Prediction and Recommendation Engine for Airbnb

CMPE - 256 Large Scale Analytics Term Project Report

Team Perceptrons

Manisha Shivshette <manisha.shivshette@sjsu.edu>
Arshiya Hayatkhan Pathan <arshiyahayatkhan.pathan@sjsu.edu>
Tarun Arora <tarun.arora@sjsu.edu>

Guided By

Professor Magdalini Eirinaki

1. Introduction

Motivation:

Machine Learning and Artificial Intelligence are at the forefront of innovation in today's world. With Data being the most valuable resource in the 21st century more and more data-driven business decisions are being made with machine learning and large-scale data analytics.

Over the last two decades, hotel industries have seen the power of their brands' promise move away from their marketing teams into the hands of the consumer. Aided by the rise of online review platforms like TripAdvisor, hotel prices can easily be compared with other properties in the market. Considering this competitive market, every hospitality brand is trying to provide better services to customers by utilizing machine learning advances in the technology sector. Hence, in this project, we proposed a few competitive solutions useful to Airbnb users and hosts. We are using huge data of Airbnb NYC to create a recommendation model to recommend listings to the users using collaborative filtering algorithm and predict optimum listing price to the host.

Objectives:

Airbnb application supports adding your apartment/room to the platform as a host and booking an apartment/room as a user. The objective is to use Airbnb's publicly available data in order to develop business logic and recommendation engine to support customers as well as hosts in order to recommend favorable properties and to predict optimal pricing for host respectively.

Predict Rating from review comments

Performing sentiment analysis of the reviews and categorize reviews into a different set of ratings using machine learning algorithms. This is useful to visualize the user ratings better for a particular property.

Recommend the listing to the user, based on collaborative filtering

Building a feature matrix based on the user reviews ratings computed from sentiment analysis in order to predict recommend properties to the user.

Predict the optimum listing price for the host

As the listing prices of the property changes based on holidays, weekends, season and month. The idea is to analyze the price trends over the period and predict the optimum listing price for the selected period. This price can be used by the host to maximize profit and increase the occupancy rate.

2. System Design & Implementation details

Algorithms considered/selected:

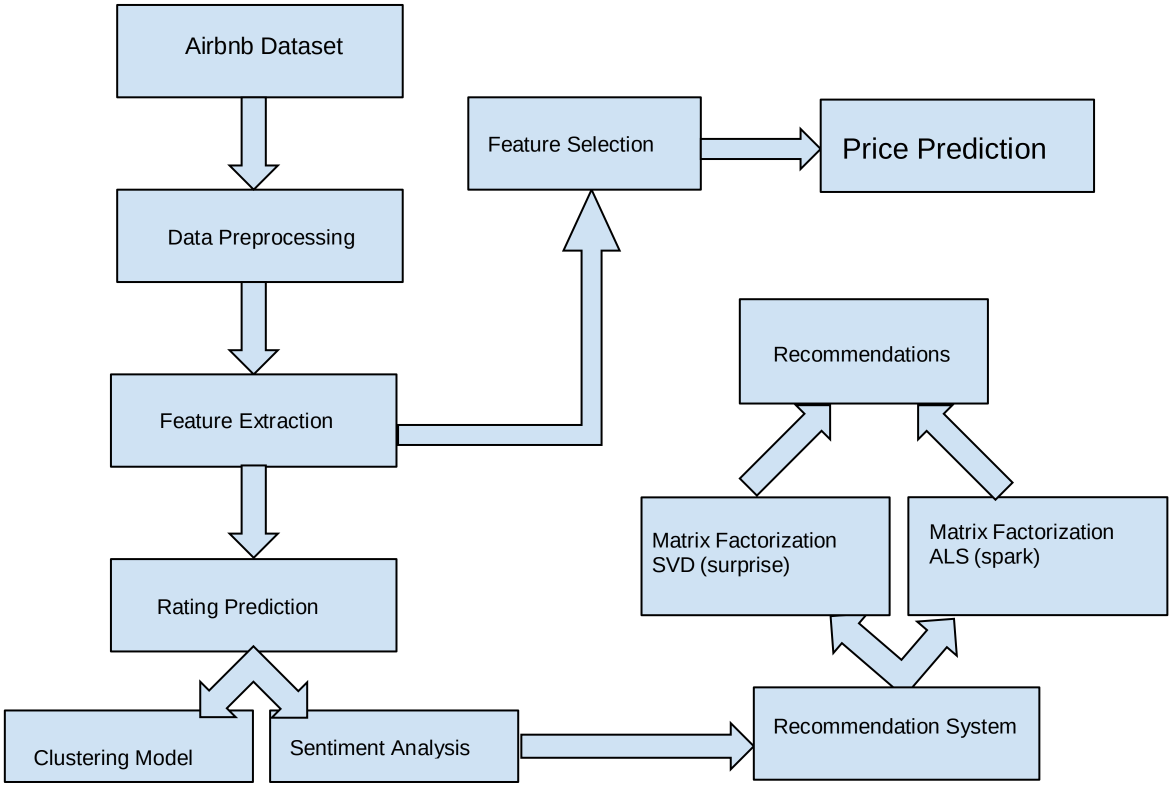
Rating Prediction:

- We considered two major algorithms to predict the rating, first, we implemented ML model based on K-Means clustering algorithm to predict the rating using review comment text and later we implemented ML model based on NLTK sentiment analyzer.

Recommendation System:

- As we don't have any data about the user except the name of the user and reviews, we decided to recommend listings to the user, based on collaborative filtering algorithms.
- We used Surprise library which is python scikit for building recommender systems. We considered different prediction algorithms such as basic collaborative filtering algorithm using user similarity and item similarity, matrix factorization using SVD. As data is sparse and very big, user based and item based filtering algorithms failed to give predictions. Hence we selected matrix decomposition algorithm using Singular vector decomposition (SVD).
- We also used the matrix factorization algorithm using Alternating Least Squares (ALS) with which we were able to scale data to 700k users and 35k listings.

System Design:



Technologies & Tools used:

Libraries: pandas, numpy, matplotlib, Surprise, Spark ml, Spark MLib, NLTK, scikit
Python 3 with Jupyter Notebook, GitHub for repository and collaboration

3. Experiments / Proof of concept evaluation

Dataset(s) Used

Dataset: AirBnB Open data NYC
Size: 910.7 MB
Link to source code: https://github.com/DeveloperManisha/CMPE256-Airbnb

The dataset includes the following:

calendar_detail.csv: Details about listing, date, availability, and price in New York.
listings_detail.csv: Insights into the description of the property including facilities and details about the host.
listings_summary.csv: The description of host and location of the property, price, last review, and availability.
neighbourhoods.csv: For a particular listing, details of the neighborhood group.
reviews_detail.csv: Review comments by the user for a specific property.
reviews_summary.csv: The date on which the listing was reviewed by the user.

Module 1 - Rating Prediction

Rating prediction uses review_details.csv data, it contains information like listing_id, review_comments,reviewer_id, reviewer name, date. Prediction uses only reviewer_id, listing_id and review_comments to predict the rating.

Module 2 - Recommend Listings using collaborative filtering

Rating prediction module generates a ratings.csv which contains information like reviewer_id, listing_id, and Rating. This data is used in order to predict ratings to listings, which user has not visited/reviewed.

Module 3 - Optimal Price Prediction

In order to predict the optimal prices of the property for the property host, calender_detail.csv data is used. It contains the date and price of the property along with its listing id and whether it is available or not on the specific date.

Data Preprocessing Decisions

Module 1-Rating Prediction

Input data of review details contained 802000 review comment records and we preprocessed all those records before the training model.
Exploration: Analyzed review comments data for missing values, duplicates etc.
Cleaning: Removed symbols, redundant characters from Review Comments and processed missing and duplicate values
Preprocessing: Text data cannot be processed as is by a few ML algorithms hence, tokenized data using TF-IDF for approach 1.

Module 2 - Collaborative Filtering

Generated_ratings.csv consists of 800995 rows with 703051 unique users and 34839 unique listings.

As our goal is to use collaborative filtering to recommend listings to the user, we considered different algorithms such as User-based collaborative filtering, Item-based collaborative filtering and Matrix factorization using SVD and ALS. In all the algorithms the matrix will be created which will be of size m*n where m is the number of unique users and n is the number of unique items. Also, Many users have given ratings for only one listing out of 35K listings. So matrix created is very sparse. We were not able to select all the users and listings for SVD algorithm due to dataset sparsity and machine limitations.

After analyzing data, we decided to limit our data to users who have given at least 3 reviews and listings which have got more than 10 reviews for Recommendation using algorithm SVD. For recommendation using Spark ALS, we were able to provide recommendations by taking the whole dataset. So we included all the users and listings.

Module 3 - Optimal Price Prediction

The data consists of listing id and with the date and price of the property. As the price of the property depends on a lot of factors which include season, month, weekends and public holiday, the data should be transformed in order to incorporate all these variables.

