**ΤΕΧΝΙΚΕΣ ΕΞΟΡΥΞΗΣ ΔΕΔΟΜΕΝΩΝ**

**ΑΣΚΗΣΗ 2**

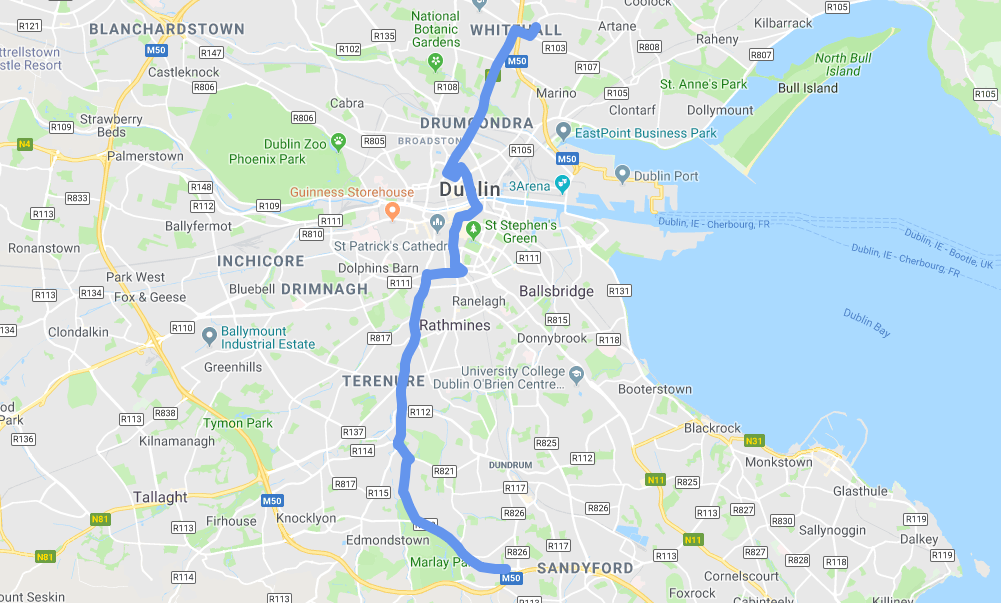
**ΣΤΑΜΑΤΟΠΟΥΛΟΣ ΒΑΣΙΛΕΙΟΣ (1115201400188)**

