CMPS 140 Progress Report: How to Become Top Earner in Airbnb?

Tangni Wang, Tung Hoi Man, Yunxiang Fu
{twang63, tuman, yfu7}@ucsc.edu

1 Introduction

So far we studied the scikit-learn tutorial about boston housing price. We also follow instructions and documentation on scikit-learn website to load external datasets, use pandas to parse our csv files. We also implemented Baseline model with the sklearn.dummy class.(Reference 2)

1.1 Goal and impact

Our goal of the project stays the same as the proposal that we will be using the datasets from the Airbnb Seattle case to compare different performers' data especially those ones with more bookings and high rating in a comparison with the ones with less bookings and low rating. The goal is to make recommendations for the low earner hosts and make their listings more profitable and gain more revenue than before.

1.2 Related Work

Many comparisons have been done between high earners and low earners (see references). We are inspired by those work. We want to see if there are other interesting factors or combination of factors that would affect booking rate. The features that we will be analyzing is based on the top ranked correlation of score_review. By having the monthly_reviews and the review_rating, we are multiplying them and divide by 100 to know approximately which owners have more completed orders or better reviews than others. By looking at the score reviews of top earner we will be able to identify what are the factors to be a top performer and what make the low performers.

2 Data

We have three set of data including calendar.csv, listings.csv, and reviews.csv. Calendar has attributes: listing_id, the price and availability for that day. The review has listing_id, id, date, reviewer_id, reviewer_name, and comments. Listings have the full descriptions of each post. Since it has a lot of attributes. We will extract some useful attribute to build our model like price, number_of_reviews, review scores. (See reference 1)

Calendar.csv: 1048576 rows
Listings.csv: 3819 rows
Reviews.csv: 84850 rows
We will have divide the dataset into Training set and Test set
Training set: 70%
Test set: 30%

3 Methodology

3.1 Features

We are using library Panda and BeautifulSoup to pull our data from the given csv files from the Airbnb Seattle case includes calendar.csv, listing.csv and reviews.csv. By taking some features from these input data files, we can take some attributes from it such as price, review_scores_rating and some others in order to analyze the data and have our result prediction as the output. We will be using these libraries to gather data and clean them based on our needs, so it will be much easier for us to analyze through the algorithm. By taking the analyzed data as our output we will have our results to reach the goal. (Reference)

3.2 Baselines

We are using DummyRegressor from sklearn library to make our baseline model. DummyRegressor implements several simple strategies for Regression. Stratified generates random predictions by respecting the training set class distribution. Most_frequent predicts the more frequent label in the training set. Prior always predicts the class that maximizes the class prior. Uniform generates predictions uniformly at random. We are experimenting these strategies to make the baseline model and then we will approach to make some better model to outperform our baseline model.

4 Evaluation

We are using one or more of the following three methods to evaluate the success of our algorithm and results:

-Performance measure

We will use this method to measure the performance of our result. Once we have our result finalized, we will test its performance by taking the predictions we made and compare with the results. By going through the features that imply how to be a good Airbnb earner and the habits of a low Airbnb earner, we will check the accuracy by taking the percentage of total correction divided by the total predictions we made to test the performance.

-Test and train datasets

At the end of the project, we will have a test set and a training set to evaluate our result. The test set will be a smaller amount of dataset being evaluated in our algorithm that reaches our goal as an expected value of our data and it should be very likely to prove our goal. The training set will be the dataset we have to indicate the performance of the model.

-Cross Validation

In this method, we will be using the entire dataset to train and test our algorithm. We will first separate the dataset into a number of equally sized groups of instances, then we train on all folds exception one that was left out and we test the model on the left part. Then we repeat the process so each fold gets a chance to be left out. After the steps, we test the performance on the average data with the algorithm.

5 Updated Milestones

-Progress and results achieved (Reference Appendix: Tables and Images)
So far by the beginning of this report, we have done the second and third task. In the second task: collect the dataset, begin to implement the baseline models, we have taken the dataset from the dataset files calendar.csv, listing.csv and reviews.csv and parsed the data with the attributes of price, review_score_rating and some other ones as output to a csv file. Except the data we need, we also imported some distinct data attributes such as *id* so that we can categorize them to analyze the differences among reviews, prices and other features to analyze how they affect our goal but still have the record of the unique Airbnb earners that's being compares with the copy of its own dataset.

