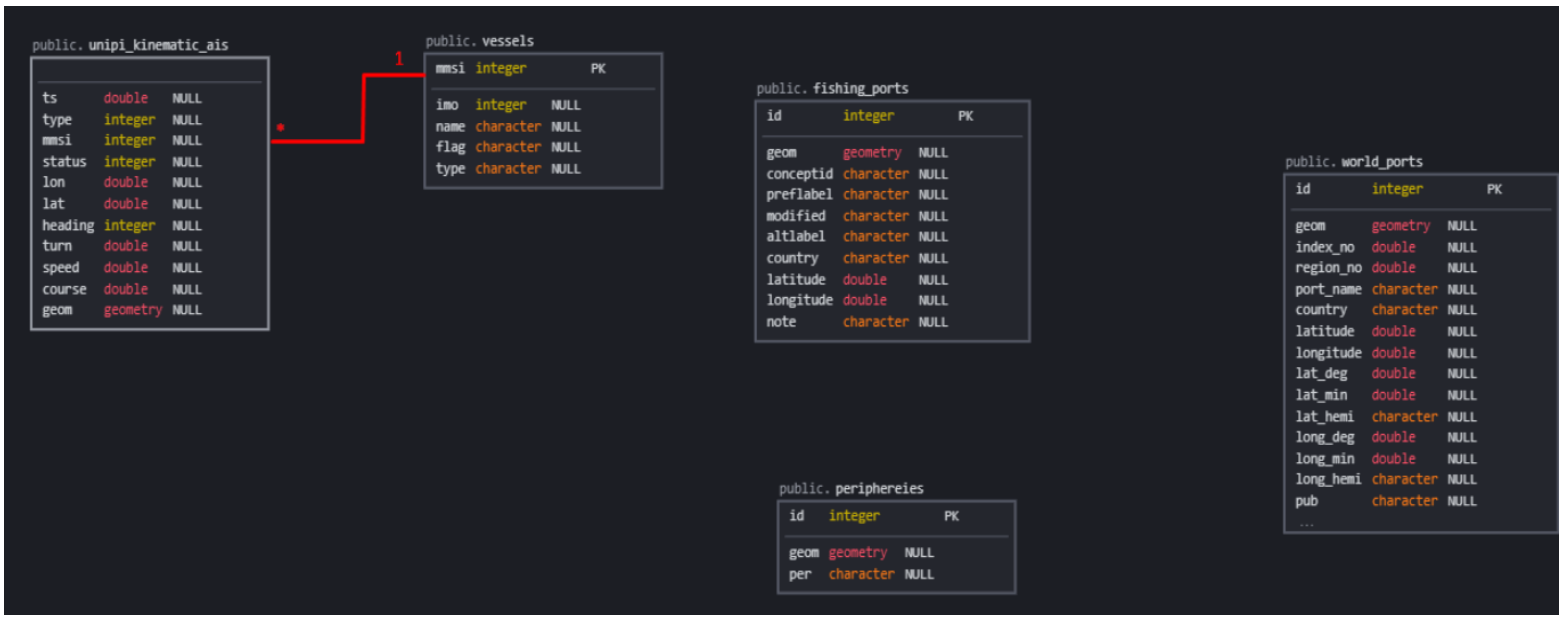
Περιεχόμενα

1. Φόρτωση Δεδομένων

Για την φόρτωση δεδομένων χρησιμοποιήσαμε την PostgreSQL 14 με pgAdmin 4, το QGIS 3.16.12 και την PostGIS. Τα δεδομένα που μας δόθηκαν αφορούσαν τα σήματα της κεραίας του πανεπιστημίου Πειραιά κατά την διάρκεια του Απριλίου 2018 έως και του Μαρτίου 2018. Επίσης στην διάθεση μας έχουμε τα δεδομένα για τα λιμάνια όλου του κόσμου και τις μαρίνες. Για την ανάγκη αφαίρεσης θορύβου από τα δεδομένα μας, χρησιμοποιήσαμε και τα δεδομένα που περιέχουν τις περιφέρειες της Ελλάδος.

Όλος ο κώδικας SQL για αυτό το βήμα περιέχεται στο αρχείο **db.sql**, να σημειωθεί πως για την εισαγωγή των πινάκων που αφορούν τις περιφέρειες και τα λιμάνια, χρησιμοποιήθηκαν τα αντίστοιχα shape files και τα φορτώσαμε στην βάση μας μέσω του QGIS.

Να σημειωθεί επίσης πως για την δημιουργία της στήλης geom που περιέχει τις αντίστοιχες γεωμετρίες, δημιουργήθηκε με την βοήθεια των στηλών lon, lat και με την συνάρτηση που μας προσφέρει η PostGIS ST_SetSRID.



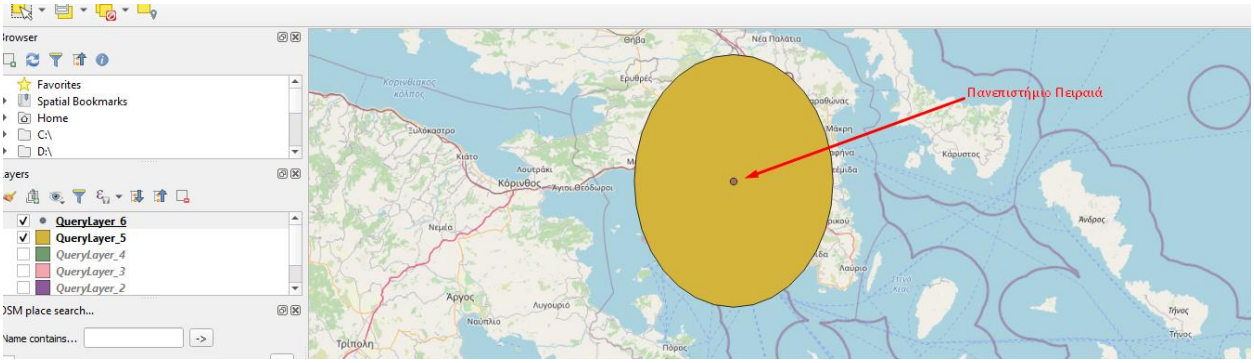
Εικόνα 1 Διάγραμμα βάσης δεδομένων

2. Γνωριμία με τα δεδομένα και προετοιμασία

2.1 Καθαρισμός δεδομένων από θόρυβο

Πριν αρχίσουμε να εξερευνούμε τα δεδομένα μας, θα πρέπει να λάβουμε υπόψη μας πως κάποια σήματα που έχει λάβει η κεραία, είναι λανθασμένα. Αυτό είναι συνηθισμένο να συμβαίνει και για αυτό θα πρέπει να αποβάλουμε κάποιες άκυρες εγγραφές.

Αρχικά θα πρέπει να αφαιρέσουμε τα δεδομένα που βρίσκονται εκτός εμβέλειας της κεραίας. Για να το πετύχουμε αυτό, θέσαμε ένα buffer 37 χιλιομέτρων γύρω από το κτήριο του πανεπιστημίου και όποια εγγραφή βρισκόταν έξω από αυτό το πολύγωνο, το διαγράψαμε.



Εικόνα 2 Εμβέλεια κεραίας

Στη συνέχεια διαγράψαμε όσες εγγραφές ήταν εντός της στεριάς που καλύπτει η κεραία. Για να το καταφέρουμε αυτό, χρησιμοποιήσαμε τον πίνακα με τα δεδομένα των περιφερειών της Ελλάδος. Το συγκεκριμένο dataset, περιέχει τα πολύγονα όλων των περιφερειών, συνεπώς όσα πλοία βρέθηκαν εντός αυτών των πολύγονων, διαγράφηκαν.

Επίσης διαγράψαμε από τους πίνακες των λιμανιών, τα λιμάνια τα οποία βρίσκονται εκτός της εμβέλειας.

