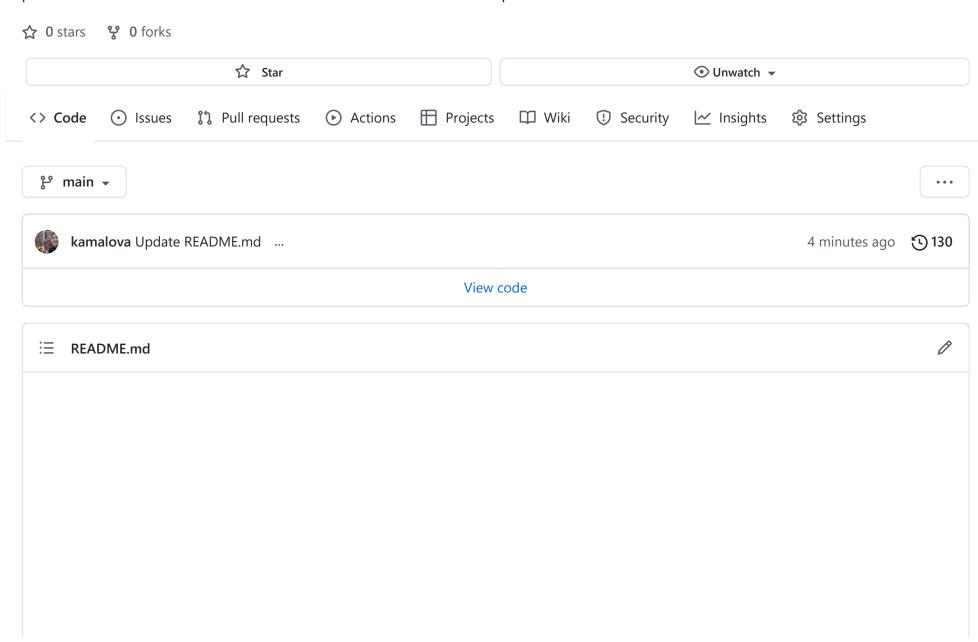
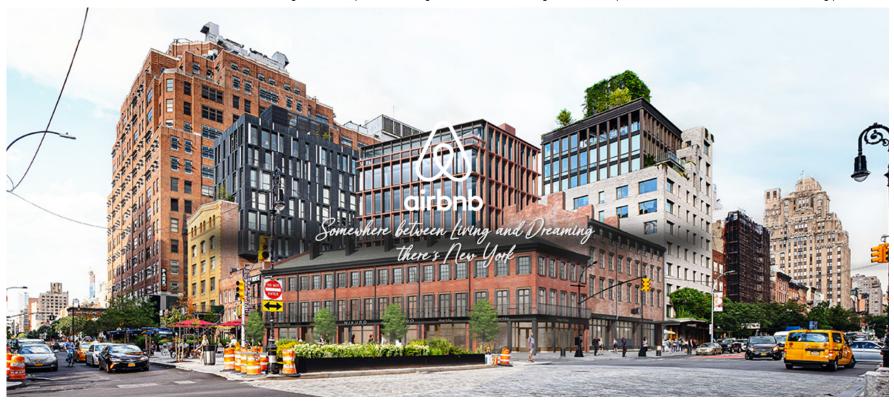
kamalova / NYC-Airbnb-Recommendation-Engine-NLP Public

Capstone: Building a Recommendation Engine with CF to predict sentiment score for all reviewer-listing pairs and making personalized recommendations for each user based on their ranked preferences.





Airbnb Recommendation Engine for NYC through Sentiment Analysis

Author: Nurgul Kurbanali kyzy

Table of Contents

- Overview and Business Statement
- Data Understanding
- Sentiment Analysis
- Recommendation Engine
- Limitations and Future Consideration
- References

For More Information

Overview and Business Statement

About Airbnb (Air Bed and Breakfast): You can host anything, anywhere, so guests can enjoy everything, everywhere.

Nowadays the demand for short and long-term temporary accommodation is increasing thanks to easing travel conditions. This demand positively affects the number of online platforms that allow you to make reservations before traveling. Airbnb is one such platform, which allows travelers to make accommodation reservations based on the fact that the host leases all or part of his or her home to the traveler.

