Anomaly Analysis in Airbnb Data to Detect Fraud

Swapnil Ashtekar, Nupur Kulkarni, Dhruva Patil

CS 435 Final Report,April 2016

Contents

1 Introduction

Airbnb is a trusted community marketplace that allows property owners and travellers to list, discover and book unique accommodation around the world from 2008. They are focused on creating a place where people can belong anywhere. Part of that sense of belonging comes from trust amongst their users and knowing that user safety is an utmost concern. Currently it has hundreds of employees across the globe supporting property rentals in nearly 25,000 cities with over 500,000 listing across 192 countries. The Airbnb community users' activities are conducted on the company's website and through its iPhone and Android applications.

Trust is a key part of Airbnb as a marketplace and the responsibility is equally up to the users of Airbnb as it is of the company itself. Good reviews help users to select or rent accommodation that have been highly rated and avoid the ones that are low rated. Thus ratings and reviews play a significant role for users to make a choice among the available options. The Airbnb review system helps users and community members who earn this trust through positive interactions with others, and the ecosystem as a whole prospers. Though it has changed the way people rent their homes with its peer-to-peer lodging network, it carries enormous set of challenges around payment, data collection, and global availability.

Whilst Airbnb hosts would like to have a magic wand to appear in top search results, hosts should be earning the top appearance in search results. There are however few specific things host may take care of and the likelihood of appearing in search results will be increased. Few attributes we found can be useful to appear in top search results like responsiveness for booking, booking acceptance, previous bookings, pricing, reviews, instant booking facility etc. Here we found that reviews are more important along with other attributes and reviews create a good reputation in the Airbnb community, also share the various experiences in Airbnb community.

The overwhelming majority of web users act in good faith, but unfortunately there exist bad actors who attempt to gain profit by defrauding websites and their communities. To appear in top search results, bad actors can put fake reviews and can try to misguide the users about search results.

2 Motivation

Fraud detection systems evaluate the transactions and produce a suspicion score (generally a probability between 0 and 1) which shows the possibility of that transaction to be fraudulent. Computational procedures of these scores are relevant to the techniques used to build the model/models in the fraud detection systems. These scores are used with a predefined threshold value to differentiate the fraudulent transactions from the legitimate ones. However, most of the time, these scores are not directly used; but help the observer staff with domain expertise who examine and try to identify the frauds. Because the organizations have limited staff for this process, the ability of the detection systems to produce accurate suspicion scores helps these staff in many ways. Nevertheless, the success of the detection systems lies in distinguishing the fraudulent transactions from legitimate ones through producing suspicion scores with high precisions.

In the case of Airbnb, a sudden 100% increase reservation in a particular city could be a result of a huge event that's going to happen or fraud. This is where anomaly analysis and fraud detection systems comes in. Anomaly detection system that we are trying to build can identify problems in real time as they develop and catch these interruptions as quickly as possible.

- Airbnb hosts can list entire homes/apartments, private or shared rooms. It is about renting out houses and apartments to the tourists frequently

and not to long term residents,

- Airbnb guests may leave a review after their stay, and these can be used as an indicator of Airbnb activity.
- Furthermore an Airbnb host can set up a calendar for their listing so that it is only available for a few days or weeks a year. Listings for the cities are available all year round except for when it is already booked. Entire homes or apartments highly available and rented frequently year-round to tourists, probably don't have the owner present, are illegal. This could be an indication of fraud entry.

Leaving fraud reviews to find in top results, renting entire homes or apartments highly available, owner not present etc. parameters are affecting the city's housing supply, affordability and privacy, security of other legitimate residents around.

This project involves feature extraction like availability, number of reviews, review score ratings and predicting fraud entries. We aim to help users in selecting noble location. We identified top features that lead to legitimate or bad entry which might become a valuable feedback for the guests as well for the that will help them to improve the quality of the services.

3 Structure of the data

The Inside Airbnb data is a publicly available dataset for data mining. The data consists of review data, listings data and the calendar availability in important cities across the world. Some examples of the cities are Amsterdam, Barcelona, Athens, Austin etc. It has been appropriately cleansed , processed and aggregated for visualization and analysis.