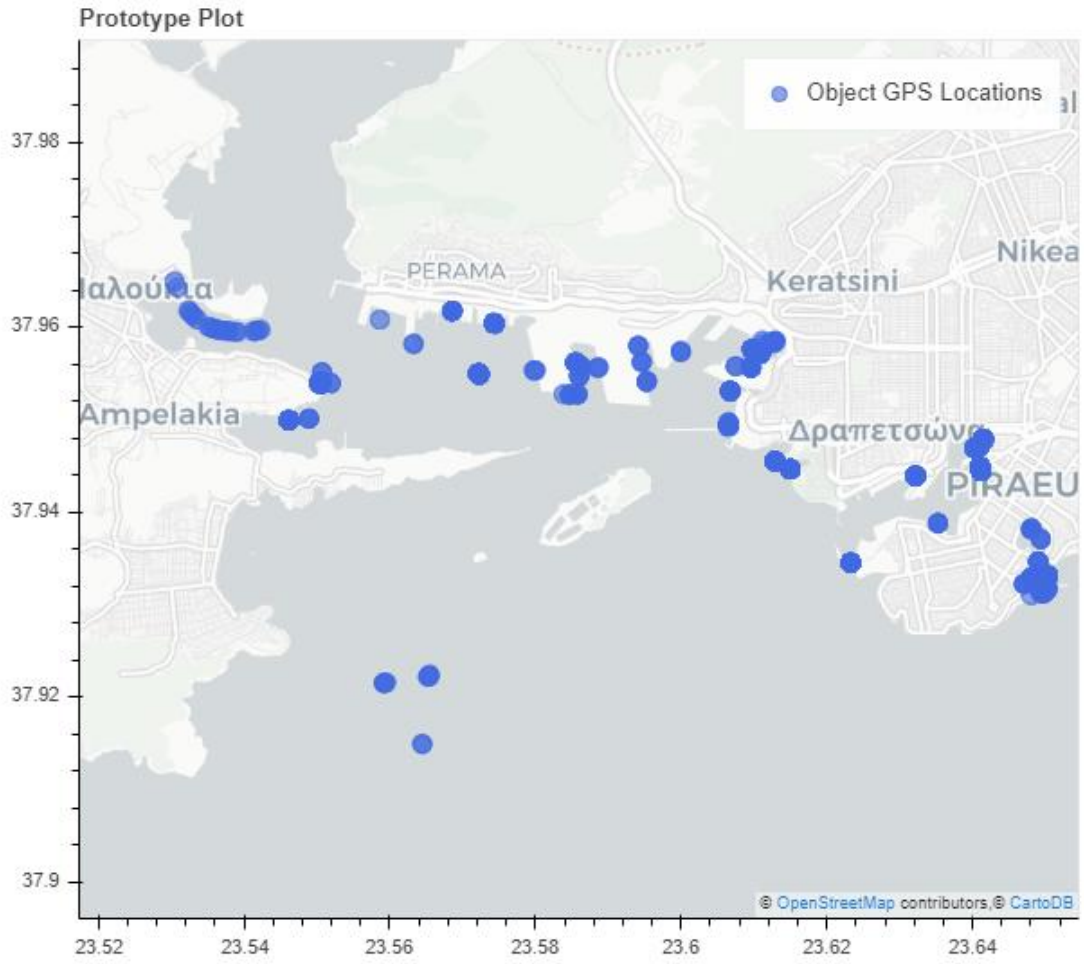
Ο κώδικας για όσα περιγράφονται παραπάνω, υπάρχει στο αρχείο **clean.sql.**

Σημειώνεται πως αυτά αφορούν πρωτοφανή δεδομένα τα οποία έπρεπε να διαγραφούν, σε επόμενο βήμα που θα γνωρίσουμε καλύτερα τα δεδομένα και θα έχουμε στην ευχέρεια μας διάφορα στατιστικά, ίσως χρειαστεί να διαγράψουμε ακραίες ταχύτητες.

2.2 Πληροφορίες για τα δεδομένα και επιπλέον καθαρισμός

Όλος ο κώδικας για το πως προέκυψαν τα παρακάτω στοιχεία, υπάρχουν στο αρχείο **statistics_and_clean.ipynb.** Επίσης να σημειωθεί πως κάναμε export το csv από την βάση δεδομένων και δεν χρησιμοποιήσαμε την βιβλιοθήκη psycopg2 στην συγκεκριμένη περίπτωση, διότι το να διαβάσουμε το csv αρχείο ήταν αρκετά πιο γρήγορη διαδικασία. Επίσης στο csv που κάναμε export, δεν συμπεριλάβαμε την γεωμετρία για λόγους μνήμης.

Αφού φορτώσαμε τα δεδομένα στην Python, αρχικά εμφανίσαμε ένα στίγμα των σημάτων με την βοήθεια του ST-Visions.

