

ΤΕΧΝΙΚΕΣ ΕΞΟΡΥΞΗΣ ΔΕΔΟΜΕΝΩΝ

2^η ΑΣΚΗΣΗ

ΠΑΝΑΓΙΩΤΗΣ-ΑΡΙΣΤΟΜΕΝΗΣ ΚΑΤΩΠΟΔΗΣ : 1115201400068
ΣΤΥΛΙΑΝΟΣ ΣΥΡΤΑΓΓΙΑΣ : 1115201400196

Η εργασία αποτελείται από το αρχείο :

ask_maps.py

και υλοποιεί όλα τα ζητούμενα ερωτήματα της εκφώνησης της άσκησης

Από την εκτέλεση του προγράμματος δημιουργούνται τα αρχεία:

- Για το Ερώτημα 1:

map0.html map1.html map2.html map3.html map4.html

ένα για κάθε μια από τις 5 πρώτες διαδρομές του train_set.csv.

Τα αποτελέσματα έχουν συγκεντρωθεί στο αρχείο **Q1.pdf**

- Για το Ερώτημα 2-A1:

Q2mapTest0.html Q2mapTest1.html Q2mapTest2.html
Q2mapTest3.html Q2mapTest4.html

Q2map0.html	Q2map1.html	Q2map2.html
Q2map3.html	Q2map4.html	
Q2map5.html	Q2map6.html	Q2map7.html
Q2map8.html	Q2map9.html	
Q2map10.html	Q2map11.html	Q2map12.html
Q2map13.html	Q2map14.html	
Q2map15.html	Q2map16.html	Q2map17.html
Q2map18.html	Q2map19.html	
Q2map20.html	Q2map21.html	Q2map22.html
Q2map23.html	Q2map24.html	

Οπου για κάθε διαδρομή από το αρχείο test_set_a1.csv έχει δημιουργηθεί ένα αρχείο με αυτή τη διαδρομή και 5 γείτονες που εντοπιστηκαν και τα αποτελέσματα αποθηκεύονται στα παρακάτω pdf:

Q2_A1_i.pdf	Q2_A1_ii.pdf	Q2_A1_iii.pdf
Q2_A1_iv.pdf	Q2_A1_v.pdf	

- Για το Ερωτημα 2-A2:

Q2A2mapTest0.html Q2A2mapTest1.html Q2A2mapTest2.html
Q2A2mapTest3.html Q2A2mapTest4.html

Q2map0.html	Q2map1.html	Q2map2.html
Q2map3.html	Q2map4.html	
Q2map5.html	Q2map6.html	Q2map7.html
Q2map8.html	Q2map9.html	
Q2map10.html	Q2map11.html	Q2map12.html
Q2map13.html	Q2map14.html	
Q2map15.html	Q2map16.html	Q2map17.html
Q2map18.html	Q2map19.html	
Q2map20.html	Q2map21.html	Q2map22.html
Q2map23.html	Q2map24.html	

Οπου για κάθε διαδρομη από το αρχείο test_set_a2.csv έχει δημιουργηθει ένα αρχείο με αυτή τη διαδρομη και 5 γειτονες που εντοπιστηκαν και τα αποτελεσματα αποθηκευονται στα παρακατω pdf:

q2_a2_i.pdf	q2_a2_ii.pdf	q2_a2_iii.pdf
q2_a2_iv.pdf	q2_a2_v.pdf	

- Για το Ερωτημα 3:

Δημιουργειται το απαιτητο αρχείο .csv της εκφωνησης

Οι χρονoi εκτελεσεις για το ερωτημα 2-A1 ειναι:

- 77 secs
- 75 secs
- 132 secs
- 113 secs
- 106 secs

για κάθε διαδρομη του αρχείου αντιστοιχα.

Οι χρονoi εκτελεσεις για το ερωτημα 2-A2 ειναι:

- 223 secs
- 250 secs
- 120 secs
- 174 secs
- 215 secs

για κάθε διαδρομη του αρχείου αντιστοιχα.

ΣΧΟΛΙΑΣΜΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

- Για το Ερωτημα 1:

Παιρνουμε τις 5 πρωτες διαδρομες από το αρχείο και για κάθε μια αποθηκευουμε σε μια λιστα τα lon και σε μια άλλη τα lat και μεσω της gmplot δημιουργουμε τα 5 αρχεια χαρτη.

- Για το Ερωτημα 2-A1:

Παιρνουμε για κάθε μια από τις διαδρομες του αρχείου test_a1.csv και αποθηκευουμε σε μια λιστα τα lon και σε μια άλλη τα lat της κάθε διαδρομης για να δημιουργησουμε μετεπειτα το αρχείο χαρτη για την καθε μια. Την ιδια πρακτικη χρησιμοποιουμε και για ολες τις διαδρομες του αρχείου train_set.csv και για κάθε μια απ αυτές αποθηκευουμε σε μια λιστα tuples που το κάθε ένα περιεχει την αποσταση dtw και τον αριθμο της γραμμης της διαδρομης μεσα στο αρχείο. Στη συνεχεια ταξινομουμε την παραπανω λιστα ως προς το 1^ο πεδιο και παιρνουμε τα 5 πρωτα(δλδ τα ελαχιστα), και παντα με την προαναφερθεισα τεχνικη δημιουργουμε τα αρχεια χαρτη.

- Για το Ερωτημα 2-A2:

Εχουμε τις συναρτησεις my_LCS, my_backTrack που δημιουργει τον πινακα υπολογισμου του LCS και το βελτιστο μονοπατι αντιστοιχα.

Παιρνουμε για κάθε μια από τις διαδρομες του αρχείου test_a2.csv και κανουμε την εξης διαδικασια για κάθε διαδρομη του αρχείου. Δημιουργουμε μια λιστα που περιεχει τις συντεταγμενες των σημειων χωρις το χρονο από τη διαδρομη του αρχείου test_a2.csv και μια λιστα η οποια για κάθε διαδρομη του αρχείου train_set.csv τις συντεταγμενες των σημειων χωρις το χρονο. Τρεχουμε τον LCS για όλα τα ζευγη που προκυπτουν για κάθε διαδρομη του test_a2.csv με του train_set.csv και αν εχουν κοινα σημεια οι διαδρομες αποθηκευουμε σε μια λιστα από tuples το πληθος των κοινων σημειων, τον αριθμο της γραμμης της διαδρομης του train_set.csv και την κοινή διαδρομη. Ταξινομουμε τη λιστα ως προς το πληθος των κοινων σημειων και παιρνουμε τα 5 τελευταια(μεγιστα). Τελικα με τη γνωστη τεχνικη δημιουργουμε τα αρχεια χαρτη.

- Για το Ερωτημα 3:

Εχουμε υλοποιησει τον KNN χωρις Cross Validation

Εχουν χρησιμοποιηθει οι εξης βιβλιοθηκες

```
import pandas as pd
import time
from ast import literal_eval
from gmplot import gmplot
import numpy as np
from haversine import haversine #project PyPi haversine
from fastdtw import fastdtw
import operator
```