All those data can be used to train machine learning algorithm to detect fraud so that Airbnb admin can fix it as soon as the fraud happens in the future. Similarly, fraudulent actors often exhibit repetitive patterns. As we recognize these patterns, we can apply heuristics to predict when they are about to occur again, and help stop them. For complex, evolving fraud vectors, heuristics eventually become too complicated and therefore unwieldy. In this kind of situation, the accuracy is the most important, then comes feature selection, then algorithm, since they don't want to cancel correct reservations and they need to choose whatever algorithm that does the best prediction.

For our project, we are using training data from below cities

```
New York
Los Angeles
San Francisco
Washington DC
Seattle
Chicago
Portland
San Diego
New Orleans
```

Our software is tested against data from below cities,

```
Austin
Boston
Nashville
Oakland
Santa Cruz
```

The listings and the review data under consideration, offer various fields to base our study of fraud detection. Listings data for major cities considered for training has approximately more than 100,000 entries and each entry has 92 or less features .The available attributes from publicly available information are as follows,

```
Id, listing_url, scrape_id, last_scraped, name, summary, space,
description, experiences_offered, neighborhood_overview, notes,
transit, thumbnail_url, medium_url, picture_url, xl_picture_url,
host_id, host_url, host_name, host_since, host_location, host_about
, host_response_time, host_response_rate, host_acceptance_rate,
host_is_superhost, host_thumbnail_url, host_picture_url,
host_neighbourhood, host_listings_count, host_total_listings_count,
host_verifications, host_has_profile_pic, host_identity_verified,
street, neighbourhood, neighbourhood_cleansed,
    neighbourhood_group_cleansed,
city, state, zipcode, market, smart_location, country_code, country,
    latitude,
longitude, is_location_exact, property_type, room_type, accommodates,
bathrooms, bedrooms, beds, bed_type, amenities, square_feet, price,
weekly_price, monthly_price, security_deposit, cleaning_fee,
guests_included, extra_people, minimum_nights, maximum_nights,
calendar_updated, has_availability, availability_30, availability_60,
availability_90, availability_365, calendar_last_scraped,
number_of_reviews, first_review, last_review, review_scores_rating,
review_scores_accuracy, review_scores_cleanliness,
review_scores_checkin, review_scores_communication,
    review_scores_location,
review_scores_value, requires_license, license, jurisdiction_names,
instant_bookable, cancellation_policy,
require_guest_profile_picture, require_guest_phone_verification,
calculated_host_listings_count, reviews_per_month
```

Considering the project's scope, we used few attributes as determining attributes for anomaly analysis. These attributes are as follows,

```
listingId  , availability_30, availability_60, availability_90,
availability_365, number_of_reviews, review_score_ratings,
require_license.
```

These datasets are imbalanced and highly skewed. In general, the genuine entry dominate the fraudulent transactions. The fraudulent events occur rarely. So it is difficult to find the fraudulent entry. If the fraudulent entry is consider as legal then it will cause great loss.

In Airbnb dataset, as labels are not available for deciding authentication of entry, we are deciding a pattern considering features like availability, no of reviews and review score ratings. Also, few other attributes are expected to be considered like requires license, host since etc., but not pivoted as listings

information or reviews. If the apartment is always available with no reviews, we are labelling it as a fraud entry, as it can be potentially looking for renting home or apartment for long time. Also, if apartment has no availability and no reviews, it is being considered as fraud based upon few other verifications like last calendar update, requirement of license. Similarly to detect other suspicious entries, we are setting threshold values for these features. As specified by Airbnb, a host can setup a calendar for their listing so that it is only available for a few days or weeks a year. Entire homes or apartments highly available and rented frequently year-round to tourists, probably don't have the owner present, are illegal. Similarly, if the apartment is not at all available and without any reviews, it could be considered as illegal entry as in Airbnb host couldnt rent an apartment to long term residents.