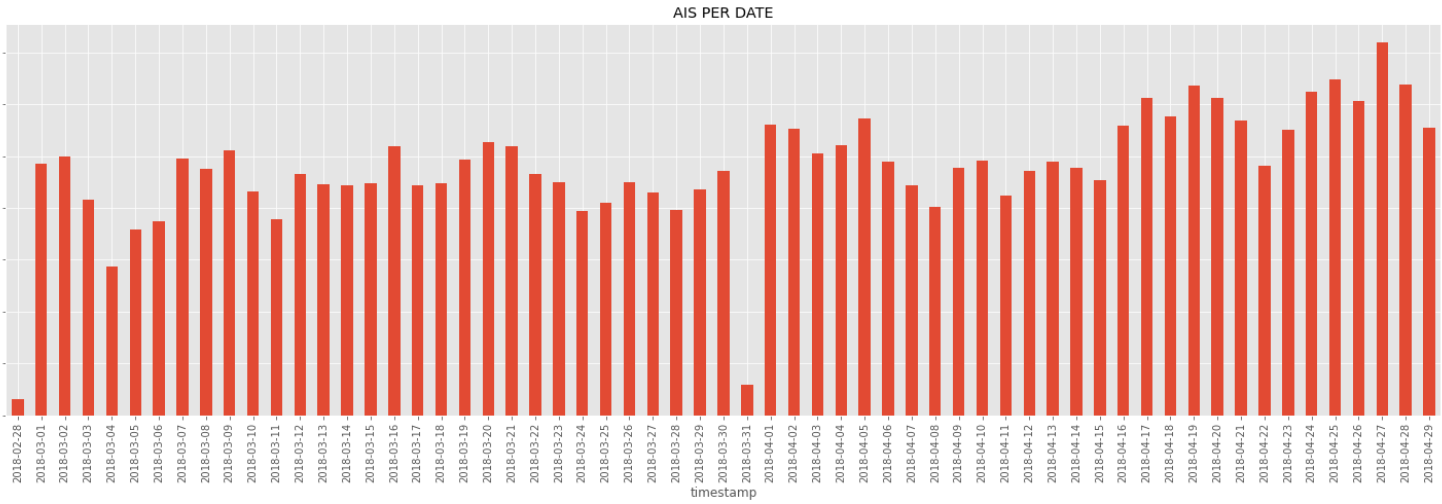


Εικόνα 3 Σήματα κεραίας

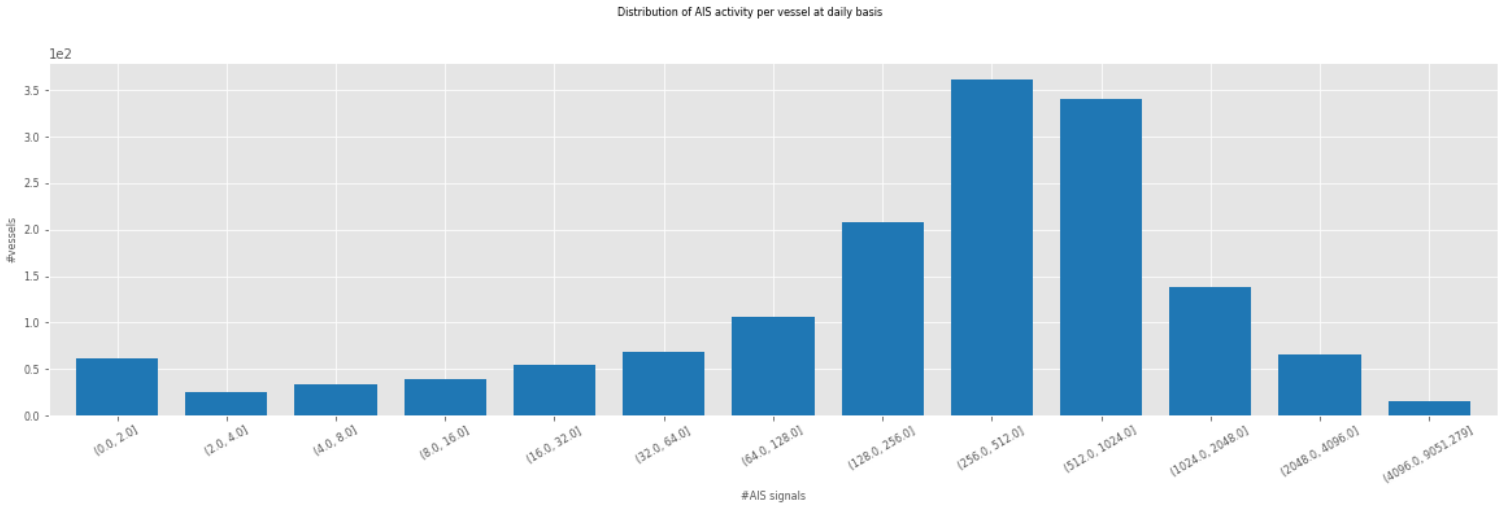
Στη συνέχεια βλέπουμε πως το dataset μας περιλαμβάνει 1520 πλοία, 14587752 καταγραφές και έχουμε τα παρακάτω features:

- mmsi: μοναδικός κωδικός του πλοίου
- lon, lat: συντεταγμένες του gps σε μοίρες
- speed: ταχύτητα πλοίου, ναυτικά μίλια ανά ώρα, 1 ναυτικό μίλι = 1852 μέτρα
- heading: γωνία σε μοίρες, κατεύθυνση πλόρης καραβιού
- course: γωνία σε μοίρες, πορεία του καραβιού

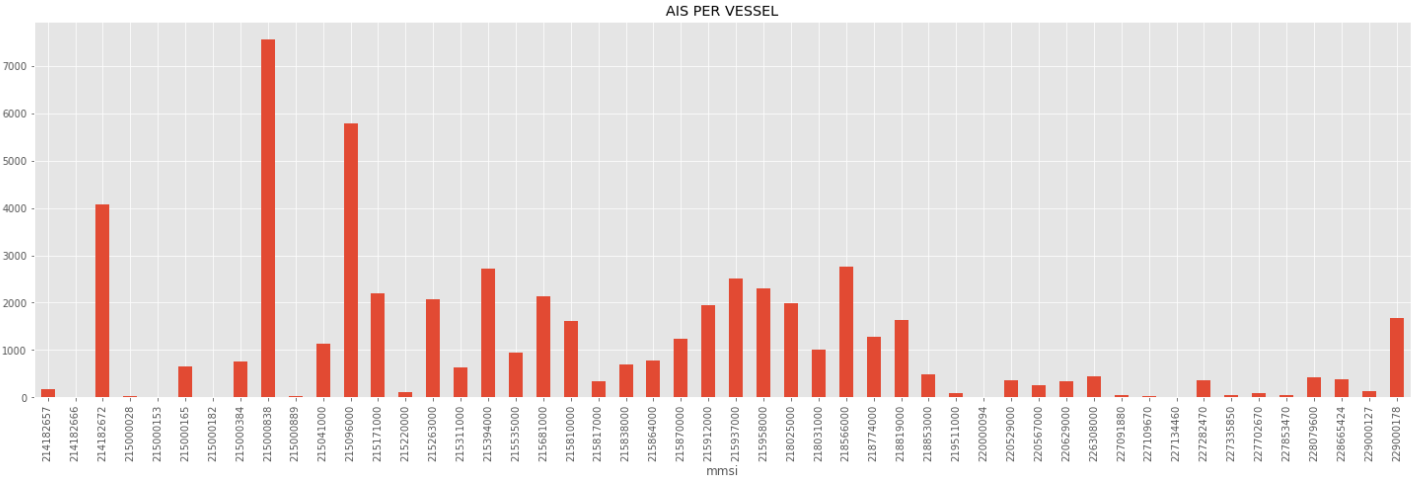
Επίσης παρακάτω ακολουθούν κάποια διαγράμματα για την καλύτερη κατανόηση των δεδομένων μας.



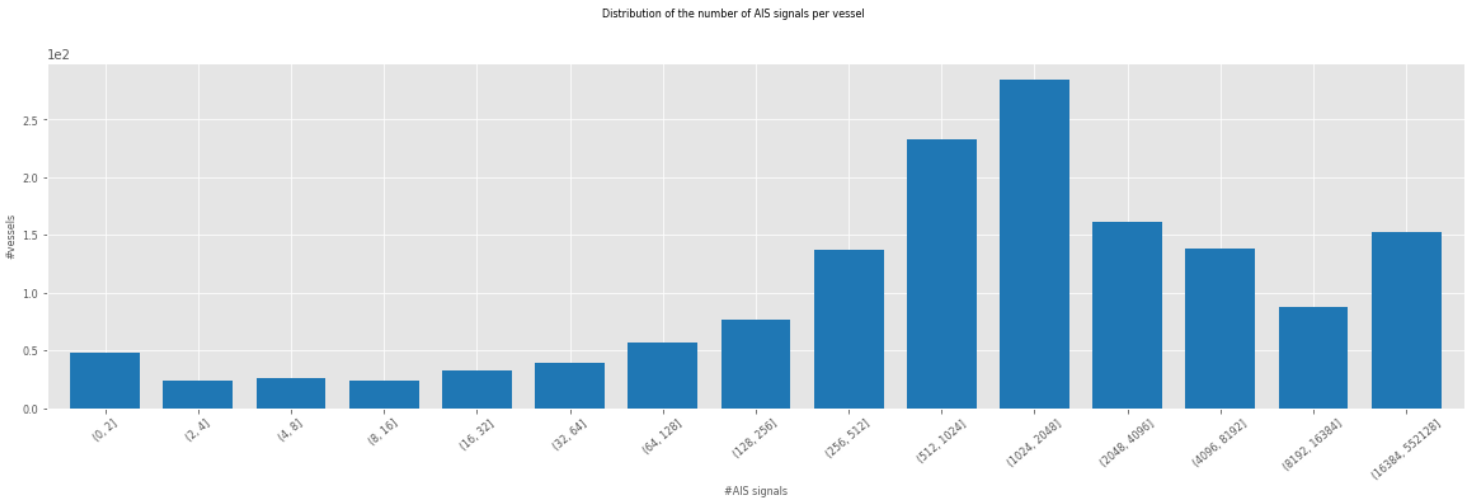
Εικόνα 6 Πλήθος σημάτων για όλες τις ημερομηνίες του dataset