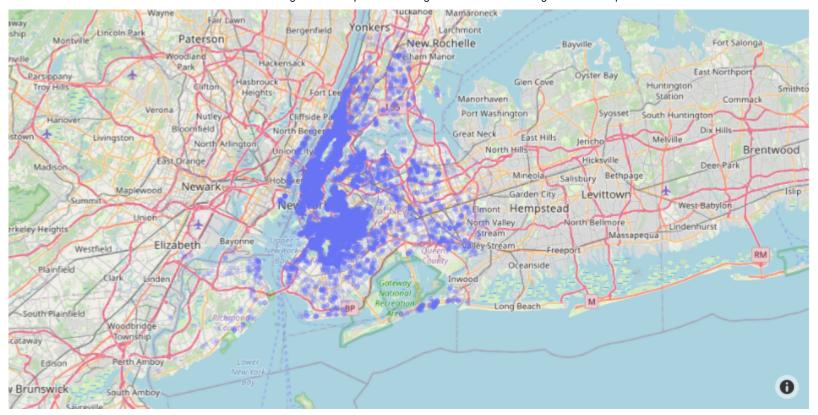
Customer reviews play an important role in the customer's decision to purchase a product or use a service. Customer preferences and opinions are affected by other customers' reviews online, on blogs or over social networking platforms. The main goal of this work is to combine both recommendation system with **Collaborative Filtering (CF)** and sentiment analysis in order to recommend the most accurate listings for Airbnb users based on their preferences in New York City. Since both domains suffer from the lack of labeled data, to overcome that, this project detects the opinions polarity score using **NLTK VADER** (Valence Aware Dictionary and Sentiment Reasoner) Lexicon.

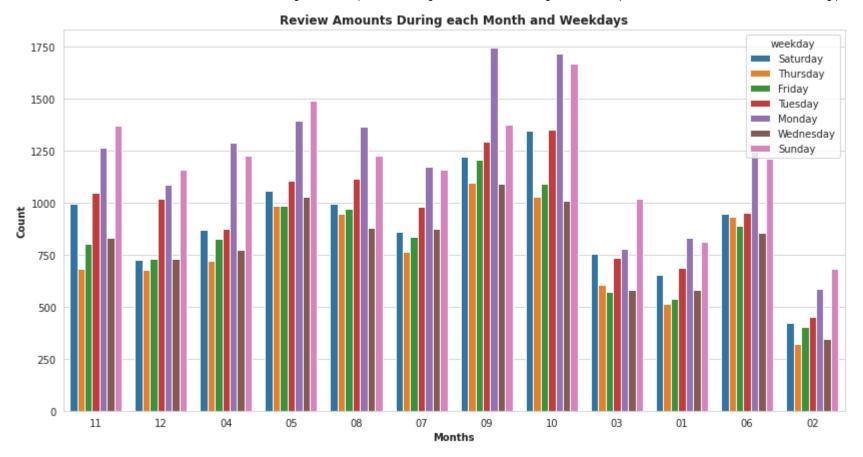
Overall, the project went through the following sections:

- Exploring available AirBnb listings in NYC
- Measuring polarity/sentiment scores with the help of vader_lexicon. This polarity measurement adapts to *pos, neu, neg*, and *compound*. By simply taking the compound from these values, a new feature was created on the data.
- Finally building a recommendation engine with **CF** to predict sentiment score for all reviewer-listing pairs and making personalised recommendations for each user based on their ranked preferences.

Data Understanding

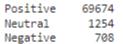
The dataset is obtained from Inside Airbnb. It is is a mission driven project that provides data and advocacy about Airbnb's impact on residential communities. For the purpose of this project we downloaded the most recent quarterly datasets between December, 2021 - September, 2022 which includes information and metrics for *listings and reviews* in **NYC**. Datasets have been concatinated, preprosessed and some *EDA* techniques have been applied on.