For classification of fraudulent and non-fraudulent entries, we are using scalable machine learning system is needed to process the large amount of data. The real data is not shared for the number of reasons such as to maintain the privacy of the user. Generally the misclassification cost is high for these detections. Efficient measure should take to reduce the misclassification cost.

4 Methodology

Commonly, Fraud detection problem is considered as a data mining problem. Data mining is the process of discovering meaningful new correlations, patterns and trends by sifting through features of large amount data stored in repositories, using pattern recognition technologies as well as statistical and mathematical technique.

Nowadays, data mining is largely applied to the classification and regression problems. Classification is the process of finding the set of model to differentiate the concept, for the purpose of being able to use the model to predict the class of object whose label is unknown.
Our project included two steps.

- Extract desired features from data
- Apply models on the extracted features

Most of the development of project has been done on locally available Hadoop and Python setup in labs. We are also using Amazon Web Services (AWS) in next step of this project. The data has been downloaded and extracted from the Inside Airbnb data website. It is deployed on AWS. The MapReduce jobs employed is sorting the data by reviews and availability features.

The main focus of the fraud detection software is to train a classifier to distinguish between the malicious and the authentic entries in the dataset. We are training a SVM to classify the data into two specific groups. The evaluation measures and the results are displayed by the system.

Airbnb data set we are using, is a quite large dataset. The data consists of review data, listings data is available in csv (Comma separator vector) format

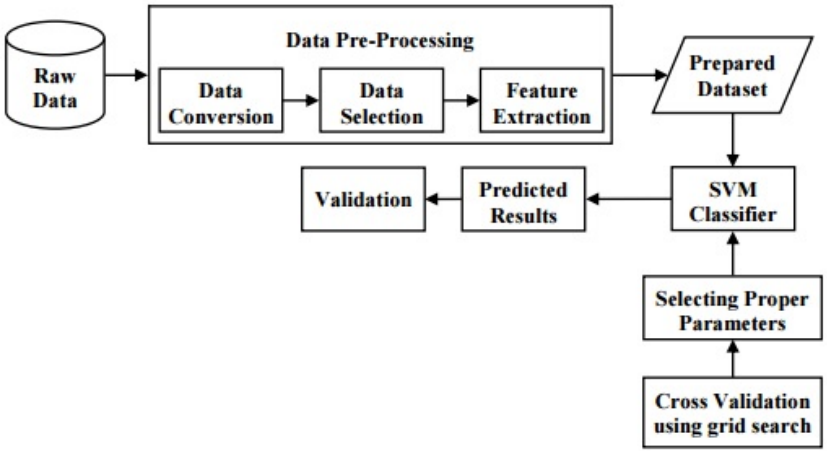

Figure 1: Block Diagram of working of the system

which is considered as raw data. First of all, the collected data is pre-processed before starting the modelling phase.

4.1 Data Pre-processing

Feature selection is a very important stage in fraud detection. Here input data is transformed into a reduced representation set of features. In this phase, features which are only able to classify are considered. Adding irrelevant features may make the classifier inefficient.

As first phase of data pre-processing, we have transformed csv files in XML format. The csv files of listings we had a lot of missing fields. That made MapReduce processing of these files cumbersome. XML converted data is easy to read in MapReduce jobs.

For feature selection, we are using Hadoop platform. Hadoop is ideal for distributed processing for such large scale data-sets on clusters of commodity hardware. This biased us towards using Hadoop for processing our data-sets. It also opens up the option for scaling up the system. XML file of data is processed using MapReduce software and important features are extracted.These features are shown in following table.

```
availability_30
availability_60
availability_90
availability_365
number_of_reviews
review_score_rating
```

4.2 Machine Learning

After extraction, features are classified using support machine vector (SVM). Here, only the behaviour features are selected for training.

Support vector machines (SVMs) are supervised learning models that analyse labelled data used for classification. When the data is not labelled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The classification work usually involves separating data into training and testing sets. Each time the training set contains one "target value" (the class labels) and several "attributes" (the features or observed variables). The SVM model classifies the test data based on the data attributes.

