TABLE OF CONTENTS

Section Page

1. ABSTRACT 1

2. INTRODUCTION 1

2.1 Context

2.2 Objectives

2.3 Related Work

3. MATERIALS AND METHODS 2

3.1 Dataset

3.2 Technologies

3.3 Implementation

3.3.1 Data Preparation

3.3.2 Regression

3.3.3 Clustering

4. RESULTS 4

5. DISCUSSION 5

5.1 Relevance of Solution

5.2 Limitation

5.3 Future Work

6. REFERENCES 7

Link to source code: https://github.com/attayeba/Airbnb-price-prediction

Abstract

With services allowing customers to book rooms to accommodate them during their next trip being implemented, a large amount of data about pricing habits has been collected. A service that shook the hotel industry is Airbnb. This service allows home owners to offer their houses as hosting locations. To help home owners set competitive and fair prices to their houses, we will implement a system which estimates the price at which a house should be offered. This will be done through the use of big data analytics techniques such as clustering and regression.

Introduction

Context

In recent years, many applications that changed our travelling habits have been implemented. Amongst the companies having a big impact on the travelling industry, Airbnb stands out when it comes to finding affordable housing for your next vacation. With multiple housing options presented to Airbnb users, home owners who rent out their property have to stay competitive in order to create some interest towards their locations. To help Airbnb hosts stay competitive, we implemented a system which recommends the price at which a property should be set.

Objectives

The objective of our project is predicting the value at which a house should be offered while minimizing the mean absolute percentage error (MAPE).

Related Work

It is estimated that Airbnb has hosted over 60 million customers in 34000 cities across the globe [1]. With such a high popularity, predicting the price of Airbnb houses has been recurring challenge. Amongst the projects trying to solve the presented problem, two studies seem to be highly similar to ours.

At their root, both studies used clustering and classification algorithms as their initial model. One study used K-means to cluster similar listings together while the other one referred to the K nearest neighbors (KNN) algorithm [1,2]. Once the listings grouped, there are multiple ways of estimating the price. One study suggested to take the average price of the K nearest neighbors [2]. However, that idea was shown to not be the most accurate. On the contrary, as displayed in multiple studies, using regression algorithms presents the most accurate results [1,5].

In short, among the explored related works, regression algorithms predict the most accurate results. Clustering the data prior to applying the regression has also displayed some promising results.

Materials and Methods

Dataset

The dataset we will be using was taken from insideairbnb.com [3]. It contains information about houses which were advertised and rented through Airbnb around the globe. As many cities have limited information, we decided to go with New York as there is a fair amount of data from said city. The dataset contains 48853 rows with 96 features each. The columns hold various types of information such as the host's information, information specific to the physical location and renting policies. As the feature space is large, we decided to focus on the information pertaining to the location and the house itself [5]. Below are some of the columns which present the most insight about the location:

- Neighborhood: location within the 5 main regions of New York city.
- Property Type: whether the location is a house or an apartment.
- Room Type: a location can be rented as an entire home/apartment, private room or shared room
- Accommodates: number of people the location can welcome
- Bathrooms: number of bathrooms
- Bedrooms: number of bedrooms
- Beds: number of beds
- Bed type: type of bed included in the location
- Price: price per night of stay in USD
- Minimum and Maximum stay: time limitations regarding the renting period
- Longitude and latitude: Coordinates of the listing
- Availability: number of days the house is available throughout the year
- Rating: overall rating of the location

In order to perform an analysis of the data, we will split the dataset into a training and testing set. The training set will be 70% of the original dataset while the testing set will account for the remaining 30%.

Technologies

To complete this project, we will be using PySpark and Python and their many predefined libraries. The main libraries used are Pandas, NumPy and PySpark's machine learning library. These will allow for an analysis of the dataset without reimplementing the algorithms which will be used.

Implementation

To achieve our goal of estimating the price at which houses should be offered, our implementation can be divide in three major steps. To begin with, we did all the necessary preparation for the data to be easily processed by the subsequent steps. Next, we passed the data through a gradient boosted regression tree (GBRT) algorithm to predict the price. Finally, we took a second approach for predicting the price by clustering the data before perform the regression. Below is each step with more details. The source code can be found at the following link. The files of interest are *DataPreparation.py*, *GBRT.py* and *ClusteredRegression.py*.

Data Preparation

As our original data set was large, some preprocessing was necessary before any regression or clustering could be performed. The original dataset contained a great amount of information. Our main focus was information pertaining to the house itself. We started by extracting the columns of interest. These columns were decided simply by reading the description of each column. This reduced our feature space from 96 to 14.

With the feature space reduced, we followed by removing data points which did not conform to some criteria. Listings with missing information about the price, bedrooms, beds, location and number of people the host can accommodate were removed. Multiple houses were also priced at $0 per night. These listings were also removed. Moreover, with 222 distinct cities, multiple cities had very little listings, thus reducing our prediction performance. Consequently, cities which appeared less than 245 times (0.5% of our dataset) were eliminated. With a reduced dataset, our final step in data cleaning was removing outliers. Outliers were noted as listing with prices located more than three standard deviations away from the mean. This left us with a data set of 35843 rows and 14 features.

