Τεχνικές Εξόρυξης Δεδομένων.
1η Άσκηση
Ομαδική Εργασία (2 Ατόμων)

Στην εργασία αυτή θα ασχοληθείτε με δεδομένα από γνωστή εφαρμογή ενοικίασης κατοικιών. Συγκεκριμένα σας δίνονται τα δεδομένα για την περιοχή της Αθήνας για 3 μήνες του 2019 και του 2023. Τα δεδομένα είναι σε μορφή csv και θα χρησιμοποιήσετε Python για να απαντήσετε στα παρακάτω ερωτήματα.

Α Μέρος

Ερώτημα 1ο: Ανάλυση Δεδομένων (Data exploration)

Τα δεδομένα που σας δίνονται είναι οργανωμένα σε φακέλους (πχ April, March, February). Κάθε φάκελος περιέχει διαφορετικά csv αρχεία τα οποία πρέπει να συνδυάσετε και να συνενώσετε κατάλληλα χρησιμοποιώντας την python και το pandas. Συγκεκριμένα θα χρειαστεί να δημιουργήσετε δύο διαφορετικά csv αρχεία (ένα για το 2019 και ενα για το 2023) τα οποία θα περιέχουν τις παρακάτω στήλες.

id
zipcode
transit
Bedrooms
Beds
Review_scores_rating
Number_of_reviews
Neighbourhood
Name
Latitude
Longitude
Last_review
Instant_bookable
Host_since

Host_response_rate
Host_identity_verified
Host_has_profile_pic
First_review
Description
City
cancellation_policy
Bed_type
Bathrooms
Accommodates
Amenities
Room_type
Property_type
price
Availability_365
Minimum_nights

Σε κάθε ένα από αυτά τα δύο αρχεία (ονομάστε τα train_2019.csv και train_2023.csv) θα χρειαστεί να μελετησετε αν υπάρχουν missing data. Αποφασίστε πως θα τα χειριστείτε και απαλείψτε τα ή συμπληρώστε τα κατάλληλα στα αντίστοιχα αρχεία train. Επιπλέον, θα χρειαστεί να μελετήσει αν υπάρχουν και ακραίες τιμές (για παράδειγμα αν ο αριθμός των δωματίων είναι 7000!) και να χειριστείτε ανάλογα και αυτές τις περιπτώσεις.

Στην συνέχεια καλείστε να απαντήσετε στα παρακάτω ερωτήματα χρησιμοποιώντας γραφήματα, ιστογράμματα, heat maps, κ.α. είτε με το matplotlib είτε με το seaborn ή με οποιαδήποτε άλλη βιβλιοθήκη επιθυμείτε. Η ανάλυση αρχικά να γίνει για κάθε έτος ξεχωριστά εκτός από το **1.9** που μπορεί να γίνει μόνο για μία χρονιά της επιλογής σας.

1.1 Ποιός είναι ο πιο συχνός τύπος room_type για τα δεδομένα σας;
1.2 Φτιάξτε γράφημα ή γραφήματα που δείχνουν την πορεία των τιμών για το διάστημα των 3 μηνών.
1.3 Ποιές είναι οι 5 πρώτες γειτονιές με τις περισσότερες κριτικές;
1.4 Ποιά είναι η γειτονιά με τις περισσότερες καταχωρήσεις ακινήτων;
1.5 Πόσες είναι οι καταχωρήσεις ανά γειτονιά και ανά μήνα;
1.6 Σχεδιάστε το ιστόγραμμα της μεταβλητής neighborhood.
1.7 Ποιος είναι ο πιο συχνός τύπος δωματίου (room_type) σε κάθε γειτονιά (neighborhood);
1.8 Ποιός είναι ο πιο ακριβός τύπος δωματίου;
1.9 Χρησιμοποιήστε τη βιβλιοθήκη Folium Map με τις στήλες latitude/longitude και εμφανίστε σε ένα χάρτη για ένα μήνα της επιλογής σας τα ακίνητα και στα popup στον

χάρτη επιλέξτε όποια άλλη πληροφορία θέλετε να εμφανίζεται για το ακίνητο (πχ bed_type, room_type, transit κτλ).

