Is your house suitable for Airbnb?

CIS 520: Machine Learning

Author
Yuan Li
Xuanyi Zhao

Project Mentor
Haoxian Chen

April 30, 2019

Team Member	Contributions
Xuanyi Zhao xuanyizh@seas.upenn.edu	Solution approach, Data cleaning, Data stitching, Machine Learning Algorithms and analysis, Report
Yuan Li yliii@seas.upenn.edu	Problem formulation, Machine Learning Algorithms and analysis, Plot, Report

Abstract

In this study, we use supervised machine learning algorithms to predict price of Airbnb house listings. Our dataset contains relevant features, price and tier that each house belongs to. The objective of our study has three parts: predicting the price of a house, predicting the tier of the house, and check the accuracy of models when applied to different regions. We used several supervised machine learning algorithms and were able to predict the price with a 71% R^2 and the tier with 84% mean absolute deviation. At the end, we compiled a map to demonstrate how well our models fit in different areas. This report summarizes the data, methods, algorithms, model comparison and results.

1 Introduction

Airbnb operates a global online marketplace and hospitality service accessible via its websites and mobile apps, generating over $2 billion every year [6]. Usually, pricing of the houses is determined solely by hosts themselves. Yet we believe prices can be predicted with features, for instance, location, amenities and prices of houses close by. We use different supervised learning algorithms and models, including K-Nearest Neighbors, Support Vector Machine and Random Forest. What's more, we divide the dataset into multiple subsets, each one corresponding to a major business district in Greater Los Angeles area. With results from our algorithms, we generate a map indicating the accuracy of our models when applied to different areas of LA. The trends and causes we identified can be useful to customers looking for hotels and companies/individuals providing hospitality service.

2 Related Work

There are multiple tools available online that provide prediction on airbnb pricing, for instance Airbnb Pricing Prediction[1]. They use various regression models, include Linear Regression, Ridge Regression, and Lasso Regression. However, their models is a poor fit, indicating a R^2 around 0.3 on test dataset.

With different selected features, models and algorithms, we believe that a much higher accuracy can be achieved. The novelty of our work lies in the following three aspects:
1. Use prices of houses nearby (within 2 miles) as a feature in models;
2. Divide houses into five tiers according to prices and make prediction on which tier a house belongs to;
3. Check the accuracy of our models in different areas of LA and speculate about possible reasons for the difference in accuracies.

With these new information obtained from our models, hosts can make decision on how to decorate their places and how to advertise them on Airbnb website to attract potential guests. Guests can use the models to check the price range according to their needs and if they are getting the best value out of their money.

3 Problem Formulation

To generate a more efficient model, we need to look for intrinsic factors that contribute to pricing. Features, including location, amenities, rating, number of reviews, access to free parking and fitness center, are of importance when customers choose accomodation, and thus largely effect the pricing[2][5]. Amenity in our dataset contains TV, internet, parking and was originally seperated from the main dataset. We added each one of them as a feature to the main dataset. Other factors provided in the dataset were examined. We came to the conclusion that they don't significantly effect pricing and thus were ignored.

Based on Dan Wang and Juan L Nicolau's research on price determinants of accommodation rental, prices of housing options nearby is also driving force. For each house lising, we use the latitute and longtitue to find the prices of houses in a radius of 2 miles and get the mean price. The calculated mean is used an another feature in our models.

Originally, the price in our dataset is a column of numbers. We run several models and algorithms to make predictions on prices. We also added another column to categorize houses by price: economy($1-$150), midscale($151-$300), upscale($301-$450), luxury($451-$600). Each tier is represented by a number from 0 to 3 accordingly. Then with the features, we make prediction on the tier that houses belong to.

4 Data Preprocessing

We use Airbnb Los Angeles dataset which is sourced from publicly available information from the Airbnb site. There are 94 original features in our selected dataset. After dropping the irrelevant features, such as url, name and other descriptive information, we select the following set of features, which were agreed upon based on their influence towards the price of house.

After removing the NaN value and changing the inappropriate data type, we complete one-hot-encoding to choose the popular amenities as our features above, which are available for at least 10000 instances. In addition, considering the imbalance situation and according to the distribution plot of the target variable, we select the price per night no more than \$600 (99%). Also, given the common situation in realty that there are some houses whose review number is extremely small but have a very high review score and inappropriate price, we decide to remove these instance (review number less than 5) because of their lack of authority. Then we calculate the average housing price within 2 miles of each example.

Table 1: A tabular view of the selected features. The target value price is not included.

location	amenities	accommodates
3	18	12
longitude, latitude, exact location, nearby price	wifi, TV, kitchen, air condition, heating, washer, Shampoo, etc.	bedrooms(#), reviews(#), bathrooms(#), room type, etc.