Εικόνα 5 Κατανομή σημάτων ανά vessel σε καθημερινή βάση



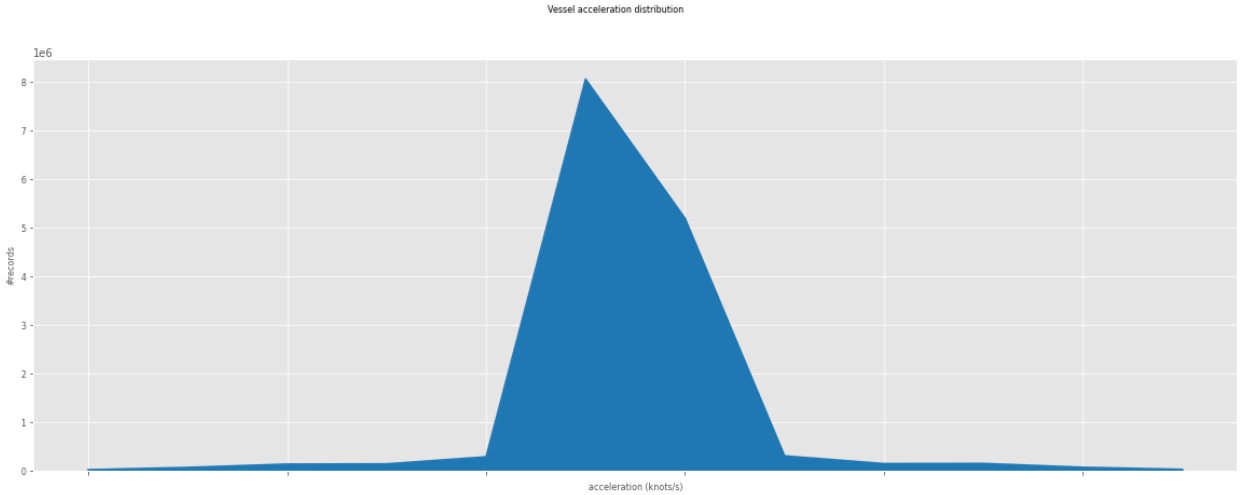
Εικόνα 4 Πλήθος σημάτων από κάποια vessel



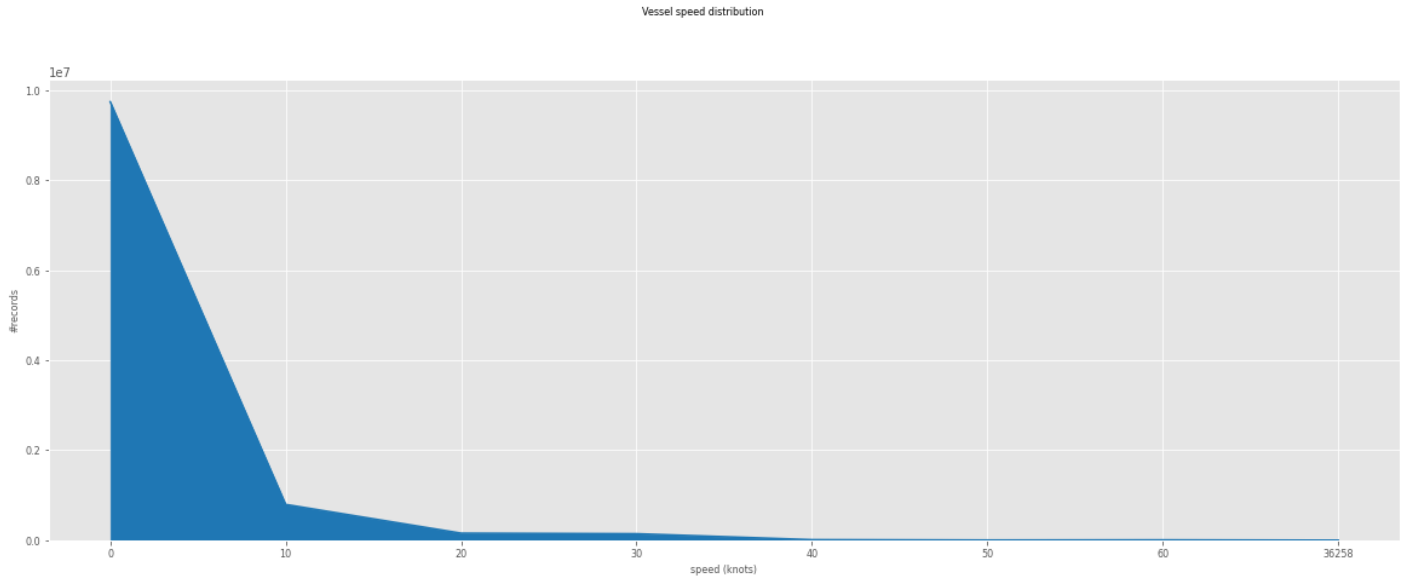
Εικόνα 7 Κατανομή σημάτων των vessel

Στην συνέχεια υπολογίσαμε κάποιες απαραίτητες τιμές για το dataset μας. Αυτές οι τιμές είναι η ταχύτητα ανά vessel (στο dataset μας έχουμε την ταχύτητα που είχε εκείνη την χρονική στιγμή που αναφέρεται), την επιτάχυνση και το bearing. Αυτές τις τιμές τις υπολογίζουμε με την συνάρτηση helper που μας έχει δοθεί από τα εργαστήρια του μαθήματος.

Παρακάτω φαίνονται οι κατανομές των ταχυτήτων και της επιτάχυνσης των vessel.



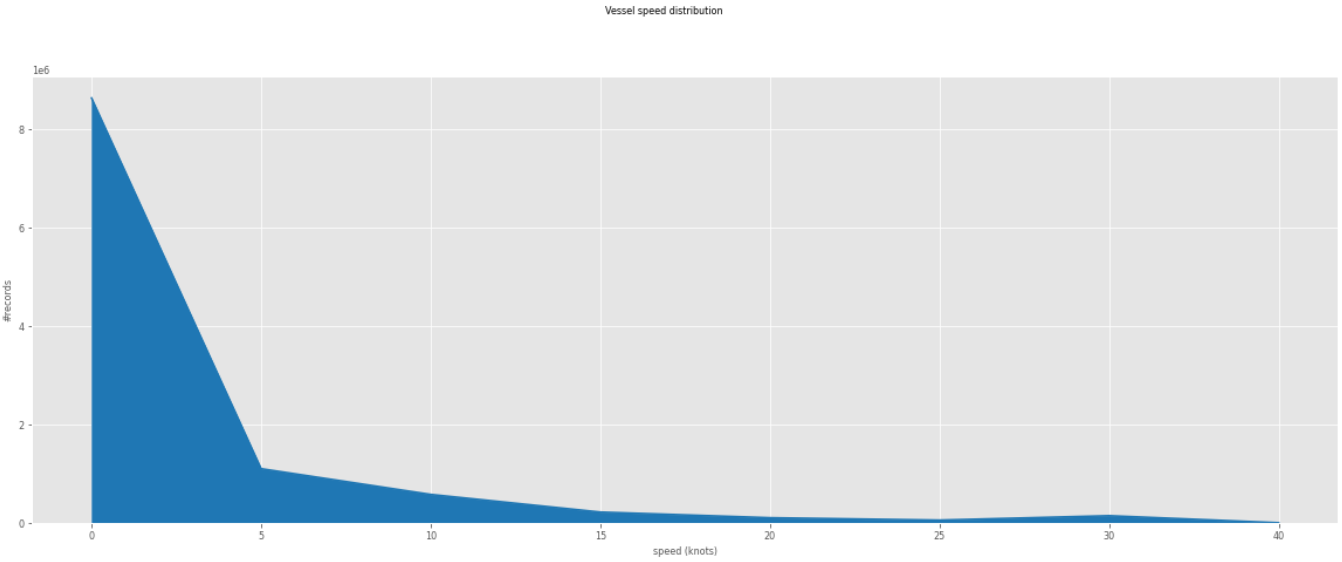
Εικόνα 8 Κατανομή επιτάχυνσης των vessel



Εικόνα 9 Κατανομή ταχύτητας των vessel

Όπως φαίνεται από το παραπάνω διάγραμμα, κάποιες ταχύτητες είναι ακραίες. Για να κρατήσουμε ένα ικανοποιητικό δείγμα και μα κρατήσουμε την ρεαλιστικότητα των δεδομένων, αποφασίσαμε πως θα σβήσουμε τις εγγραφές που αφορούν ταχύτητες μεγαλύτερες από 40 knots.