The next necessary step in our data preparation was placing the data in a format that would be accepted by the GBRT and K-means algorithms. Since both algorithms only accept numerical values, we one-hot encoded our categorical features. As represented in the tables below, each distinct value of a categorical feature was now represented by its own column. If the listing contained said feature, it was attributed the value of 1 and 0 otherwise.

ID	Neighborhood
A	East Harlem
B	Upper West Side
C	Harlem

⇨

ID	East Harlem	Upper West Side	Harlem
A	1	0	0
B	0	1	0
C	0	0	1

Consequently, when computing the distance, a categorical value would have a distance of 0 if both listings share the same value and 1 otherwise. Finally, the resulting data frame was saved to a csv to reduce computation time in the subsequent steps.

Regression

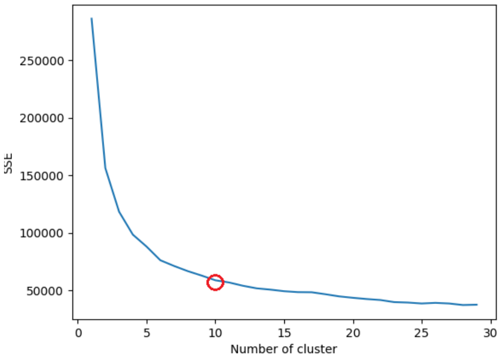
With our dataset preprocessed, we split our data into a train and test set. Seventy percent of our data was used for training and the remaining thirty served for testing. With the data preprocessed and split, all that remained was passing our data to our regression algorithm. As gradient boosted regression trees were reported to predict accurate results, we decided to employ it [6]. Rather than reimplementing the algorithm, we used the already implemented version of PySpark's machine learning library [7].

GBRT's are an ensemble of random forest regression trees. With multiple learners (ensemble of random forests), GBRT goes through three main phases. First, the early learners fit the simple data points. Next, with some data points fitted, the algorithm identifies the error or complexities it went through. It finally uses the gathered information to fit the more complex data points through its last set of learners. Lastly, the set is combined while attributing a weight to each learner. This outputs a final regressed value for each point in the test set.

Clustering

Wishing to obtain the most accurate results, we followed an idea outlined in previous studies. Rather than performing the regression on the whole set, we decided to first cluster the data [1]. To perform the clustering, we used the K-means implementation of PySpark's machine learning library [8].

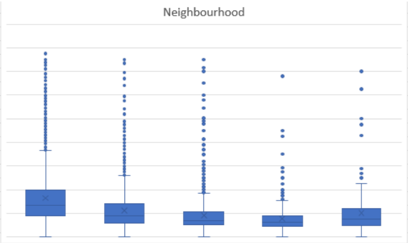
Prior to clustering, we performed the elbow test on the preprocessed data to appoint an accurate cluster value. Our test provided the output displayed in the figure below.



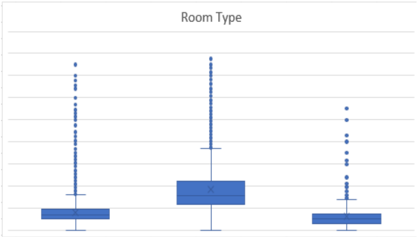
To validate the value of our elbow method, we decided to iteratively test each value within a range of 3 from our suggested K value of 9. Once the data clustered, we performed a regression using GBRT on each cluster.

Results

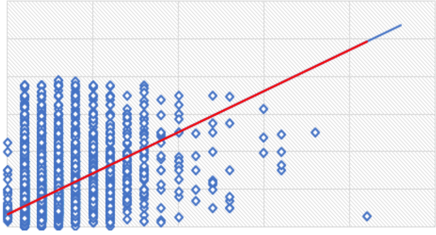
As part of our data preparation phase, we analyzed each feature with relation to the price. This was simply to give us a better understanding of how each factor will affect the price. Categorical data is represented with box plots and numerical data is displayed in scatter plots with the best fit line in red. Below are the six feature which presented a strong relation with the price. The remaining features had a weak relation with the price.



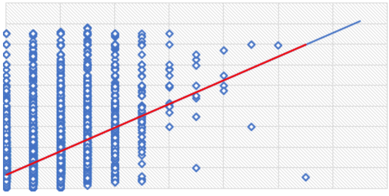
Neighborhood



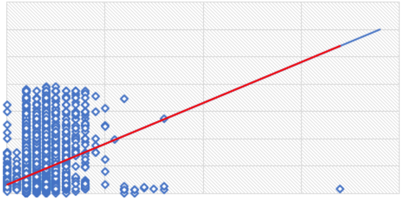
Room Type



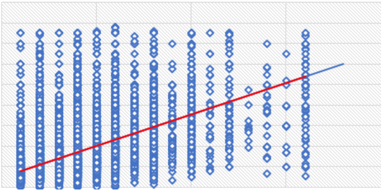
Beds



Accommodates



Bedroom



Bathroom

Next, to estimate the accuracy of our predicted model, we deployed both the mean absolute percentage error (MAPE) and the root-mean-square error (RMSE). Since the RMSE is influenced by the range of the predicted value, the MAPE gave us a better understanding of our results. The MAPE can be found using the formula below:

$$MAPE = \Sigma \frac{|Actual_i - \ Predicted_i|}{Actual_i} * \frac{100}{n}$$

As we tested both values at multiple stages, the results are displayed in the table below.