After that, based on common sense and knowledge from some hotel and booking app, we split the whole price range of our data into 4 different intervals that have the same size. These four price intervals can represent four house tiers, such as economy, mid-scale, up-scale and luxury. As discussed eariler, we use the original price data for regression target and the labeled price data for multiclassification use.

5 Methods

We use different methods to learn reasonable models for our dataset. The regression purpose usually requires us to minimize mean squared residuals between the observed responses in the dataset and the responses predicted by our model. Also, the support vector regression method calls for minimizing mean absolute deviation between the true price and the prediction on the basis of an insensitive ϵ gap. We utilize R^2, mean squared error and mean absolute error to measure our regression performance.

On the other hand, the multiclassification assignment is often measured via 0-1 loss. In order to consider the different penalties for different false results of the house tiers prediction, we also use the mean absolute difference between the true labels and predicted labels to measure our classification performance. Although the latter accuracy might be lower than the accuracy of 0-1 loss, we think that it explains the misclassification better. What's more, we also use multiclass geometric mean performance measure, such as geometric mean of each class's recall, precision and F_1 measure to help to explain our model.

6 Algorithms

All the supervised machine learning algorithms we use is implemented via the Scikit-learn package [4], which is commonly used in Python language. We also use Python 3 to complete all data wrangling and plotting work. For the last part of our project, we use Tableau to develop a scatter plot of latitude and longitude in Los Angeles.

For the regression analysis of our project, we use several algorithms: linear regression, ridge regression and kernel support vector regression. For the classification purpose, we also apply various methods: linear logistic regression with L2 penalties, kernel support vector classification, k-nearest neighbors and random forest. The specific discussion of each algorithms are shown below.

6.1 Regression

6.1.1 Linear Least Squares Regression

Ordinary Least Squares Regression is aimed to find the minimized sum of residual squares between the predicted values and correct values. We use it to minimize the sum of residual squares between true price and predicted price.

6.1.2 Ridge Regression

Because the number of features is not small, we choose to utilize L_2 regularization on the size of coefficients of least squares regression to avoid overfitting, resulting in the Ridge Regression. Mathematically, it solves the problem by minimizing the L_2 regularized mean squared loss

$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}^\top \mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{w}\|_2^2$$

where λ is the complexity parameter that controls the amount of shrinkage.

6.1.3 Kernel SVR

Usually the price of a house in Airbnb should have some fluctuation within a year. Therefore, we use Support Vector Regression algorithm to allow a $\epsilon-$insensitive loss for the regression process. Specifically, we minimize

$$\frac{1}{m}\sum_{i=1}^{m}(\mid \mathbf{w}^\top \phi(\mathbf{x}_i) - y_i \mid -\epsilon)_+ + \lambda \|\mathbf{w}\|_2^2$$

where $\phi(\mathbf{x})$ is the kernel function. We use the radial basis function (RBF) kernel and the polynomial kernel to fit our model.

6.2 Multiclassification

6.2.1 Linear Logistic Regression

Logistic Regression is a commonly useful classification method. Multiclassification problem use the softmax function to derive the most possible class for each example. We try the logistic regression with L_2 regularizer in order to avoid overfitting. To be specfic, we do this by minimizing the logistic loss

$$\ell_{log}(y, f) = log_2(1 + e^{-yf})$$

In addition, we use the solver 'newton-cg' provided by Scikit-learn package which might need longer time but is robust to our unscaled dataset.

6.2.2 Kernel SVM

Our data is non-linearly separable, so we use the soft margin Support Vector Machines for classification. In addition, one-vs-rest strategy is often applied to tackle multiclassification problem. For each binary problem in the one-vs-rest situation, we utilize the method by minimizing the following L_2 regularized hinge loss

$$\frac{1}{m}\sum_{i=1}^{m}(1 - y_i(\mathbf{w}^\top \mathbf{x}_i))_+ + \lambda \|\mathbf{w}\|_2^2$$

6.2.3 K-Nearest Neighbors

Houses with similar features are believed to have similar prices. k-NN algorithm stores the training set. For every test point, k-NN finds the k nearest points in the training set and predicts by averaging their labels.

6.2.4 Random Forest

Since we are classifying each house listing, it is intuitive to use classification trees. Tree models are suitable to multiclass prediction. Compared to other algorithm for instance SVM, Random Forest scales nicely and works better with a mixture of numerical and categorical features. We can use the original data without pre-processing. In fact, we can see from the performance measure of our models for multiclassification, Random Forest fits better than SVM. Furthermore, one big problem of machine learning is overfitting. Results from cross validation of SVM shows that the model is overfitted. Yet for random forest, the possibility of overfitting can be largely reduced as long as there are enough trees in the forest. On the other hand, having a large number of trees may make the algorithm slow. With our dataset and chosen number of trees 60, it is unlikely to have a significant negative impact on the speed.