Αφού σβήσαμε όλες αυτές τις εγγραφές, πλέον έχουμε 1504 πλοία (πριν είχαμε 1520) και 14.546.856 (πριν είχαμε 14.587.752). Το dataset που προκύπτει το αποθηκεύουμε σε ένα csv αρχείο με όνομα ‘unipi_ais_clean.csv’ για να συνεχίσουμε στα επόμενα ερωτήματα.



Εικόνα 10 Κατανομή ταχύτητας στα καθαρά δεδομένα

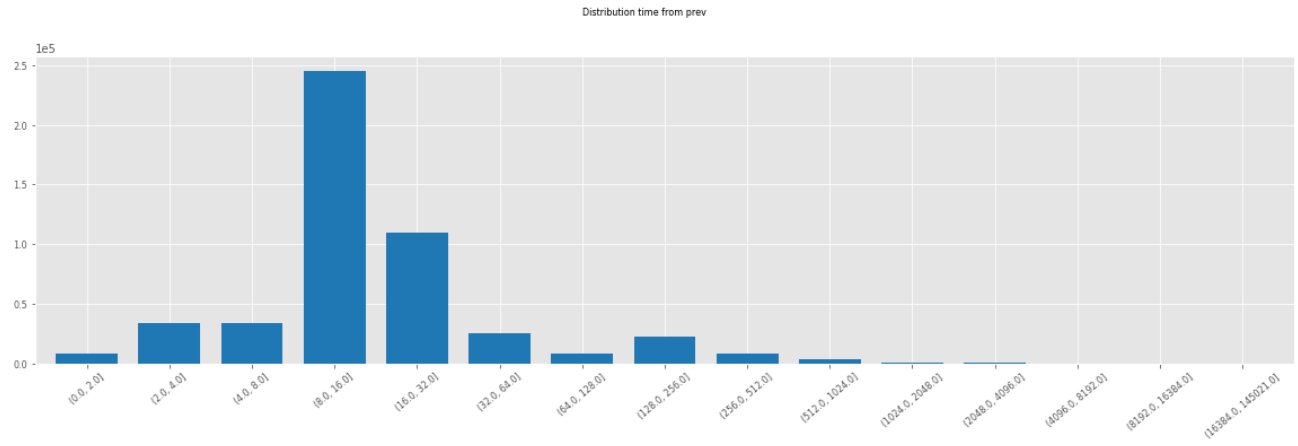
2.3 Δημιουργία Ιστογραμμάτων

Για την δημιουργία ιστογράμματος της διαφοράς των χρονικών στιγμών μεταξύ διαδοχικών σημείων και για την δημιουργία του ιστογράμματος της διαφοράς απόστασης ανά σημείο, δουλέψαμε ως εξής. Αρχικά φορτώσαμε το dataset με τα καθαρά δεδομένα, και στη συνέχεια το κάναμε ταξινόμηση με βάση το mmsi και το timestamp. Με αυτό τον τρόπο καταφέρνουμε να έχουμε όλες τις εγγραφές ενός πλοίου συνεχόμενες στο dataframe και ταξινομημένες με βάση το timestamp.

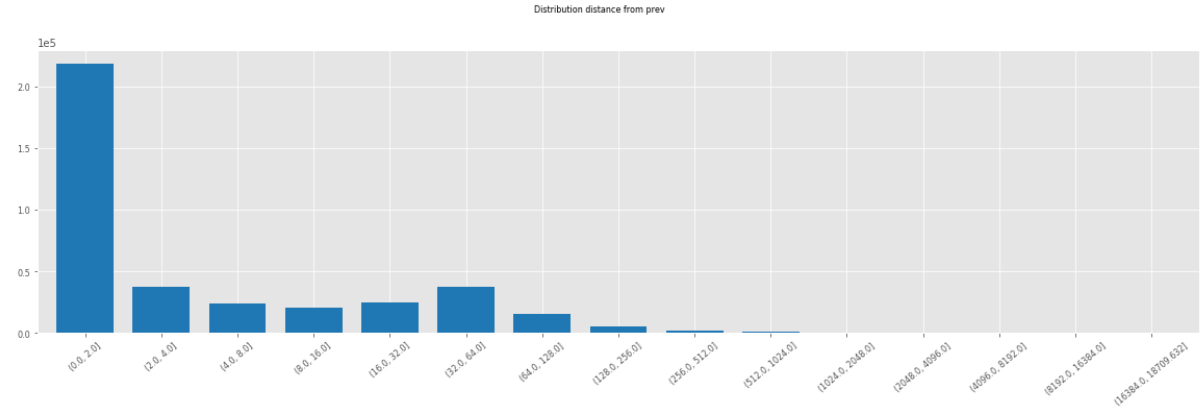
Στην συνέχεια κάναμε ένα loop το οποίο σε κάθε επανάληψη είχαμε την τωρινή γραμμή των data και την αμέσως επόμενη.

Για αυτές τις δύο γραμμές υπολογίζαμε την διαφορά τους σε δευτερόλεπτα και αν η τωρινή γραμμή είχε ίδιο mmsi με την επόμενη γραμμή (δηλαδή είναι το ίδιο πλοίο), κάναμε append σε μια λίστα time_difference την διαφορά των timestamp και σε μια άλλη λίστα points_difference κάναμε append το haversine των δύο σημείων. Ο υπολογισμός για το haversine distance έγινε με την βοήθεια την συνάρτησης haversine που περιέχεται στο αρχείο helper που υπάρχει και στα εργαστήρια.

Στην περίπτωση που το τωρινό mmsi ήταν διαφορετικό από το επόμενο mmsi, απλά κάναμε append στις λίστες τη τιμή μηδέν. Τα ιστογράμματα που προκύπτουν είναι τα παρακάτω.



Εικόνα 11 Ιστόγραμμα διαφοράς χρονικών στιγμών



Εικόνα 12 Ιστόγραμμα διαφοράς απόστασης σημείων

Ο κώδικας για τα παραπάνω υπάρχει στο αρχείο **istogramma.ipynb**

2.4 Δημιουργία Χωρικού Ευρετηρίου

Για την δημιουργία ευρετηρίου χρησιμοποιήσαμε την παρακάτω εντολή, η οποία κάνει spatial index πάνω στον πίνακα με τα δεδομένα της κεραίας.

```
CREATE INDEX spatial_index
ON unipi_kinematic_ais
USING GIST (geom);
```

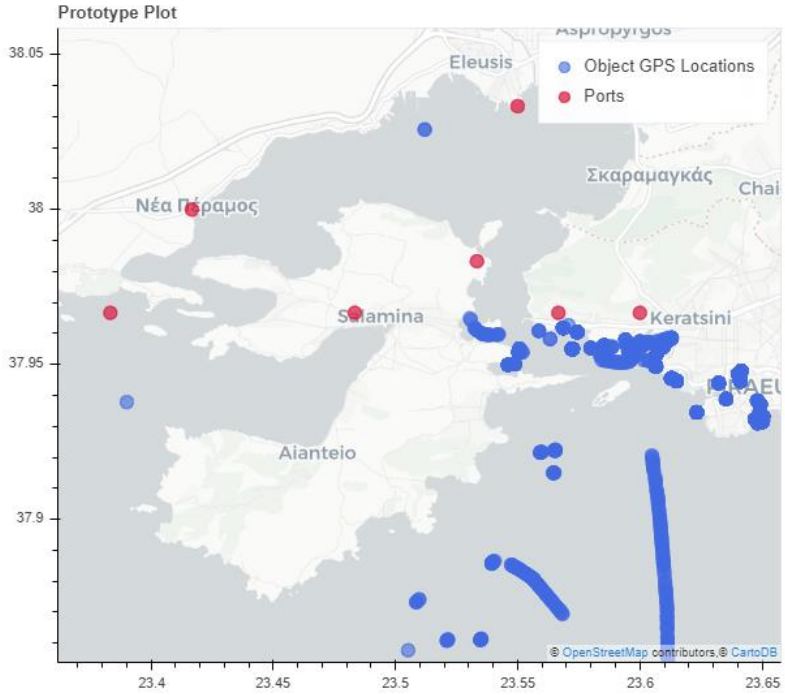
Τα ερωτήματα που χρησιμοποιήσαμε μπορείτε να τα βρείτε στο αρχείο **spatial_indexing.sql**. Επίσης σε επόμενο ερώτημα δημιουργούμε ένα ευρετήριο με την χρήση της Python. Προφανώς αυτό το κάνουμε για λόγους ταχύτητας.