With the implementation on the baseline model, we are comparing the result of top earner and lower earner with several categories such as extra_people, availability, host_has_profile_pic, bathrooms and more. We are going to look deeply into the rate and do more comparison.

-Quality of the result
The result we have in the appendix shows a basic collection of one of the data we worked with. Since the input dataset is huge, we only used parts of it to test. The data was parsed correctly but we have some special characters in the dataset that still needs to be picked and separate out from the data collection. We need to keep narrowing down the dataset with feature implemented so it will be easier for us to get more accurate calculation results. The dataset we use is only from Seattle but it has a big variety of data so the result is somewhat reasonable. We need further implementation to check the result with a graph that is easier to test.

-Presentation of the result
We use pandas dataFrame class to present our data in a nice looking form.

date	task	progress
01/23/19	Submit a project proposal. (1 page)	Done
02/06/19	Collect the dataset, begin to implement the baseline models.	Done (100%)
02/15/19	Complete the baselines & submit a progress report (3 - 4 pages).	In Progress (70%)
02/20/19	implement the stretch models.	Not Start (0%)
02/27/19	Complete the stretch models & begin the evaluation.	Not Start (0%)
03/06/19	Finish the evaluation & work on the poster and the final report.	Not Start (0%)
03/14/19	Poster presentation.	Not Start (0%)
03/17/19	Submit final project report (5 - 7 pages).	Not Start (0%)

6 References

1. Resource data - Seattle Airbnb Open Data: https://www.kaggle.com/airbnb/seattle
2. Baseline model Resource code - scikit-learn:https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyRegressor.html
3. Regression Resource code - scikit-learn: https://scikit-learn.org/stable/modules/linear_model.html
4. Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
5. Panda toolkit: https://pandas.pydata.org/pandas-docs/stable/
6. Beautiful Soup 4.4.0 documentation: https://www.crummy.com/software/BeautifulSoup/bs4/doc/
7. Evaluate algorithm: https://machinelearningmastery.com/how-to-evaluate-machine-learning-algorithms/

Appendix: Tables and Images

An example of using pandas to parse and present csv data.

```
Seattle_file_path = '/Users/I/Downloads/Seattle/listings.csv'
Seattle_data = pd.read_csv(Seattle_file_path)
Seattle_data.describe()
```

	id	scrape_id	host_id	host_listings_count	host_total_listings_count	latitude	longitude	accommodates	bathrooms	bedr
count	3.818000e+03	3.818000e+03	3.818000e+03	3816.000000	3816.000000	3818.000000	3818.000000	3818.000000	3802.000000	3812.0
mean	5.550111e+06	2.016010e+13	1.578556e+07	7.157757	7.157757	47.628961	-122.333103	3.349398	1.259469	1.3
std	2.962660e+06	0.000000e+00	1.458382e+07	28.628149	28.628149	0.043052	0.031745	1.977599	0.590369	0.8
min	3.335000e+03	2.016010e+13	4.193000e+03	1.000000	1.000000	47.505088	-122.417219	1.000000	0.000000	0.0
25%	3.258256e+06	2.016010e+13	3.275204e+06	1.000000	1.000000	47.609418	-122.354320	2.000000	1.000000	1.0
50%	6.118244e+06	2.016010e+13	1.055814e+07	1.000000	1.000000	47.623601	-122.328874	3.000000	1.000000	1.0
75%	8.035127e+06	2.016010e+13	2.590309e+07	3.000000	3.000000	47.662694	-122.310800	4.000000	1.000000	2.0
max	1.034016e+07	2.016010e+13	5.320861e+07	502.000000	502.000000	47.733358	-122.240607	16.000000	8.000000	7.0

8 rows × 30 columns

```
[ ]
import pandas as pd
import io

df = pd.read_csv(io.StringIO(uploaded['cal_test.csv'].decode('utf-8')))
df
```

	listing_id	date	available	price
0	241032	1/4/16	t	$85.00
1	241032	1/5/16	t	$85.00
2	241032	1/6/16	f	NaN
3	241032	1/7/16	f	NaN
4	241032	1/8/16	f	NaN
5	241032	1/9/16	f	NaN
6	241032	1/10/16	f	NaN
7	241032	1/11/16	f	NaN
8	241032	1/12/16	f	NaN
9	241032	1/13/16	t	$85.00
10	241032	1/14/16	t	$85.00