Following is an image illustrating the structure of Random Forest for multiclassification [3]:

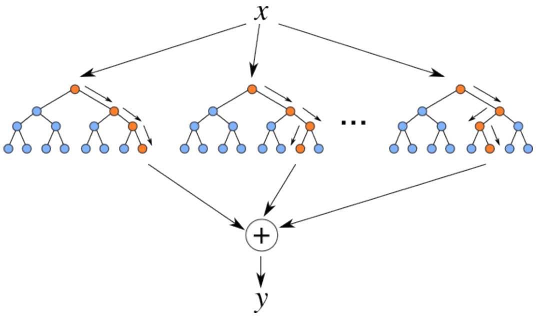

Figure 1: An example forest

7 Model Results and Comparison

We split our whole processed data into 80% training set and 20% test set. The whole prediction is divided into two parts. The first one is for the regression analysis on the house price prediction, the second one is for the multiclass classification on the house price tiers prediction.

7.1 Price Prediction: Regression

After using OLS regression, ridge regression and kernel SVR methods, we can conclude that the most accurate model we used is SVR with polynomial kernel, for which we use degree 2. The best R^2 for test data achieved by it is 71%. In addition, because of the fact that SVR algorithms minimize the mean absolute deviation between true price and predicted price, the two SVR algorithms perform better on MAE measure. The polynomial kernel SVR performs well but it needs the longest training time, because we use the 'newton-cg' solver which is robust to our unscaled dataset but slower for large datasets.

Table 2: Comparison of regression models on predicting Airbnb house price. We use R^2, mean squared error(MSE) and mean absolute error(MAE) to measure performance.

Model	Type	R^2	MSE	MAE
OLS Regression	Training	0.6858	2732.55	34.97
	Test	0.6827	2627.23	34.14
Ridge Regression	Training	0.6855	2735.84	35.00
	Test	0.6845	2611.92	34.04
SVR (RBF Kernel)	Training	0.6651	2912.76	32.02
	Test	0.6633	2787.60	32.37
SVR (Poly Kernel)	Training	0.7015	2596.64	30.94
	Test	0.7118	2385.52	29.81

7.2 Price Tiers Prediction: Multiclass Classification

We apply 5-fold cross validation to choose the hyperparameters for each methods. Each choose the highest cross validation accuracy. We display a few results below.

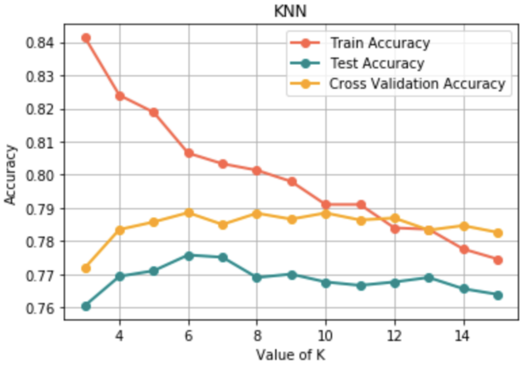

Figure 2: K-Nearest Neighbors for Multiclass Classification

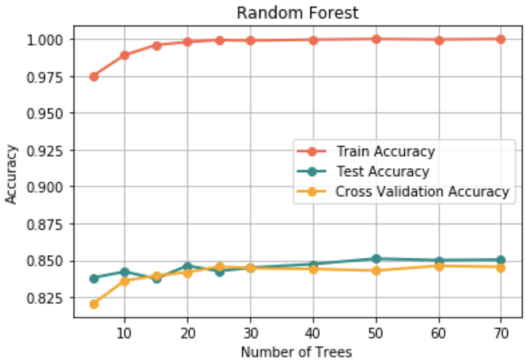

Figure 3: Random Forest for Multiclass Classification

The prediction results show that the random forest with 60 trees gives the highest test accuracy 85%, which means that most of our test data have been predicted correctly.

What's more, in order to consider different results for different misclassification, we reevaluate the accuracy by including the cost sensitive factor. Specifically, we use the mean absolute deviation between the true price tiers and predicted price tiers for each test instance, which might cause a small decrease in accuracy compared to the one using 0-1 loss measurement but we think this penalty explains the case for multiclass classification better.

Table 3: Comparison of multiclass classification models on predicting Airbnb house price tiers. The performance of test is measured by the 0-1 loss and the mean absolute deviation (MAD) between the true tiers and predicted tiers.