3 Επεξεργασία και αναλυτική των δεδομένων

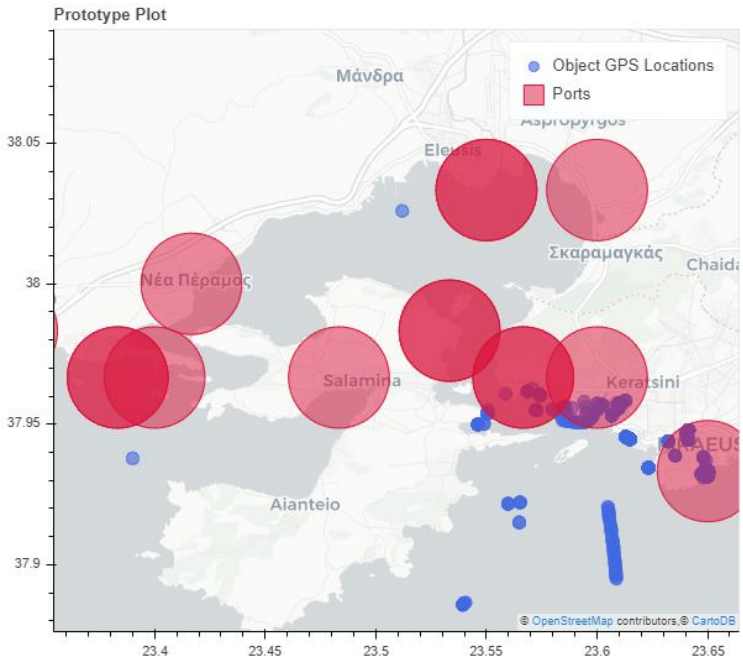
Όλος ο κώδικα για τα παρακάτω υπάρχει στο αρχείο **trajectory.ipynb**

Αρχικά θα πρέπει να αξιοποιήσουμε την πληροφορία που μας δίνεται από τους πίνακες των λιμανιών και των μαρίνων. Θυμίζουμε πως από αυτούς του πίνακες έχουμε κρατήσει μόνο τα δεδομένα εκείνα που η γεωμετρία τους βρίσκονται εντός της εμβέλειας της κεραίας.

Να σημειωθεί πως οι γεωμετρίες που μας δίνονται είναι σημεία, οπότε για κάθε δεδομένο θα πρέπει να ορίσουμε ένα buffer έτσι ώστε να δημιουργηθεί το εύρος του λιμανιού. Αυτό που κάνουμε είναι να ορίζουμε ένα buffer 2 χιλιομέτρων για κάθε σημείο.

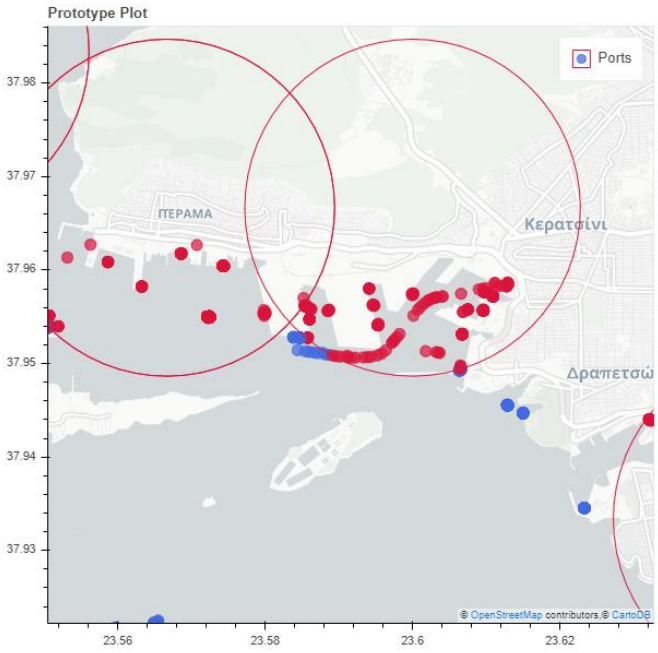


Εικόνα 13 Εικόνα με τα λιμάνια και τα σήματα της κεραίας



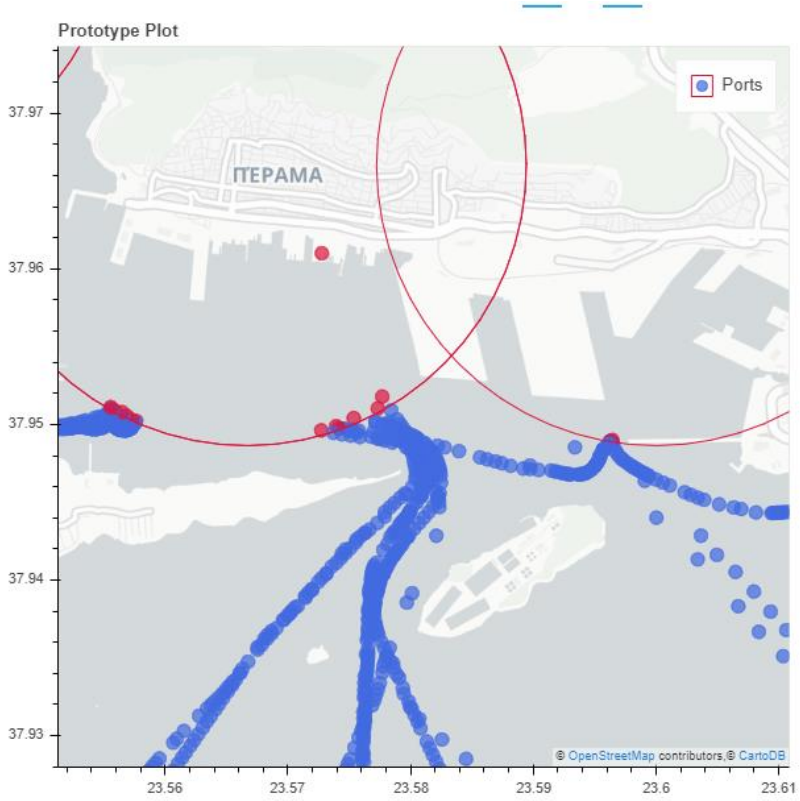
Εικόνα 14 Λιμάνια μετά το buffer

Αφού δημιουργήσαμε το εύρος των λιμανιών, στην συνέχεια δημιουργούμε ένα ευρετήριο με την pandas πάνω στο dataframe με τα ais σήματα και στην συνέχεια βρίσκουμε ποια σημεία είναι εντός κύκλου και ποια σημεία είναι εκτός κύκλου. Για τα σημεία που είναι εντός κύκλου δημιουργούμε μια καινούργια στήλη traj_id με τιμή -1 και για τα υπόλοιπα σημεία η τιμή αυτή θα είναι 0.



Εικόνα 15 Τα κόκκινα δείχνουν τα σημεία που βρίσκονται εντός λιμανιού και τα μπλε δείχνουν τα σημεία που είναι εκτός λιμανιού

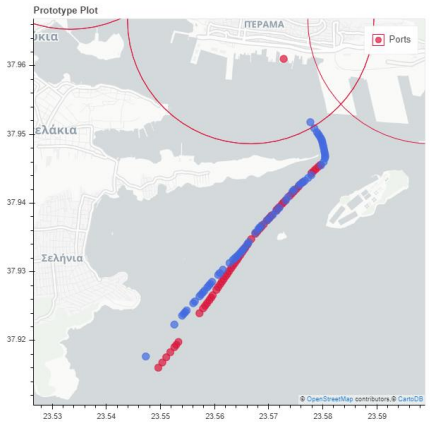
Στην συνέχεια για την δημιουργία τροχιών, χρησιμοποιούμε την έτοιμη συνάρτηση create_trajectories που μας παρέχεται από το σύνολο συναρτήσεων που μας δίνει η helper. Αυτό που κάνει αυτή η συνάρτηση είναι να δέχεται σαν όρισμα ένα dataframe ταξινομημένο με βάση τον χρόνο και επιστρέφει τις τροχιές των πλοίων με βάση τα traj_id που δημιουργήσαμε. Στην ουσία αυτό που κάνει είναι κρατάει το τελευταίο -1 και όλα τα υπόλοιπα τα κάνει 0.