1.10 Φτιάξτε διαφορετικά wordclouds με τα δεδομένα από τη στήλη neighbourhood, transit, description, last_review.

1.11 Στο ερώτημα αυτό θα αξιοποιήσουμε τη στήλη **amenities.** Αν για τη στήλη αυτή εκτυπώσετε τις μοναδικές τιμές που περιέχει θα διαπιστώσετε ότι θα ήταν καλύτερα να απλοποιηθούν αυτές οι τιμές. Ενδεικτικά θα μπορούσαμε να τις ομαδοποιησουμε στις παρακάτω κατηγορίες: *kitchen, accesibility, Electricity_and_Technology, facilities, kids_friendly, security, services.* Για παράδειγμα όλες οι τιμές Breakfast, Cooking basics, BBQ, grill, oven, Coffee maker, κτλ μπορούν να αντικατασταθούν από τη λέξη *kitchen.* Με την ίδια λογική μπορείτε να σκεφτείτε πως θα αντικαταστήσετε και τις υπόλοιπες τιμές στη στήλη amenities. Το αποτέλεσμα θα είναι η στήλη αυτή να έχει λιγότερες και διακριτές τιμές που θα μπορούσαν αντίστοιχα να χρησιμοποιηθούν για εξαγωγή πληροφορίας (πχ στο ερώτημα 1.13) . Οι κατηγορίες που προτείνουμε είναι ενδεικτικές, μπορείτε να χρησιμοποιήσετε και λιγότερες κατηγορίες (ή και περισσότερες ή ακόμα και διαφορετικές αρκεί να μην χαθεί ο στόχος της απλοποίησης). Στην συνέχεια και αφού έχετε απλοποιήσει αυτήν τη στήλη κάντε ένα ιστόγραμμα το οποίο θα κατανέμονται οι νέες τιμές.

1.12 Υπολογίστε αρχικά την μέση τιμή ανα γειτονιά (θα χρησιμοποιηθεί η στήλη price, ενώ για να μπορούμε να συγκρίνουμε όμοια πράγματα κρατήστε μόνο τα δωμάτια που επιτρέπουν φιλοξενία δύο ατόμων). Δείξτε σε ένα γράφημα τις γειτονιές ταξινομημένες ανάλογα με την μέση τιμή τους. Στην συνέχεια κατατάξτε τις γειτονιές σε 3 ομάδες (πολύ ακριβές, μέτριες, οικονομικές) ανάλογα με την μέση τιμή των δωματίων.

1.13 Ποιά άλλη πληροφορία θα μπορούσατε να εξάγετε από τα δεδομένα που σας δίνονται; Σκεφτείτε **3 επιπλέον** ερωτήσεις για την περιοχή της Αθήνας και εμφανίστε τα αποτελέσματα (μπορείτε και να συνδυάσετε όσες στήλες θέλετε από οποιαδήποτε χρονιά).

1.14 Ο πρόσφατοι νόμοι για τα Airbnb ξεχωρίζουν τους ιδιώτες που αναρτούν προς μίσθωση (ή υπεκμίσθωση) τρία ή περισσότερα ακίνητα σε ψηφιακή πλατφόρμα. Φτιάξτε έναν πίνακα με την πληροφορία (host_id, num_host_listings) που θα έχει τους 10 hosts με τα περισσότερα ακίνητα.

1.15 Δεδομένου ότι μελετήσατε τα παραπάνω για δύο διαφορετικές χρονιές ποιο θα μπορούσε να είναι το δικό σας συμπέρασμα για την περιοχή της Αθήνας όσο αφορά τις βραχύχρονες μισθώσεις; για παράδειγμα, είναι περισσότερες/ διαφορετικές οι γειτονιές ανάμεσα στις δύο χρονιές; αυξήθηκαν οι τιμές; Οι ακριβές γειτονιές παρέμειναν ακριβές; Προσπαθήστε δηλαδή να κάνετε μία σύγκριση ανάμεσα στα δύο έτη με βάση τα γραφήματα που έχετε παρουσιάσει στα παραπάνω ερωτήματα.