Season => Season is allocated based on the month of the date.
Month => Month name is extracted from the date
Day => Days are extracted based on the date whether its Monday or Tuesday ...etc
Holidays => A list of public US holidays are taken into account and added to the data of the date is a given public holiday

Methodology followed:

Module 1-Rating Prediction

As there were no actual ratings available in the dataset, two approaches were evaluated to predict rating from review comments

Approach 1 - Predict Rating using Clustering Algorithm

- Create TF-IDF Matrix from input review comments
- Generate Clusters of similar review comments
- Identify the cluster containing maximum positive tokens
- Replace the cluster label with the rating
- Export listing id and rating

Approach 2 - Predict rating using Sentiment Analysis Algorithm

- Repeat
- Clean Review Comment text data
- Using NLTK Sentiment analyzer generate polarity scores
- Using positive and negative polarity scores, compute the rating
  until all ratings are computed

Module 2 - Recommendation system using Collaborative filtering

Approach 1 - Predict Rating using Matrix factorization - SVD and recommend top-N listings

- Dividing the dataset into 90% of data and keeping 10 % for checking unbiased accuracy.
- Select the model by choosing prediction algorithm - SVD and tune the parameters using Grid search with 3 fold splitting.
- Evaluate the model by testing on part 1 as well as part 2 and compare biased and unbiased accuracy.
- Using K fold cross validation train the model.

- Build the test set by considering users and all the listings which user has not given reviews for.
- Get the predictions for all the listings.
- Recommend top 3 listings to the user.

Approach 2 - Predict Rating and recommend top N listing using Matrix factorization - Alternating Least Squares (ALS)

- Split data-set into training, validation and test set.
- Train the model and tune the hyperparameters used in ALS the algorithm such as maximum iterations, rank and regularization parameter
- Evaluate the model on the validation set
- Improve the performance score using k-fold cross validation
- Test the model on the test set and calculate RMSE.
- Recommend top 3 listings to all users.
- Save the model using persistence so that it can be loaded later.

Module 3 - Optimal Price Prediction

Approach 1 - Predict Price using Linear Regression

- Split data into training and testing sets
- Evaluation various parameters to take into account for training data
- Converting categorical values to numerical values
- Training the test set by price and listing id along with variation in month and seasons.
- Predicting price for the testing set

Approach 2 - Predict Rating using Matrix Factorization - SVD

- Splitting the data into test and train sets
- Data preprocessing for matrix creation
- Converting values to numerical formats
- Predicting prices based on SVD matrix

Graphs:

Below graph shows the comparison between ratings generated by clustering approach and ratings generated by sentiment analysis for same review comment. We can observe that there are few discrepancies between rating generated. The ratings predicted by the sentiment analysis algorithm considers negative sentiment in the review comment and it is independent of the review comment length. Hence it is more accurate.