Εικόνα 16 Τροχιές πλοίων

3.1 Χωρική τροχιά ενός πλοίου

Τώρα με το fix_trajectories που υπάρχει στην helper θα κάνουμε split στο -1 και χωρίζει τα μηδενικά, και κάθε ομάδα μηδενικών κρατάει το δικό του id. Έτσι η τροχιά για το πλοίο με mmsi *1193046* έχει την παρακάτω τροχιά.

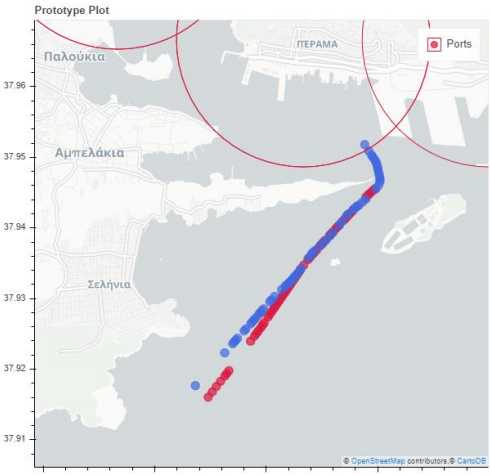


Εικόνα 17 Τροχιά πλοίου 1193046

Στην εικόνα φαίνεται στην κόκκινη γραμμή που σχηματίζεται πως το πλοίο μπαίνει στο λιμάνι και το μπλε είναι η γραμμή που βγαίνει.

3.2 Χρονική τροχιά ενός πλοίου

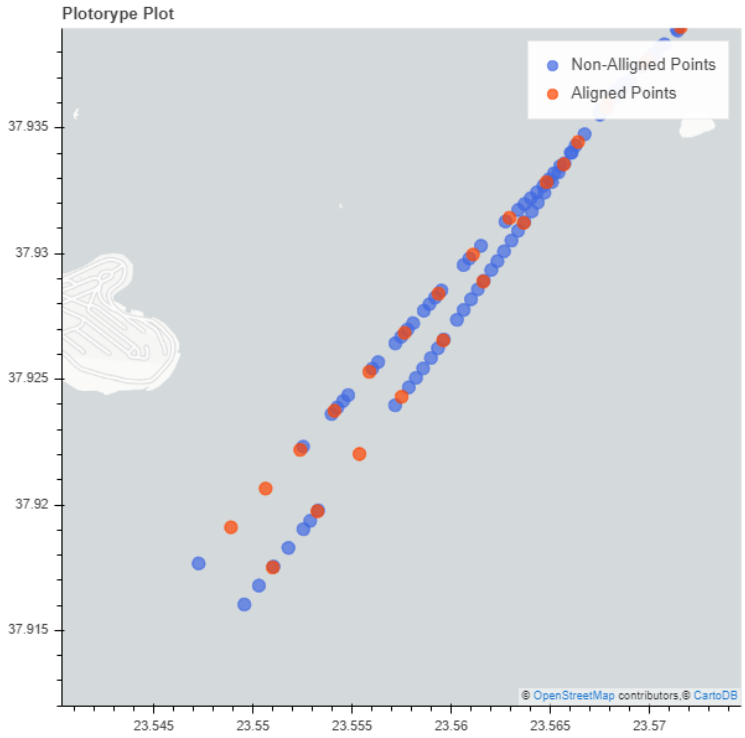
Προηγουμένως φάνηκε η πορεία του πλοίου με βάση το πότε βγαίνει και πότε μπαίνει μέσα στο λιμάνι. Όμως η πορεία αυτή μπορεί να διήρκησε πολλές ώρες. Για αυτό την παραπάνω τροχιά θα την χωρίσουμε με βάση τον χρόνο. Αυτό το καταφέρνουμε με την συνάρτηση temporal_ segmentation της helper η οποία χωρίζει την τροχιά ανά 12 ώρες. Όπως φαίνεται και από την παρακάτω εικόνα, η τροχιά όντως είναι ενιαία.



Εικόνα 18 Τροχιά πλοίου χωρισμένη με βάση τον χρόνο

3.3 Αναδειγματοληψία και χρονικός συγχρονισμός

Για να πετύχουμε τον χρονικό συγχρονισμό, χρησιμοποιούμε την συνάρτηση temporal_alignment από το σύνολο συναρτήσεων της helper, η οποία δημιουργεί για κάθε τροχιά ένα συγκεκριμένο εύρος από ημερομηνίες. Παρακάτω φαίνεται ο συγχρονισμός της τροχιάς για το πλοίο 1193046.



Εικόνα 19 Συγχρονισμένα και μη συγχρονισμένα σημεία του πλοίου 1193046

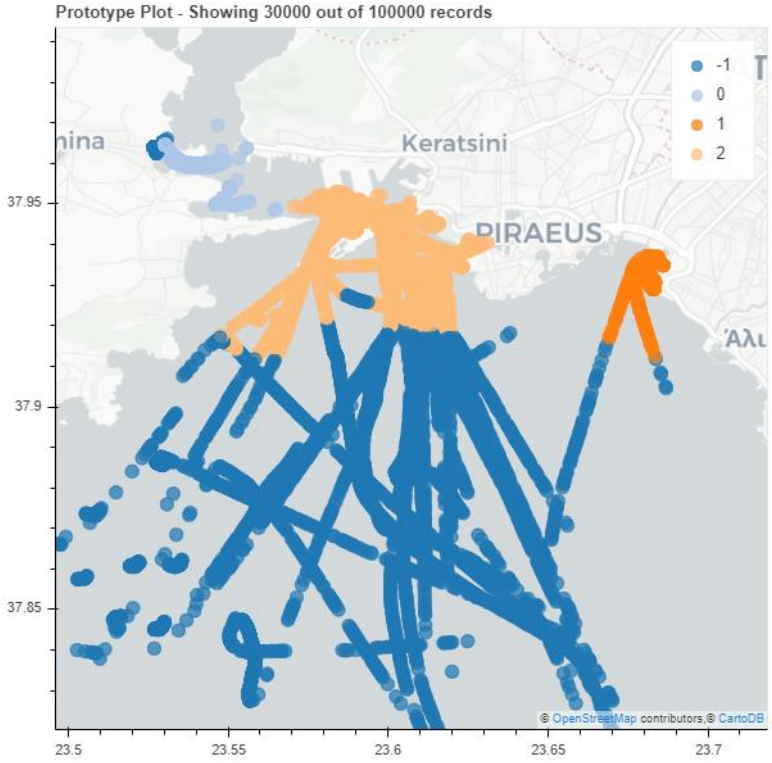
Στη συνέχεια αποθηκεύουμε τα συγχρονισμένο dataset σε ένα csv αρχείο για να το χρησιμοποιήσουμε σε επόμενο ερώτημα.

3.4 Εύρεση πυκνών περιοχών

Για την εύρεση πυκνών περιοχών χρησιμοποιήσαμε τους αλγόριθμο DBSCAN και OPTICS που μας προσφέρει η βιβλιοθήκη της SKLEARN. Ο κώδικας για τα ερωτήματα 3.4 και 3.5 βρίσκονται στο αρχείο **clustering.ipynb**. Να σημειωθεί πως για την εκτέλεση των αλγορίθμων χρησιμοποιήσαμε 100.000 εγγραφές από το συνολικό dataset διότι όταν φορτώναμε περισσότερα δεδομένα ο υπολογιστής κράσαρε για τα καλά και αναγκαζόμασταν να κάνουμε επανεκκινήσεις.