Ερώτημα 2ο: Recommendation System

Σε αυτό το ερώτημα θα χρειαστείτε τις στήλες : 'id', 'name', 'description'.

Σκοπός είναι να εξάγετε χρήσιμη πληροφορία από αυτά τα δεδομένα και να προσπαθήσετε να φτιάξετε ένα πρόγραμμα το οποίο θα παράγει προτάσεις (recommendations) για την περιοχή της Αθήνας. Στο πρώτο ερώτημα έχετε ήδη φτιάξει τα wordclouds για τη στήλη description. Σε αυτό το ερώτημα αφαιρέστε τα stop words, πειραματιστείτε με τις παραμέτρους του wordcloud και εντοπίστε τις πιο χαρακτηριστικές λέξεις που χρησιμοποιεί ο επισκέπτης για την περιοχή της Αθήνας. Στη συνέχεια δημιουργήστε μία νέα στήλη που θα έχει την ένωση (concatenation) των στηλών name και description (fill NA with NULL). Απαντήστε στα παρακάτω:

1. Δημιουργήστε τον TF-IDF (Term Frequency - Inverse Document Frequency) πίνακα των unigrams και των bigrams για τη νέα στήλη (χρησιμοποιήστε την παράμετρο stop_word του TfidfVectorizer).
2. Cosine Similarity: Η μετρική αυτή υπολογίζει την ομοιότητα μεταξύ δύο διανυσμάτων x,y, χρησιμοποιώντας τη γωνία μεταξύ τους (όταν η γωνία είναι 0 σημαίνει ότι τα x και y είναι ίσα , αν εξαιρέσουμε το μήκος τους). Διατρέξτε τον TF-IDF πίνακα και υπολογίστε το similarity καθενός ακινήτου με τα υπόλοιπα. Αποθηκεύστε σε ένα python dictionary τα 100 πιο όμοια ακίνητα.
3. Πρόβλεψη : Φτιάξτε μία συνάρτηση η οποία παίρνει σαν είσοδο ένα id και έναν ακέραιο αριθμό N, και επιστρέφει τα N πιο όμοια ακίνητα.

   ```
   recommend(item_id = 4085439, num = 5)
   ```

   Η έξοδος της συνάρτησης να είναι της παρακάτω μορφής μορφής:

   Recommending 5 listings similar to Studio

   Recommended: NAME
   Description: DESCRIPTION
   (score:0.12235188993161432)

   Recommended: NAME
   Description: DESCRIPTION
   (score:0.12235188993161432)
   ……..
4. Λέξεις που εμφανίζονται συχνά μαζί με άλλες λέξεις (collocation). Χρησιμοποιήστε τον BigramCollocationFinder για να βρείτε 10 words που “τείνουν” να εμφανίζονται συχνά μαζί.

Παραδοτέο:

Η εργασία μπορεί να εκπονηθεί **ατομικά ή σε ομάδες 2 ατόμων (προτιμήστε να δουλέψετε σε ομάδα!)** .

Θα ανεβάσετε στο eclass ένα φάκελο της μορφής sdixxxx.zip (όπου sdi το ΑΜ ενός εκ των ατόμων της ομάδας) ο οποίος θα περιέχει μόνο τον κώδικά σας σε μορφή **Ipython notebook** (προσοχή: δεν χρειάζεται να ανεβάσετε και τα αρχεία train.csv). Μπορείτε να έχετε και περισσότερα από ένα notebooks αν σας εξυπηρετεί.

Το notebook πρέπει να έχει “τρέξει” ώστε να φαίνονται τα αποτελέσματα της εργασίας σας. Το notebook αποτελεί και την ολοκληρωμένη αναφορά για την εργασία σας (δεν θα παραδώσετε τίποτα σε doc, pdf) , σχεδιάστε το με προσοχή, να θυμάστε να γράψετε ποιό ερώτημα απαντάται σε κάθε κελί. Διευκρινίσεις για την εργασία θα δίνονται μέσω του e-class.