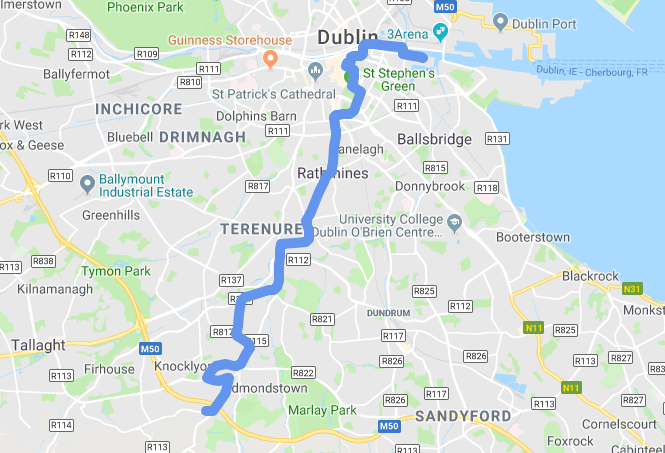
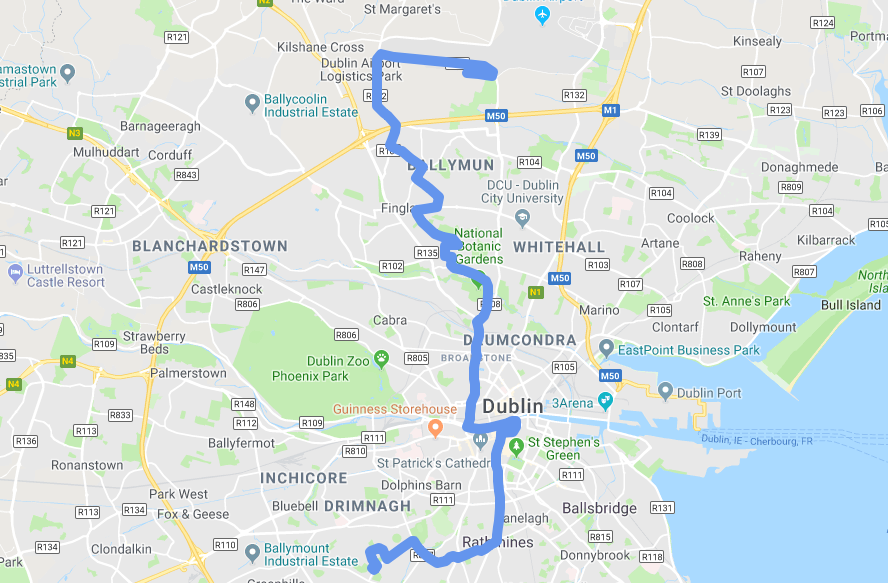
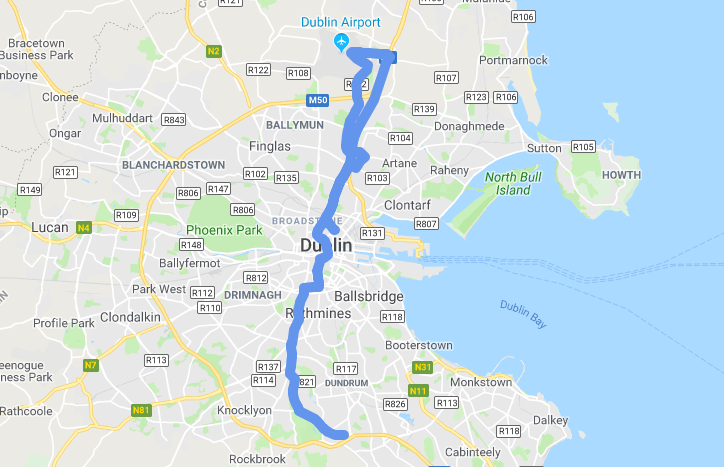
**–**

**ΚΟΥΚΑΚΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ (1115201400289)**

**Ερώτημα 1**

Το ερώτημα αυτό απαντήθηκε στο αρχείο mapping.py με τα αποτελέσματα του να βρίσκονται στο φάκελο maps. Πήραμε 5 τυχαία ids από το dataset και βρήκαμε την πρώτη εμφάνιση του καθενός στο σύνολο και χαρτογραφήσαμε την καθεμιά από αυτές. Έχουμε 5 δυάδες λιστών lats και lons, και αναλόγως τις γεμίζουμε με τα δεδομένα των συντεταγμένων.  
Τέλος, χαρτογραφούμε τις διαδρομές με τη χρήση του **gmplot** και παίρνουμε τα παρακάτω αποτελέσματα.