Αρχικά να αναφέρουμε πως τα hot spot είναι σημεία που παρατηρείται πολύ κίνηση. Οπότε αυτό που επιδιώκουμε με τους παραπάνω αλγορίθμους είναι να βρούμε clusters οι οποίοι θα μας δώσουν κάποια επιπλέον πληροφορία για τα δεδομένα μας.

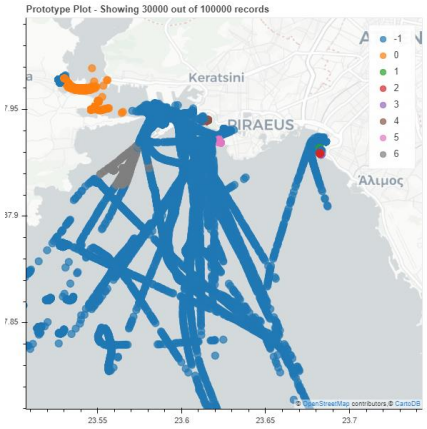
Παρακάτω φαίνονται τα αποτελέσματα του αλγορίθμου DBSCAN



Εικόνα 20 Clusters που ανακάλυψε ο DBSCAN

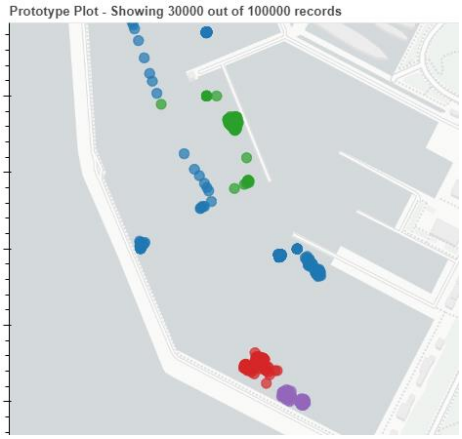
Ο DBSCAN βρήκε 3 clusters. Τα σημεία που είναι με βαθύ μπλε είναι ο θόρυβος και τα υπόλοιπα χρώματα υποδεικνύουν τους clusters. Στην ουσία αυτό που ανακάλυψε ο DBSCAN είναι τρία λιμάνια. Μάλιστα τα λιμάνια που βρίσκονται στο Κερατσίνι και στον Πειραιά τα αναγνωρίζει σαν ένα. Επίσης με ανοιχτό μπλε φαίνεται το λιμάνι της Σαλαμίνας.

Το γεγονός ότι αυτά τα δύο λιμάνια τα δείχνει σαν ένα μπορεί να σημαίνει πως συχνά υπάρχει έντονος συνωστισμός ανάμεσα στα καράβια των δύο λιμανιών, οπότε θα μπορούσαμε να πούμε πως θα χρειάζεται να γίνεται κάποια διαδικασία έτσι ώστε να μην υπάρξουν τυχόν ατυχήματα ή αναταράξεις λόγω κίνησης. Επίσης φαίνεται πως και πιο έξω από τα λιμάνια υπάρχουν νοητές (με ανοικτό κίτρινο) γραμμές που δηλώνουν τυχόν hot path. Τώρα ας δούμε τα αποτελέσματα από τον αλγόριθμο OPTICS.



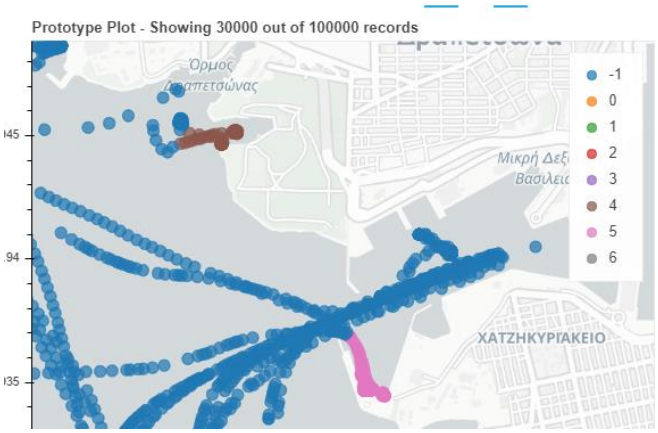
Εικόνα 21 Αποτελέσματα αλγορίθμου OPTICS

Ο αλγόριθμος OPTICS έχει βρει 6 clusters, όπου ο πορτοκαλί cluster αποτελεί το λιμάνι της Σαλαμίνας. Επίσης έχει βρει κάποια σημεία όπου πιθανόν να είναι κάποια ακίνητα καράβια.



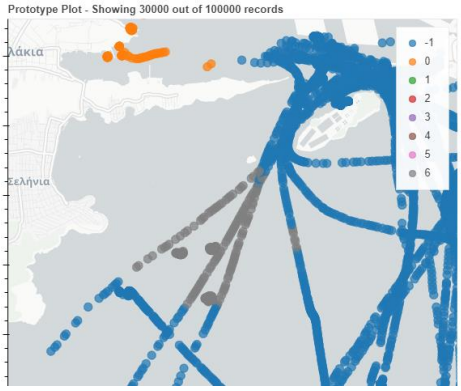
Εικόνα 22 Πιθανά ακίνητα καράβια

Επίσης έχει ανακαλύψει την παρακάτω κίνηση που είναι είσοδος για τα πλοία.



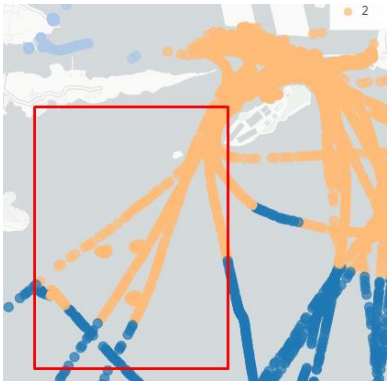
Εικόνα 23 Clusters που δηλώνουν πιθανή είσοδο για τα πλοία

Και τέλος έχουμε κάποια hot paths



Εικόνα 24 Hot paths

Το παραπάνω hot path στον προηγούμενο αλγόριθμο φαινόταν σαν ένας ενιαίος cluster μαζί με τα λιμάνια



Εικόνα 25 CLUSTER από τον DBSCAN

Επομένως το συγκεκριμένο σημείο είναι πιθανό να χρειάζεται κάποια περαιτέρω εξέταση από το λιμεναρχείο έτσι ώστε να λάβει πιθανά μέτρα έτσι ώστε να μην υπάρξει κάποια ανεπιθύμητη κίνηση ή τυχόν ατύχημα λόγω έντονου συνωστισμού.

3.5 Εύρεση ομάδων πλοίων

Για την εύρεση ομάδων πλοίων χρησιμοποιήσαμε τον αλγόριθμο EvolvingClusters, θέτοντας το distance_threshold ίσο με 3704, το time_threshold ίσο με 10 και το min_cardinality ίσο με 3. Ωστόσο ο αλγόριθμος δεν βρήκε κάποια ομάδα πλοίων. Αυτό ίσως συμβαίνει διότι χρησιμοποιούμε ένα μικρό μέρος του dataset, ή διότι έχουμε κάνει κάποιο λάθος που δεν έχουμε αντιληφθεί ή όντως δεν υπάρχει κάποια ομάδα πλοίων που τηρεί τις προϋποθέσεις που θέσαμε. Ωστόσο ο κώδικας υπάρχει στο αρχείο **clustering.ipynb**

Βιβλιογραφία

- Σημειώσεις εργαστηριακών και θεωρητικών διαλέξεων του μαθήματος.
- https://github.com/DataStories-UniPi/ST-Visions
- https://github.com/DataStories-UniPi/EvolvingClusters