blue: Prediction using Sentiment Analysis **green**: Prediction Using Clustering

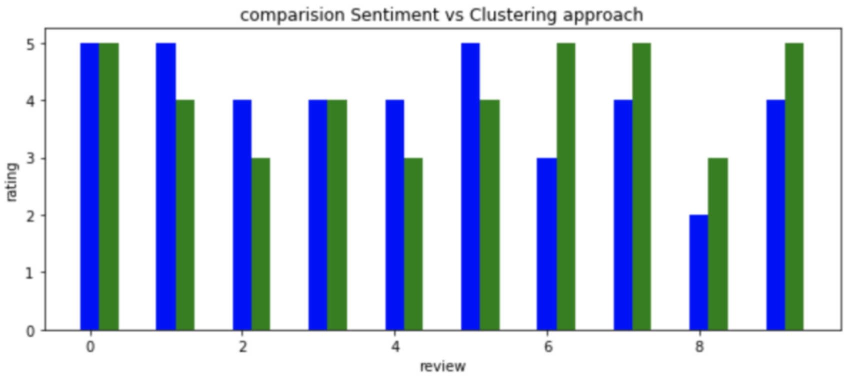


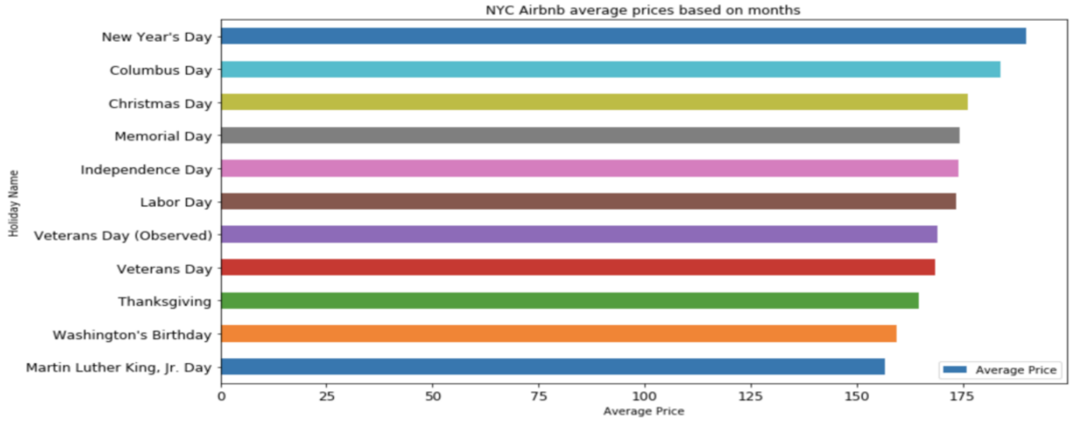
Fig: Comparison between rating generated by clustering and sentiment analysis

Sample rating analysis generated by clustering and sentiment analysis

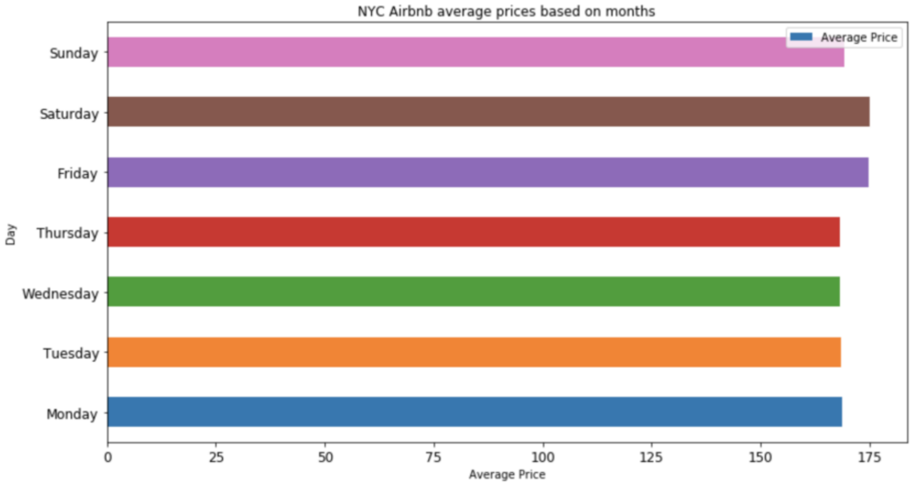
Review	Rating (Clustering)	Rating (Sentiment Analysis)
Great place. Helena was prompt to reply and very helpful. The place has a great location short walk away from Times Square and is close to everything and anything you'll need! **Analysis: sentiment analysis model evaluates better rating irrespective review length**	3.0	5.0
One bad thing is that the club which is at downstairs is really noisy at weekend. So someone who really sensitive to sleep needs to think about it. But this house is still a good place though! **Analysis: sentiment analysis model understands the negative context better**	5.0	3.0
WARNING: NO central working heat. This makes the unit uninhabitable. A 5in by 5in space heater does NOT heat an apartment. This unit is illegally operating. Air BNB has not honored my resolution requests and Don has not honored my request for a refund after we froze all night when the place was mis-listed. He believes I should pay for nights I did not use and stay, I believe I was misinformed about the situation and legal habitability of the unit. **Analysis: sentiment analysis model captures negativity in the review perfectly where clustering fails miserably as it only considers single and frequent words**	5.0	1.0

Optimal Price Prediction

Below graph shows the trends in the pricing of properties based on public holidays. As the prices tend to be more than the mean prices during holidays, hence, this can be taken into consideration while predicting the optimal price of the property.



Below graph shows the trends in the pricing of properties based on the days of the week. As the prices changes during weekends, hence, this is also taken into account to predict the optimal price of the property.



Analysis of results:

After analyzing clustering based model, sentiment analysis based model and considering the accuracy of ratings predicted by sentiment analysis model, we decided to use ratings generated by sentiment analysis model as input to the recommendation system of module 2.
By Evaluating Matrix factorization model using SVD and ALS, we conclude that SVD gives better accuracy than the ALS algorithm. RMSE obtained using SVD is 0.8505 whereas ALS has higher RMSE of 2.1. If we want to support large dataset, the model using ALS is the best choice.