SVM finds a linear separating hyper plane with the maximal margin in this higher dimensional space. In this project, supervised learning approach for SVM is used for fraud detection to classify the data into authentic and fraud listings.

In order to implement SVM, we used the Python libraries, as working with machine learning is easier in Python than it is in Java. Using the scikit-learn module available for machine learning, we trained the data using the svm module in it. As the training of the data requires a lot of time, we optimized the processing time by writing the classifier obtained into a Pickle file (.pkl extension, using cPickle library) and using it for every test data.

4.3 System and Services

For development purpose, we have deployed our software on local Hadoop and Python setup. Hadoop setup involves systems from CS120 lab. Scikit-learn is used for clustering, classification algorithms in machine learning.

We are also deploying our software on Amazon EMR in order to reduce the dependency of MapReduce and Python Scikit-learn with each other. Our AWS EMR cluster setup includes *Hadoop* and Spark for now. Initial MapReduce tasks will be handled by Hadoop and the resulting file from MapReduce will be trained and tested for classification using Spark libraries.

To prepare the cluster to execute python programs, we have to install below packages,

```
1    sudo yum install python-numpy python-scipy -y
```

Other steps are nothing but copying necessary jar and data files to S3 bucket and creating custom jar step for MapReduce, and two steps for train and test Spark Applications.

5 Results

5.1 Evaluation Metrics

In order to quantify the results of the classification model to detect fraudulent listings, we use the Root Mean Square Error (RMSE) as the evaluation measure.