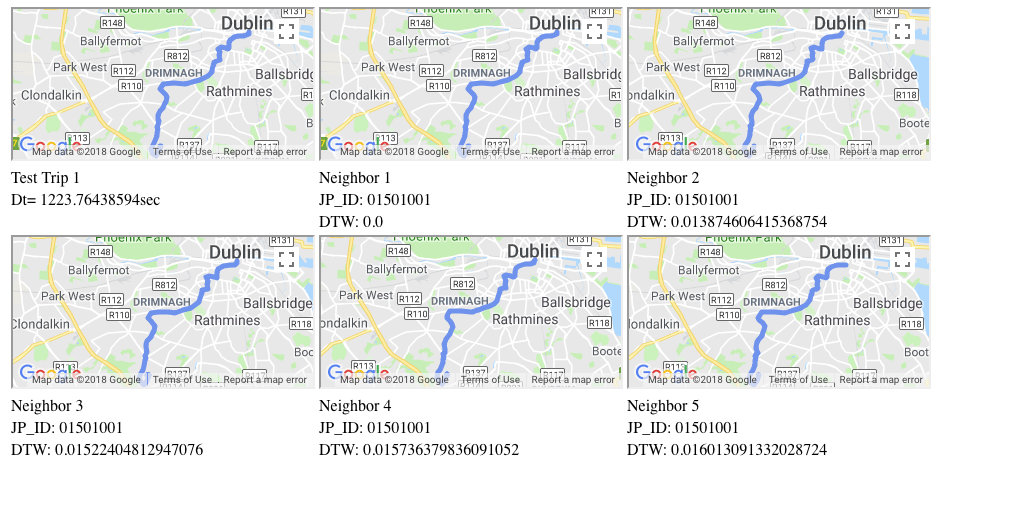


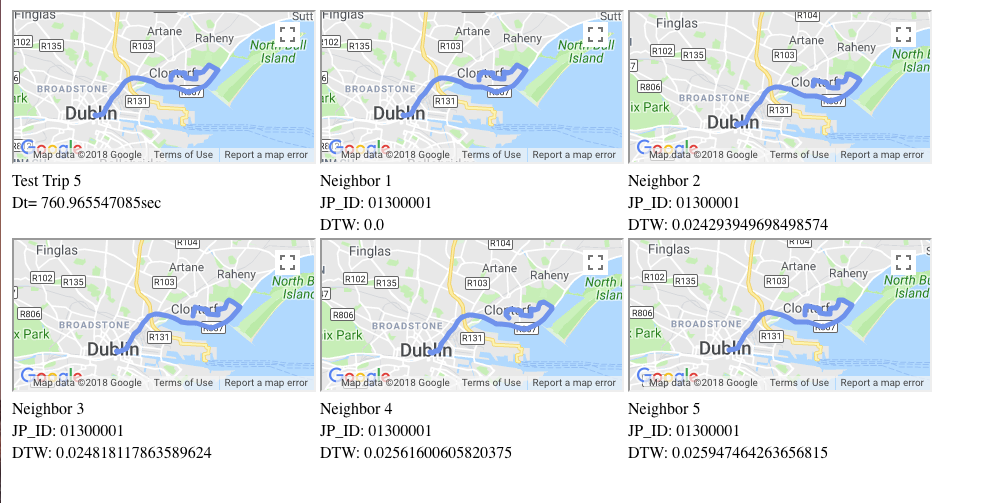
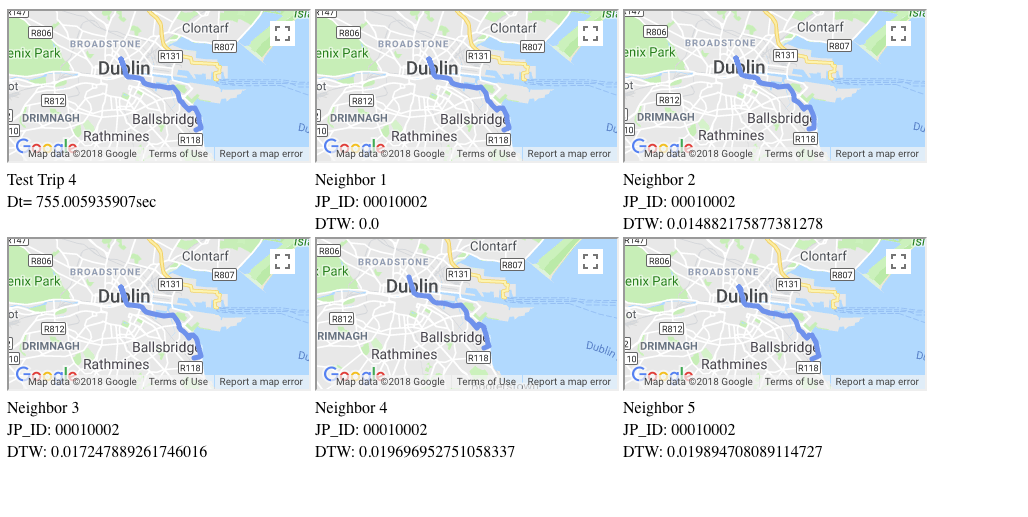
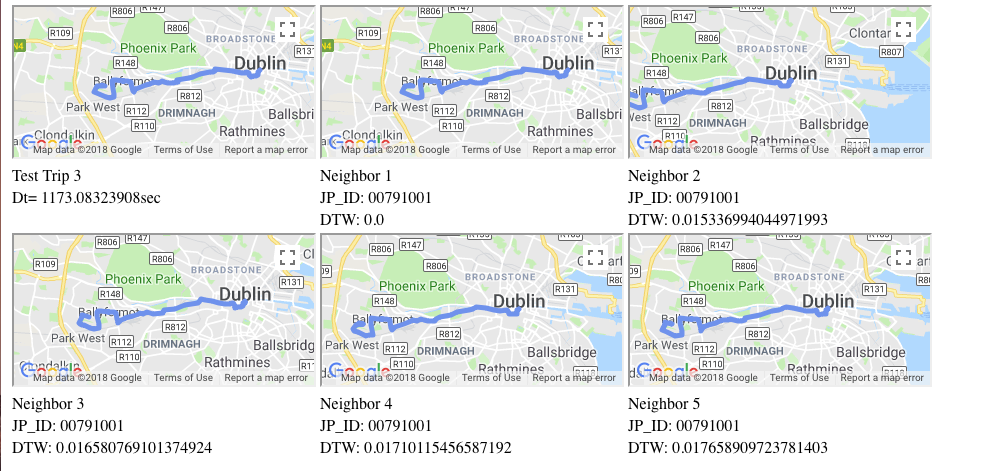
**Ερώτημα 2**

