Data Mining Techniques. 1st Exercise Group Work (2 People)

In this assignment you will deal with data from a well-known housing rental application. Specifically, you are given the data for the Athens area for 3 months of 2019. The data is in csv format and you will use Python to answer the following questions.

Question 1: Data exploration

The data given to you is organized in 3 folders (april, march, february). Each folder contains different csv files which you need to combine and concatenate appropriately using python and pandas. Specifically, you will need to create a single csv file that contains the following columns.

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

In this file (name it train.csv) you will need to study if there are missing data. Decide how to handle them and delete them or fill them appropriately in the train.csv file.

Answer the following questions using graphs, histograms, heat maps, etc. either with matplotlib or with seaborn or any other library you wish.

1.1 What is the most common room_type for your data? **1.2** Make a graph or graphs showing the course of prices over the 3-month period. **1.3** What are the top 5 neighborhoods with the most reviews? **1.4** What is the neighborhood with the most property listings? **1.5** How many listings are there per neighborhood and per month? **1.6** Plot the histogram of the neighborhood variable
1.7 What is the most frequent room_type in each neighborhood?
1.8 What is the most expensive room type? **1.9** Use the Folium Map library with the latitude/longitude columns and display the properties on a map for a month of your choice and in the popups on the map select any other information you want to be displayed for the property (eg bed_type, room_type, transit etc.).
1.10 Make different wordclouds with the data from the neighborhood, transit, description, last_review column. **1.12** What other information could you extract from this data? Think of 2 additional questions for the Athens area and display the results (you can also combine more than 2 columns).

Question 3: Recommendation System

In this query you will need the columns
Id
Name
Description

The purpose is to extract useful information from this data and try to build a program that will produce recommendations for the Athens area. In the first question you have already created the wordclouds for the description column. In this query remove the stop words, experiment with the wordcloud parameters and identify the most characteristic words used by the visitor for the Athens area. Then create a new column that will have the concatenation of the name and description columns (fill NA with NULL). Answer the following:

1. Create the TF-IDF (Term Frequency - Inverse Document Frequency) table of unigrams and bigrams for the new column (use the stop_word parameter of TfidfVectorizer).

2. Cosine Similarity: This metric calculates the similarity between two vectors x,y, using the angle between them (when the angle is 0 it means that x and y are equal TF-IDF matrix and calculate if we exclude their length). Run him through the similarity of each property with the rest. Store in a python dictionary the 100 most similar properties. 3 .Prediction : Make a function that takes as input an id and an integer N and returns the N most similar properties.

,

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

The output of the function should be of the following format

```
Recommending 5 listings similar to Studio
-----------------------------------------

Recommended: NAME
Description: DESCRIPTION
(score:0.12235188993161432)

Recommended: NAME
Description: DESCRIPTION
(score:0.12235188993161432)
……..
```

4. Words that appear frequently together with other words (collocation).
   Use the BigramCollocationFinder to find 10 words that "tend" to appear often together.

Deliverable:

The work can be done **individually or in groups of 2 people.**

You will upload to eclass a folder of the format sdixxxx.zip (where sdi is the ID of one of the group members) which will contain only your code in **Ipython notebook** format (attention: you do not need to upload the train.csv file as well).

The notebook must be "running" in order to see the results of your work. The notebook is also the complete reference for your work (you won't hand in anything in doc, pdf), design it carefully, remember to write a step-by-step description of what your code does in each cell.

Clarifications for the work will be given through the e-class.