4. Discussion & Conclusions

Decisions made:

Rating Prediction

Comparing two approaches we decided to use ratings generated by Approach 2- rating predictions using sentiment analysis. As the ratings generated were based 1) more sophisticated algorithm 2) it was scalable for this large amount of data 3)it analyzed sentiment from the text rather than just count of positive and negative words.

Whereas the clustering based approach was 1) not scalable and need to use dimensionality reduction to operate on large scale data 2)could not improve proposed algorithm to work on all reviews data in single pass 3) it can not differentiate word “not good” from word “good”.

Recommendation System:

After comparing different collaborative filtering algorithms, we decided to go for matrix factorization as it gives better RMSE and supports large dataset. SVD is better-provided dataset is not too sparse. Model using SVD cannot be scaled to support large and sparse dataset. If we want to support large dataset, the model using ALS is the best choice.

Price Predictions:

After exploring Linear regression and SVD techniques, Linear Regression algorithm was used to calculate and predict ratings as the time to handle the huge amount of data was much better supported in linear regression whereas in SVD the matrix formation was so huge it took forever for our underpowered systems to work. Also, the tweaks to improve was impossible to compare and run again and again and it was not scalable enough to take into account.

Things that worked

Data exploration, cleaning, preprocessing and results of the first approach helped to analyze the issues with the first approach and helped to compose scalable, robust and reliable approach to predict rating. After exploring the data, by counting the number of reviews received per listing and number of reviews given by individual users, data set was limited to support collaborative filtering using SVD. The model trained using this algorithm provided ratings with quite good RMSE. In order to handle large dataset, we also used the ALS algorithm. Hence we were able to scale our dataset to support all users and listings.

Things that didn't work well

Predicting the rating using the clustering approach, was naive due to lack of knowledge, experience, and methodology. We could not formulate evaluation matrix for this problem and just used survey data of users to compare the results of the proposed algorithm. As the Airbnb dataset is too large and users have given only one or two reviews, user based and item based algorithm could not be used to build and evaluate the model. Also for matrix factorization using SVD, we had to limit the dataset. In order to get price predictions, SVD matrix formation was very slow to change and build and test with different tweaks.

Conclusion

After evaluating clustering and sentiment analysis based approaches, we decided to use rating generated by the sentiment analysis model to train the recommendation algorithm.

After evaluating different collaborative filtering algorithms like user-based filtering, item-based filtering, and matrix factorization, we concluded that matrix factorization is the best algorithm to provide recommendations for the large and sparse dataset like Airbnb dataset. Also in order to provide recommendations over the large dataset, Spark's ALS implementation is more suitable.

Building this module helped to build skills required for 1) data preprocessing 2) building ML models for large-scale data 3) evaluating methods. 4) visualize data and use it for predictions.

5. Project Plan / Task Distribution

Module	Tasks	Responsibility

Module 1 Rating Prediction From review text	1. Study associated algorithms 2. Text Data preprocessing 3. Feature Extraction 4. Predict rating using clustering 5. Predict rating using sentiment analysis 6. Analysis of both approaches 7. Evaluation and graphs in module 1	Manisha Shivshette (012560353)
Module 2 Recommendation System using Collaborative filtering	1. Data exploration and preprocessing 2. Study and compare different Collaborative filtering algorithms. 3. Choose algorithm to support large dataset. 4. Predict rating using matrix factorization -SVD 5. Predict Rating using Matrix factorization ALS 6. Evaluate models using RMSE 7. Recommend top 3 listings to users	Arshiya Pathan (012431536)
Module 3 Predicting Optimal Price of the Property	1. Data exploration 2. Data Preprocessing 3. Evaluation of public holidays 4. Visualization based on price and variations 5. Linear Regression to predict the ratings	Tarun Arora (012429772)
Report and Presentation	Document All the steps performed.	All

References

1. https://opensourceforu.com/2016/12/analysing-sentiments-nltk/
2. https://skift.com/2016/04/11/how-machine-learning-is-making-hoteliers-smarter/
3. https://dataplatform.cloud.ibm.com/exchange/public/entry/view/99b857815e69353c04d95daefb3b91fa
4. https://towardsdatascience.com/how-did-we-build-book-recommender-systems-in-an-hour-the-fundamentals-dfee054f978e
5. https://surprise.readthedocs.io/en/stable/getting_started.html
6. https://spark.apache.org/docs/preview/ml-collaborative-filtering.html
7. https://spark.apache.org/docs/preview/api/python/pyspark.ml.html#pyspark.ml.recommendation.ALS