**A.**

Για το ερώτημα αυτό ο κώδικας βρίσκεται στο αρχείο dtw\_nn.py. Εκεί, υπάρχει μια συνάρτηση η οποία υπολογίζει την απόοσταση haversine μεταξύ δύο σημείων συντεταγμένων. Για κάθε trajectory στο σύνολο ελέγχου, παίρνουμε τις συντεταγμένες και τις αποθηκεύουμε σε δύο λίστες. Έπειτα, δημιουργούμε ένα .html αρχείο το οποίο αναπαριστά τη διαδρομή και θα το χρησιμοποιήσουμε αργότερα στα αποτελέσματα. Υπολογίζουμε την dtw της διαδρομής με κάθε διαδρομή στο σύνολο εκπαίδευσης, και για αυτή τη διαδικασία χρησιμοποιούμε τον αλγόριθμο της βιβλιοθήκης dtw.

Στη συνέχεια, τοποθετούμε όλες τις αποστάσεις σε μια λίστα από tuples {διάσταση, index}, ταξινομούμε την λίστα βάση της απόστασης και βρίσκουμε τις 5 μικρότερες. Τέλος, αφού χαρτογραφήσουμε τις γειτονικές αποστάσεις, με τη βοήθεια του iframe, φτιάχνουμε έναν πίνακα html για να δώσουμε τα αποτελέσματα, τα οποία δίνονται στη συνέχεια.





**B.**

Για το ερώτημα αυτό ο κώδικας βρίσκεται στο αρχείο lcss.py. Η διαδικασία μοιάζει αρκετά με αυτή του προηγούμενου ερωτήματος μόνο που εδώ αντί για μικρότερες αποστάσεις, ψάχνουμε τα περισσότερα κοινά σημεία. Τα σημεία αυτά τα βρίσκουμε με τον αλγόριθμο lcs (longest common subsequence), ο οποίος επιστρέφει δύο πράγματα. Τον αριθμό των ταιριαστών σημείων και το μεγαλύτερο κοινό μονοπάτι. Δύο σημεία ταιριάζουν όταν απέχουν λιγότερο από 200m. Για το κοινό μονοπάτι χρησιμοποιούμε την συνάρτηση common\_path οι οποία πέρνει των πίνακα που δημιουργεί η lcs και το test path και ελέγχει τα ταιριαστά διαγώνια σημεία και αναλόγως μειώνει το αντίστοιχο index (I,j). Στην ουσία κάνει traverse των πίνακα όπως θα τον έκανε και η lcs. Τέλος, αφού έχουμε το κοινό μονοπάτι και τα κοινά σημεία τα τοποθετούμε σε μια λίστα και δημιουργούμε ένα tuple του τύπου: matching = [ [#no of points, [common\_path]], index].

Σε αυτή τη δομή βρίσκονται όλα τα στοιχεία που χρειαζόμαστε για να καταγράψουμε τα αποτελέσματα μας.

Τα αποτελέσματα δίνονται παρακάτω:

