**ADDITIONAL MATERIALS**

**DATA**

Data used for this research is open source. It is available on [http://insideairbnb.com](https://insideairbnb.com/get-the-data/).

Steps to use the Data: -

1. Go to this website -> [http://insideairbnb.com](https://insideairbnb.com/get-the-data/)
2. Click on Data dropdown menu and then select Get the Data.

A screenshot of a computer

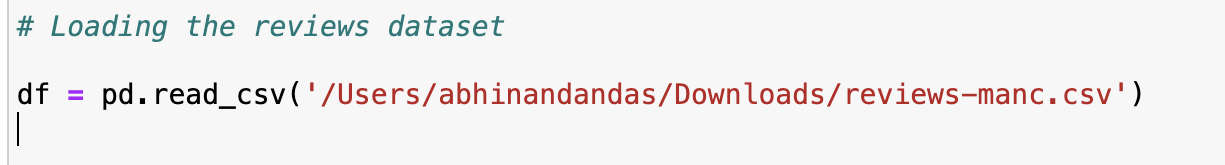
Description automatically generated

1. Choose Greater Manchester from the list of options.

A screenshot of a document

Description automatically generated

1. From the options choose the detailed review data and detailed listings data, namely, reviews.csv.gz and listings.csv.gz.
2. Once downloaded extract the csv files from the zip files and then read on jupyter notebook using pandas read\_csv function.



1. Anonymize the host name and reviewer name column while using to ensure data privacy.

HOST NAME

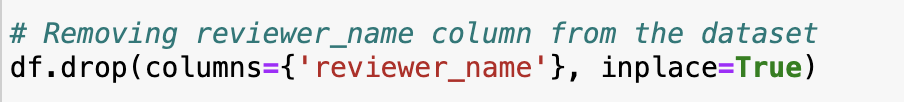
A close-up of a computer screen

Description automatically generated

This function will anonymize the host names to host 1,2,3…

REVIEWER NAME

Drop this column since we do not need that data for our analysis.



1. Once all the above steps are done, the data is ready to use for further analysis.

**GITHUB REPOSITORY**

The code for the entire research has been made publicly available on <https://github.com/Abhinandan305/BM-11349153-ERP>. The GitHub repository consists of all the files needed for running the analysis.

1. Code.py – Full python code for the research
2. Requirements.txt – File containing the list of python libraries and packages needed for this research.
3. Readme.md – A default readme file on GitHub with all necessary steps providing an overview of the entire study.

**PYTHON DOCUMENTATION**

Based on the python (version 3.11.5) code used for this research and for meeting the objectives, the below mentioned packages needs to be installed in the system.

pandas==2.0.3

nltk==3.8.1

langdetect==1.0.9

Unidecode==1.2.0

vaderSentiment==3.3.2

matplotlib==3.7.2

seaborn==0.12.2

numpy==1.24.3

wordcloud==1.9.3

Syntax: -

Run the below command on terminal or Jupyter notebook with individual package names.

! pip install <package name>

**TECHNICAL APPENDIX**

1. **SETUP VIRTUAL ENVIRONMENT –** A virtual environment is recommended before starting any project so that its dependencies do not overlap with other projects creating errors and ambiguity.

Windows

Open Terminal

Step 1: Navigate to your project directory cd path\_to\_your\_project\_directory

Step 2: Create a virtual environment python -m venv env

Step 3: Activate the virtual environment env\Scripts\activate

MacOS/Linux

Open Terminal

Step 1: Navigate to your project directory cd path\_to\_your\_project\_directory

Step 2: Create a virtual environment python3 -m venv env

Step 3: Activate the virtual environment source env/bin/activate

1. **LAUNCH JUPYTER NOTEBOOK**

Launch jupyter notebook or any other python IDE.

1. **INSTALL REQUIRED PACKAGES**

Install all the packages necessary for research using requirements.txt file.

Use command: pip install -r requirements.txt.

1. **RUN THE PYTHON SCRIPT**

Use command - > **python code.py** on terminalto run the main python code file containing scripts for all the analysis.

**Code Walkthrough: -**

1. Load the reviews dataset.
2. Apply data pre-processing on the reviews data frame which includes: -

* Remove missing values using dropna function.
* Expand contractions by specifying certain word mappings like can’t to can not to eliminate discrepancies.
* Remove comments less than 30 characters since they don't carry much information.
* Keep only English reviews for data consistency.
* Convert non-ASCII characters to ASCII.
* Parse Date formats if multiple formats are present in the data.
* Filter reviews and keep only the reviews from 2019 to make the analysis less computationally intensive.
* Convert all comments to lowercase to maintain consistency.
* Add a column to track number of reviews per year.
* Add columns to count number of words, sentences and characters in reviews.
* Replace certain words with similar meanings to a certain word to prevent ambiguity.
* Remove all the html tags.
* Replace certain words to ensure data consistency.
* Drop null values

1. Load the Listings dataset.
2. Rename id column to listing\_id to make it meaningful.
3. Merge both the datasets based on ID.
4. Anonymize the data using a pseudonym mapping technique with the help of a loop which will rename the host names serially by numbers like host 1, host 2, etc.
5. Start with **Exploratory Data Analysis (EDA)** to generate key insights from the data.

* Compare price across neighbourhoods to know the price distribution across Manchester and later use that to check whether price influences the sentiment of reviewers in a particular area be it costly or cheap.
* Find total number of host and listings by counting unique values from the host\_name and listing\_id column.
* Analyze and find neighbourhood with the most number of listings and neighbourhood with the least number of listings which can also be used later on to infer whether number of properties has any effect on overall sentiment.
* Identify the number of people who returned to the same listing or the customer return rate to get a clear idea of how well the hosts are doing.
* Analyze the review trend over the years to check for any patterns when it spiked or dipped to much signifying potential future counts.
* Analyze review length to see whether reviewers normally prefer to keep short or long reviews.

1. Perform **sentiment analysis** on the comments column by using pre-trained VADER sentiment analysis tool which was already imported previously. It will classify each review with 3 scores positive, negative, and compound where the value of compound score can be used to determine the overall sentiment.
2. Create a sentiment sequence column which will track the sentiment scores of each sentence in a review. This will help us in understanding the sentiment shift patterns in reviews which is our **first research question.**
3. Convert the values of sentiment sequence which was in string to list so that we can count the number of transitions present within each review.
4. Visualize using a bar plot for better readability of the sentiment shift patterns.
5. Generate a pie chart to get an idea of how the data is and distribution of sentiments overall.
6. Now we can do some more EDA to find out the driving factors for negative and positive reviews.

Positive Factors: -

For knowing positive factors influencing reviews first filter the comments based on overall sentiment. It should be positive. Once it is filtered, remove stopwords like and, so which do not add value to the sentiment. Finally using the wordcloud library of python generate a wordcloud of the most common words mentioned in positive reviews.

Negative Factors: -

Do exactly same as above, the only difference being filter only the negative reviews this time and generate wordcloud on that.

Using these two analyses, we can easily identify certain aspects which the customers didn’t like and hosts can take action accordingly.

1. Analyze sentiment trends with respect to price, type of room and neighbourhood to understand if those factors play any role towards overall customer sentiment. These can be done by simply grouping columns like price with respect to sentiment, then room type and so on.
2. We can also analyze sentiment trends for a specific listing if we want tailored insights for a particular listing or get an idea of how well that listing did over the years.
3. Use a function to track shift patterns based on sentiment score of each sentence in a review. Based on the values obtained from this function we can count the number of sentiment transitions under different category like positive to negative or vice-versa.
4. Generate a heatmap for sentiment transitions to understand the pattern better. We can also plot sentiment shift patterns within each review to know on an individual basis.
5. Now, for the second research question, start with declaring two paths one for reading and one for exporting for simplicity and load the custom wordlist file which we created.
6. For creating this wordlist file, we must go through the most frequent words used in the reviews overall and then assign them categories based on human knowledge. Since we do not have labelled data for this task, we must follow this manual approach. Assign keywords to categories like Worth, Outcome, Amenities, Environment, Outcome, Neighbourhood, each signifying a particular aspect. We can create more categories if we want more nuanced analysis.
7. Now that the wordlist is ready, we have to split the comments into sentences and then create a custom function which will match the keywords in the wordlist with the review text.

If a match is found, it will check the category mentioned for that word in wordlist and assign that word to the list of that aspect category.

1. After adding to the list, keep a counter and cumulatively add each time a word gets added to that aspect category to get the frequency of that aspect category.
2. Follow same steps for all the categories and finally we will get the frequency count of each aspect.
3. Now we got a way to answer the second research question where our main objective was to determine the priority of aspects within a review. For that we can say that aspects which are mostly mentioned will have higher frequency.
4. We then plot a bar chart to see the counts of different aspects in a descending order.
5. One more advanced way to approach this question is to do sequence and positional analysis of aspects in reviews.
6. To perform positional analysis, we must first declare 3 positional categories – beginning/middle/end. We can calculate the position of each aspect by their index location and then apply a function to calculate an average percentage of aspect positions in reviews.

How will this help us?

Well, from this we will get to know about which aspect do the guests talk first or what comes first in their mind while reviewing a property. This will help us determine the priority of aspects better than the last approach. Aspects which are mentioned first will obviously be of high priority than the ones mentioned later. So, this answers **second research question.**

1. After we get to know about the aspect mostly mentioned in the beginning, we can deep dive a bit more into specifics of that aspect like what sort of amenities are they talking about.
2. Best thing to do after this would be to find the aspect keywords mostly mentioned and then analyze the number of times returning guests have mentioned about that. This will help us understand what made the guests return to that same place and what did not which can be very helpful for hosts in their decision making.
3. If we want further analysis like to explore more about the aspects, we can apply pos tagging to the review comments and then use a function with predefined grammar rules to match the comments and extract noun, adjectives adverbs from the comments.
4. The adjectives and adverbs associated with each of the noun phrases can give us the sentiment related to that aspect. Again, we can use a custom lexicon based approach for this one by defining certain sentiment categories beforehand.

This was all about the code used for my analysis.