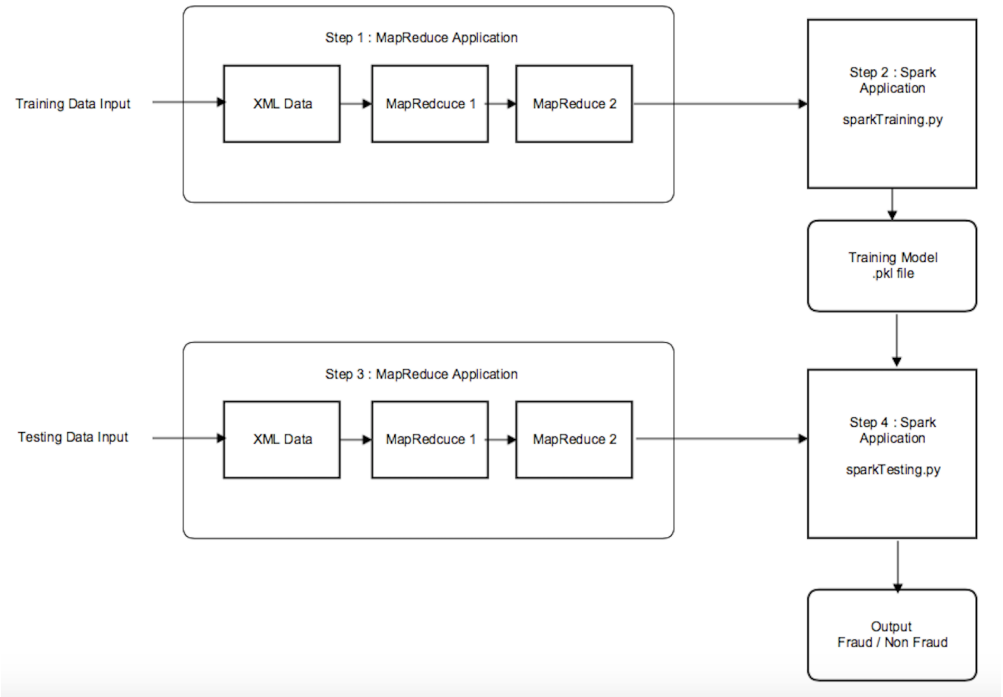

Figure 2: Modular execution with Amazon EMR

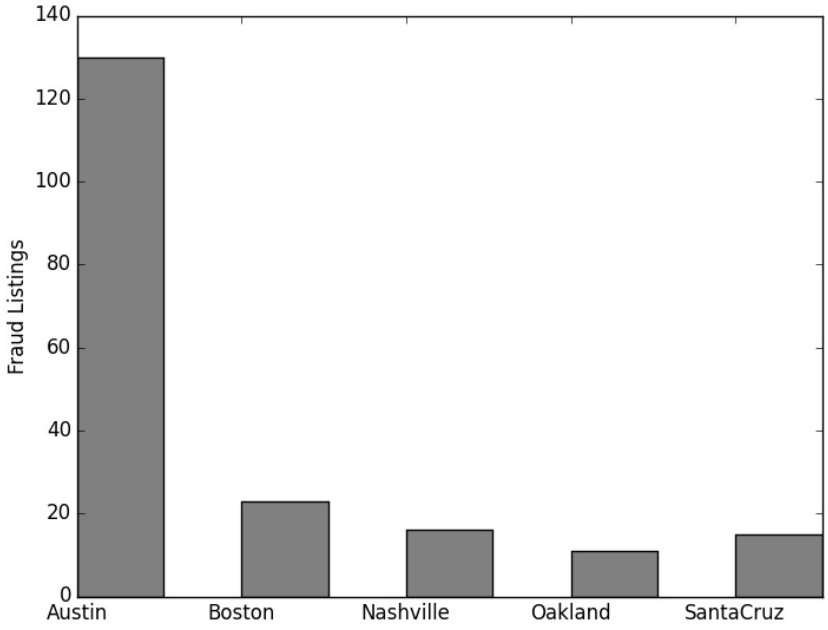

Figure 3: Graph of Cities vs Fraud Listings

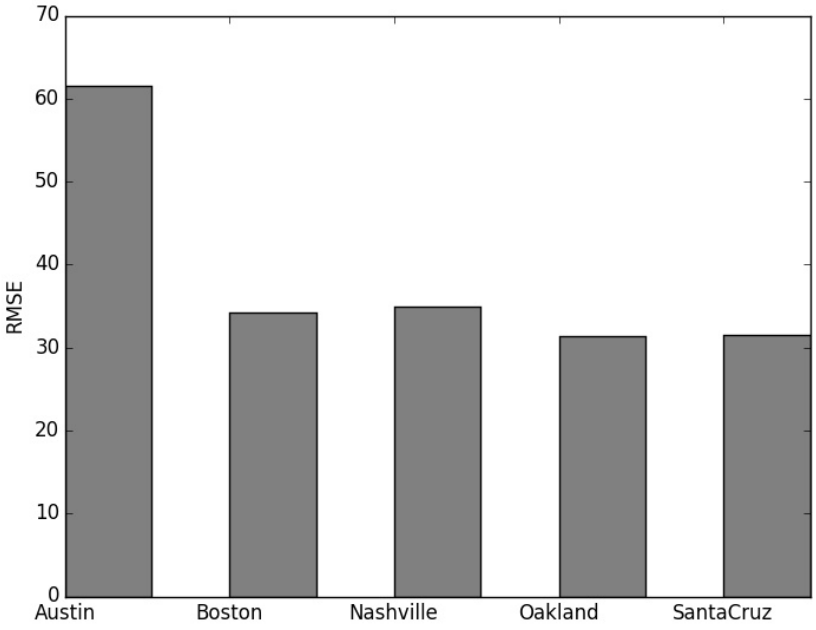

Figure 4: Graph of Cities vs RMSE values

RMSE, given as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - h(x_i))^2}{N}} \tag{1}$$

where y_i is the label of the ith data entry that is available, while h(x_i) is the result of hypothesis applied on the corresponding test data entry. We also calculated the number of fraud entries and the authentic listings in each of the test data. For cities with lesser listings, we observed lesser RMSE value, thereby having more authentic listings.