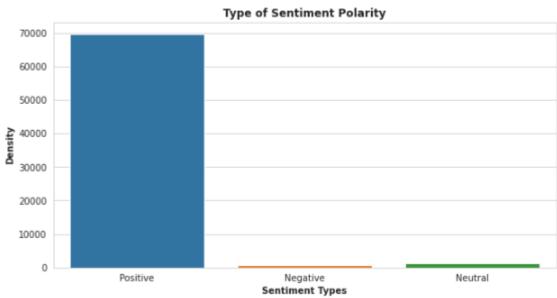


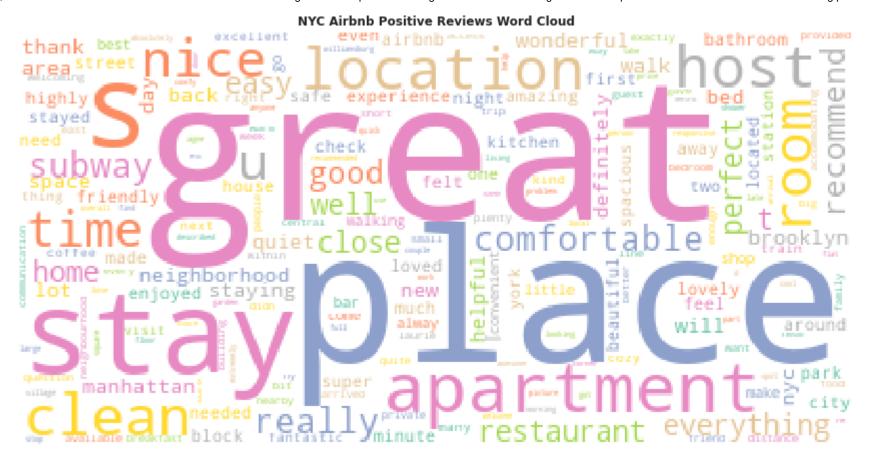
Sentiment Analysis

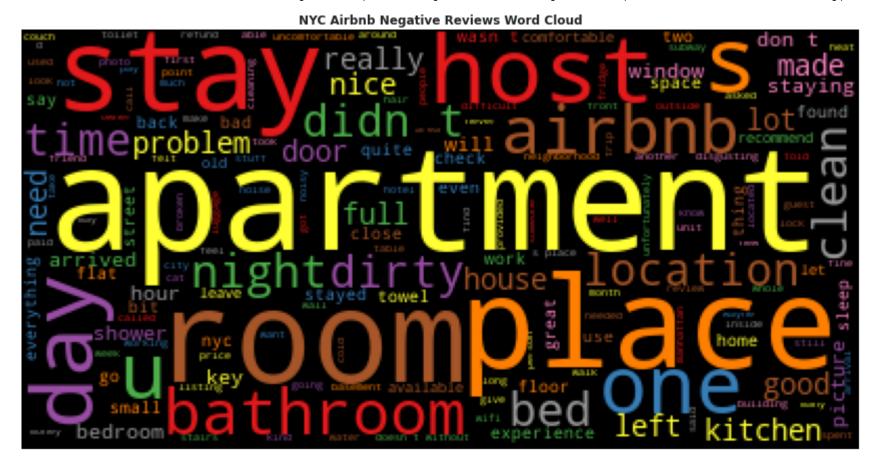
Prior to start my analysis I gave a brief information what is a **Sentiment Analysis(opinion mining)**. Further applied text preprocessing steps for each reviewer's comments (including language detection). Thanks to the NLTK's Vader lexicon I was able to calculate the compound polarity score of each comment. The compound score is the sum of positive, negative & neutral scores which is then normalized between -1(most extreme negative) and +1 (most extreme positive).



Name: sentiment_type, dtype: int64





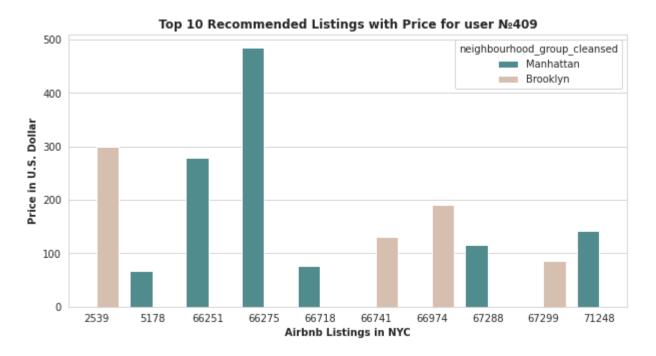


Recommendation Engine

In this section I ended up building a recommendation engine with **Collaborative Filtering(CF)** to predict sentiment score for all reviewer-listing pairs and make personalised recommendations for each user based on their ranked preferences. The **CF** method focuses on collecting and analyzing data on user behavior, activities, and preferences, to predict what a person will like, based on their similarity to other users. To plot and calculate these similarities, collaborative filtering uses a matrix style formula. An advantage of collaborative filtering is that it doesn't need to analyze or understand the content (products, films, books). It simply picks items to recommend based on what they know about the user.

Grid Search Cross Validation computed accuracy metrics for an algorithm on various combinations of parameters, over a cross-validation procedure **SVD Algorithm**

Accuracy on Test set- RMSE: 0.1629



Limitations and Future Consideration

The results show that our model outperforms in terms of the root mean squared errors of the generated predictions. I would suggest to try using models which involve regularization like: *Matrix factorization using Alternating (optimization algorithms) least squares. Matrix factorization using Bayesian Personalized ranking.* Balancing dataset with state-of-the-art oversampling or under sampling techniques such as *SMOTE* during pre-processing. Applying clustering techniques like K-means or Self-Organizing Map (SOM) also can lead to more accurate results. In addistion the time factor can play an important role in providing effective personalized recommendations and consequently improving the accuracy of predictions.

References

Comprehensive Guide to build a Recommendation Engine from scratch (in Python)

Sentiment Analysis: A Definitive Guide

Integrating contextual sentiment analysis in collaborative recommender systems

For More Information

You can review my full analysis in my Jupyter Notebooks:

- Listings Exploratory Data Analysis,
- Review Based Sentiment Analysis
- Recommendation Engine

or project presentation.

For any additional questions, please contact Nurgul Kurbanali kyzy at nurkamalova@gmail.com

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

Languages

Jupyter Notebook 100.0%