STAGE	RMSE	MAPE
Un-preprocessed data	68.83	34.28%
Preprocessed data	58.43	27.63%
Significant Features	62.29	29.54%
Clustered K=6	38.88	24.67%
Clustered K=7	51.82	24.11%
Clustered K=8	51.88	24.46%
Clustered K=9	51.13	24.24%
Clustered K=10	39.53	24.27%
Clustered K=11	50.56	24.68%

Discussion

Relevance of Solution

From the table in the results section, we can see that preprocessing the data set had a positive effect. By simply cleaning our dataset and removing erroneous results, we improved our MAPE by 6.65%.

Moreover, to put in practice the significant features extracted, we tested out the accuracy of a model which only has access to the neighborhood, room type, number of bedrooms, beds, bathrooms and individuals it can accommodate. The regression with only 6 features was worse by 1.91%. That being said, with only these six features, one can make a prediction almost as accurate as when they have all 14 features available.

Next, it is observable the clustering the data proved to be beneficial. An improvement as high as 3.56% was displayed between the simple regression and the regression on the clustered data. In addition, the elbow method was in fact a quick and simple method to approximate a good K value. Lastly, we decided to keep K at 7 as it provided the lowest MAPE.

Limitation

As a great amount of effort was allocated to improve the MAPE, we wanted to better understand our results that we considered to be fairly poor. Upon comparing to a similar study performed, we found that their MAPE was 17.34% which is not too far from our findings [6]. We believe that with better feature engineering, our MAPE could be further improved.

To further understand why the error rate is as such, we decided to look further into the data. Comparing multiple listings, we came to the realization that multiple data points with the same feature values had a large disparity in prices. We found 87 private room apartments in East Harlem with one bedroom, bathroom and bed with prices ranging from $25 to $250. Using the longitude and latitude to locate the listing with the highest and lowest price, we found that they are located only 790 meters apart from one another. While verifying if this was an odd case, we found multiple groupings of identical listing with large variability in prices. We came to the conclusion that Airbnb prices are hard to predict as they are subjective.

Future Work

For future work, we outlined 3 aspects that might help improve the study. To begin with, we lost about 27% of our data to preprocessing. A larger dataset would allow for more information, thus probably having a better understanding on how individuals set prices.

Next, although the renting period was a feature in the original dataset, more than half the listings were missing said information. Having the renting period could give a better understanding on the price variability. For instance, it is possible that prices increase during holidays and summer.

Finally, with the host information given, it would be of great interest to examine if the host's information has an effect on the renting price.

References

[1]"Airbnb Pricing Predictions", *Airbnb-pricing-prediction.herokuapp.com*, 2017. [Online]. Available: https://airbnb-pricing-prediction.herokuapp.com/index.html. [Accessed: 07- Feb- 2018].

[2]"Machine Learning Fundamentals: Predicting Airbnb Prices", *Dataquest*, 2017. [Online]. Available: https://www.dataquest.io/blog/machine-learning-tutorial/. [Accessed: 09- Feb- 2018].

[3] *Inside Airbnb*, 2018. [Online]. Available: http://insideairbnb.com/get-the-data.html. [Accessed: 17- Apr- 2018].

[4] P. Kaur, M. Goyal and J. Lu, "Pricing Analysis in Online Auctions Using Clustering and Regression Tree Approach", *Springer*, 2012.

[5] J. Wang, H. Lu and C. Xu, "Predicting listing price on Airbnb dataset", 2017. [Online]. Available: https://cseweb.ucsd.edu/classes/wi17/cse258-a/reports/a052.pdf. [Accessed: 17- Apr- 2018].

[6] L. Chen and W. Liang, "Airbnb price prediction using Gradient boosting", 2017. [Online]. Available: https://cseweb.ucsd.edu/classes/wi17/cse258-a/reports/a043.pdf. [Accessed: 19- Apr- 2018].

[7] "Classification and regression - Spark 2.3.0 Documentation", *Spark.apache.org*, 2018. [Online]. Available: https://spark.apache.org/docs/latest/ml-classification-regression.html#gradient-boosted-tree-regression. [Accessed: 19- Apr- 2018].

[8] "Clustering - Spark 2.3.0 Documentation", *Spark.apache.org*, 2018. [Online]. Available: https://spark.apache.org/docs/latest/ml-clustering.html#k-means. [Accessed: 19- Apr- 2018].