5.2 Observations and Graphs

Figure 3 and 4 show the plot of the cities with the number of fraud listings as well as the RMSE values. While Austin has the highest RMSE of 61%, it is because of the large number of listings in the data, we see more fraudulent entries. Consequently, Oakland has the least fraudulent entries, and the least RMSE value of 31.417%.

No	City Name	Total Listings	Fraud Entries	RMSE
1	Austin	5835	130	61.500
2	Boston	2558	23	34.189
3	Nashville	2110	16	34.968
4	Oakland	1155	11	31.417
5	SantaCruz	814	15	31.545

5.3 Challenges in Project

While defining problem and objective for solid direction towards analytics project, we need to explore and understand the data available. This becomes critical when you have multiple attributes available. We faced below challenges while playing around with the data,

- Data Pruning (removing the unnecessary and irrelevant data fields in the data).
- Data Inconsistency, handling special characters like smiley face, cat face etc. in reviews etc.
- Treating missing values or *NaN*, Infinity or a value too large for dtype.
- Detecting outliers.

Modeling (training) of data is highly time consuming process as data grows. While applying grid search to optimize the parameters for enhancement of performance, training of data is again time consuming. So tweaking parameters becomes extremely cumbersome.

5.4 Turnaround and performance time

Data pre-processing in Hadoop took very less time (roughly 5 minutes). Given the massive amount of data, the training time of the SVM in python was huge. By writing the classifier to a file and accessing it, it significantly reduced the processing time. The table given below explains the time required for the execution of the system.

No	Executing Module	Time (in minutes)
1	Data pre-processing in Hadoop	5
2	Training and testing (without writing to file)	15
3	Training of the data (writing to file)	12
4	Testing of data	0.5

These values are obtained upon executing the system on the machines in CS 120 lab. Values will vary for the amount of training testing data, as well as the processing power of the machine.

6 References

1. Airbnb http://www.airbnb.com
2. Airbnb data set http://insideairbnb.com/get-the-data.html
3. Airbnb Case Study https://aws.amazon.com/solutions/case-studies/airbnb/ Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." ACM computing surveys (CSUR) 41.3 (2009): 15.
4. Šubelj, Lovro, Štefan Furlan, and Marko Bajec. "An expert system for detecting automobile insurance fraud using social network analysis." Expert Systems with Applications 38.1 (2011): 1039-1052.
5. Suthaharan, Shan. "Big data classification: Problems and challenges in network intrusion prediction with machine learning." ACM SIGMETRICS Performance Evaluation Review 41.4 (2014): 70-73.
6. Singh, Kamaldeep, et al. "Big data analytics framework for peer-to-peer botnet detection using random forests." Information Sciences 278 (2014): 488-497.
7. Miller, Benjamin A., Nicholas Arcolano, and Nadya T. Bliss. "Efficient anomaly detection in dynamic, attributed graphs: Emerging phenomena and big data." Intelligence and Security Informatics (ISI), 2013 IEEE International Conference on. IEEE, 2013.
8. Nassem Hakim, 'Architecting Machine Learning System for Risk ', 2014. [Online]. Available: http://nerds.airbnb.com/architecting-machine-learning-system-risk. [Accessed: 16- Jun- 2014].

9. Timothy Prickett Morgan, 'Airbnb shares a keys to its Infrastructure ', 2015 [Online]. Available: http://www.nextplatform.com/2015/09/10/airbnb-shares-the-keys-to-its-infrastructure/. [Accessed: 16- September- 2015].

10. Dheepa, V., and R. Dhanapal. "Behavior based credit card fraud detection using support vector machines." ICTACT Journal on Soft Computing 4.4 (2012): 391-7.

11. https://en.wikipedia.org/wiki/Root-mean-square deviation

12. New - Apache Spark on Amazon EMR https://aws.amazon.com/blogs/aws/new-apache-spark-on-amazon-emr/

13. Anomaly Detection Using PySpark, Hive, and Hue on Amazon EMR http://blogs.aws.amazon.com/bigdata/post/Tx2642DKK75JBP8/Anomaly-Detection-Using-PySpark-Hive-and-Hue-on-Amazon-EMR