Algorithms	Training Accuracy	Test Accuracy	MAD
LR	0.8472	0.8457	0.8358
SVM (RBF)	0.8872	0.8375	0.8212
SVM (Poly)	0.8526	0.8429	0.8301
KNN (6)	0.8307	0.7988	0.7758
RF (60 trees)	1.0000	0.8511	0.8368

In addition, we also calculate some complex performance measures for our five models. Due to the case of multi-label and the imbalance settings of our test data, we use multiclass geometric mean measurement to calculate the results from each binary case. We summarize the results of precision, recall and F_1 score in the table below. It can be found that though polynomial kernel SVM method produces a good accuracy but it performs badly on precision and recall. Also, despite the accuracy or the mean absolute deviation of logistic regression is approximately the same as random forest, the latter method gives a higher precision(65%) and recall(51%) than the former(precision 60%, recall 46%). Therefore, among all the algorithms we used for predicting the Airbnb house tiers, random forest is the best one.

Table 4: Comparison of models with the complex performance measures.

Algorithms	Precision	Recall	F_1 Score
LR	0.6028	0.4580	0.5206
SVM(RBF)	0.5176	0.3843	0.4411
SVM(Poly)	0.5943	0.2894	0.3892
KNN (6)	0.5297	0.3809	0.4432
RF (60 trees)	0.6498	0.5127	0.5732

After analyzing the model results, we split our test data into 7 different groups based on their location (latitude, longitude) in Los Angeles and then recompute the test accuracy of each group. According to the results on different groups, in the future analysis of this topic, the different suitable advice or enhancement can be applied to each sub-model for different area of Los Angeles.

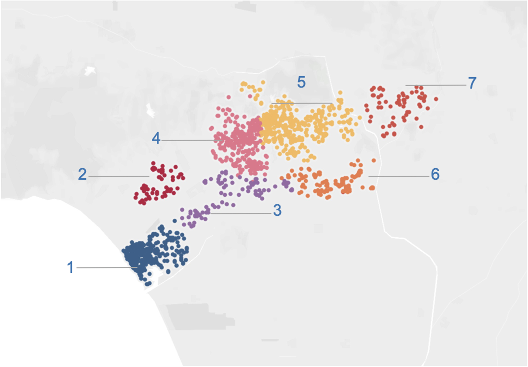

Figure 4: This is a map plotting the test data on the LA area. We split them into 7 different groups.

Table 5: The name and the test accuracy of each area.

	District	Test Accuracy
1	Santa Monica	0.7486
2	Beverly Hills	0.8666
3	Mid City	0.8851
4	West Hollywood	0.8148
5	East Hollywood	0.8771
6	Downtown LA	0.8392
7	Pasadena	0.9194

8 Conclusion

The accuracy we achieve with our models is reasonable, given the noise in the dataset. It is also higher than other similar prediction tools[1]. We are able to make detailed and relatively accurate prediction on the price and tier of Airbnb houses. Therefore, we can conclude that we have achieved the first two parts of our objective: predict price and tier for Airbnb houses.

The last part of our objective is to see if our models work equally well in different regions. We split the dataset into 7 groups, each one corresponding to a major area in Los Angeles. From the analysis and results above, we decided to run Random Forest to predict tier for houses in each group. The picked number of trees is 60, based on the result of cross validation. Above is the generated map and a table showing the accuracy of our model in 7 regions. It is clear that our model fits better in some areas than it does in others. For exmple, our model has the lowest accuracy in Santa Monica, which is a city on the coastline. With its unique location, we believe other aspects of houses that are not covered by our dataset are considered by people when they look for places to stay in Santa Monica and decide how much they are willing to pay. We will have further discussion in next section. Figure 4 is the map and Table 5 shows the area and accuracy.

9 Recommendations

As mentioned briefly above, we can conclude that people look for different aspects of a house when location varies. Further refinement of dataset could lead to improvement in performance measure of models. According to location, dataset of house listing can be broke into several subsets. For each subset, take into account features that are specific to the corresponding region. Some suggested features include: distance to tourist attractions, number of restaurants near the house, average sunny days per month, etc[5]. These suggested features do not correlate with most existing features in our dataset since our dataset mainly contains location, review, price and various amenities.

Acknowledgments

We would like to acknowledge Prof. Shivani for teaching this fabulous course; Haoxian for plenty of assistance and useful suggestions on our project.

References

[1] Airbnb Pricing Predictions. Airbnb pricing predictions, 2019. [Online; accessed 1-April-2019].

[2] Paridhi Choudhary, Aniket Jain, and Rahul Baijal. Unravelling airbnb predicting price for new listing. *arXiv preprint arXiv:1805.12101*, 2018.

[3] Harp. Harp random forests, 2019. [Online; accessed 1-April-2019].

[4] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[5] Dan Wang and Juan L Nicolau. Price determinants of sharing economy based accommodation rental: A study of listings from 33 cities on airbnb. com. *International Journal of Hospitality Management*, 62:120–131, 2017.

[6] Wikipedia contributors. Airbnb — Wikipedia, the free encyclopedia, 2019. [Online; accessed 1-April-2019].