Title: Rental Price Prediction of Airbnb In New York City

Authors: Xintong Li, Ziyue Dong, Jiyang Ge

1. Introduction and Motivation

Airbnb is a platform where millions of hosts and travelers list their space and book unique accommodations anywhere in the world.In a modern, busy cosmopolitan like NYC, there are so many competitions for housing and huge demands for accommodations, airbnb pricing is important to get right to help hosts achieve optical benefits. Therefore the goal of our project is to come up with the appropriate prediction of Airbnb price using machine learning.

2. Describe the data

The data was downloaded from the Inside Airbnb Dataset 'New York City Airbnb Open Data' (link: http://insideairbnb.com/new-york-city/).Specifically, this dataset describes the listing activity and metrics in NYC, NY in 2019. The columns include the following information: id number of the housing, name of the host, id number of the host, the borough, the neighbourhood, latitude and longitude, type of room, price per night, the minimum nights allowed to book, the number of reviews, the date of last review, average reviews per month, the number of host listings, and the available days in 365 days.

2.1 Data Exploration and Data Cleaning

We took a deeper look at our data and found out that airbnbs in Manhattan are generally pricier than the ones located in other boroughs. To prepare the data for modeling, we filled missing values, created dummy variables to transform categorical data into numerical and normalized the features. Then, we randomly splitted the data into 80% training and 20% testing. Here is an instance of our sample:

```
        latitude  longitude  entrie_home  ...  Manhattan    Queens  Staten Island
28317  -0.67356   -0.42032     0.961415  ...  -0.891833 -0.362035      -0.087677

[1 rows x 13 columns]
28317    300
Name: price, dtype: int64
```

3. Methodology

3.1.1 Linear Regression and Ridge Regression

Since our target variable, rental price, is a continuous variable, we consider this problem as a supervised regression problem instead of classification. Thus, the baseline model is the default linear regression. However, a clearly linear regression is always too oversimplified for reality problem because it assumes the covariates and response variables to have linear relationship which may not suit our problem. To improve this, we also use another linear model , the Ridge Regression Model with sklearn.linear_model.Ridge. This solves a regression model where the loss function is the linear least squares function and regularization is given by the l2-norm. To find the optimal parameters and avoid overfitting, we also compare the performance of model with alphas=[0.001,0.01,0.1,1] as well as cross validation by RidgeCV.

3.1.2 Random Forest

Random forest is also a common ensemble learning method for regression. It runs decision trees in parallel which means there is no interaction between these trees while building the trees. Different kinds of models have different advantages. The random forest model is very good at handling tabular data with numerical features, or categorical features with fewer than hundreds of categories. Unlike linear models, random forests are able to capture non-linear interaction between the features and the target. Since the linear model above did not perform well, we guess there should be a nonlinear relationship between target variable and features which means nonlinear model like random forest may predict well.One important note is that because the result of decision trees will vary a lot as the parameters change a little, it is important to try different combinations of parameters in order to find the best fitted model.

3.1.3 Linear SVR

Another common approach is Support Vector Regression. The SVR uses the same principles as the SVM for classification: to minimize error, individualizing the hyperplane that maximizes the margin. However, the implementation of sklearn.SVR is based on libsvm. The fit time complexity is more than quadratic with the number of samples, which makes it hard to scale to datasets with more than a couple of 10000 samples. So instead, we use skearn.LinearSVR. It is similar to SVR with parameter kernel='linear', but implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples. The non-linear SVR model may give a better result, but in this project, we only model the linear SVR considering computational efficiency.
We use GridSearchCV to do model tuning to find the optimal C value that gives best prediction and cross validation with fold 5 to avoid overfitting.

3.1.4 KNN

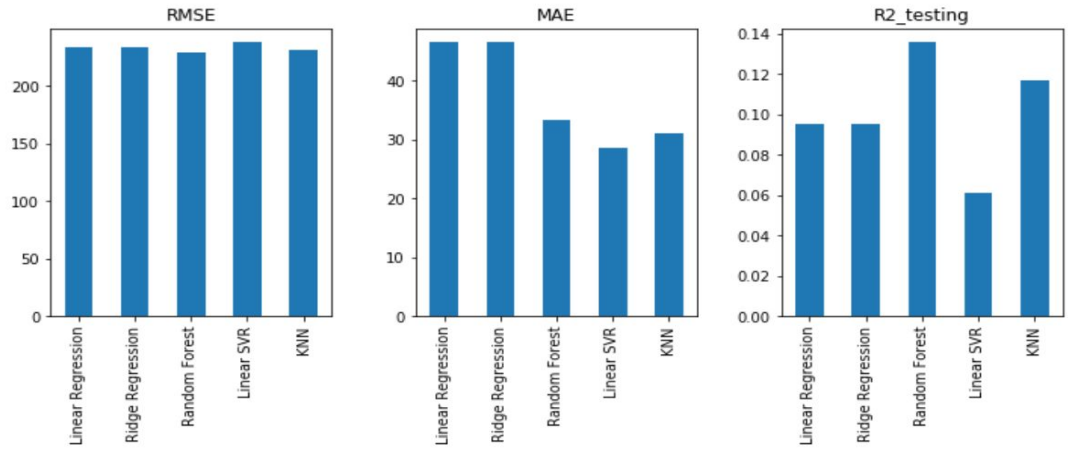
The last way to approach is by the k-nearest neighbors algorithm. It is a form of memory based learning wherein we don't learn a function of features to estimate the target variable. It searches the training data for the k nearest neighbors, as calculated by Euclidean distance function, then does the regression process based on the value of these k neighbors. To find the optimal value of k, we compared the performance of models with k in the range of (5,25).

3.2 Evaluation metrics

We used R-squared, Root Mean Squared Error(RMSE) and Median Absolute Error (MAE) for evaluation. R-squared measures how close the data are to the fitted regression line, which reflects the percentage of the response variable variation that is explained by a linear model. RMSE is simply the root of mean squared error and we used Median Absolute Error because it's less sensitive to outliers than Mean Squared Error and translates nicely to a dollar amount that is relative to price.

4. Results

The following is the evaluation metrics of each model:



From the above histograms, we can see that after implementation, the test data accuracy of Linear Regression turns out to be really low, so clearly linear regression is too oversimplified for our problem. Also, the Ridge Regression perform similarly to Linear Regression with a little bit differences in RMSE and MAE which means it also does not improve the result a lot.
Because we have a relatively large amount of data, the probability of over-fitting seem to be low. We also do the R2-score in training set and confirm that each R2-score is always between 40% and 70% which means these models do not have a too high R2 as not true.
We can see that Random Forest, Linear SVR and KNN models perform both have their advantages and weaknesses. Random Forest has the lowest RMSE while Linear SVR has the lowest MAE. However, as the models' RMSE and MAE do not vary a lot, we can use R2 score in testing dataset to determine which model perform better. As Random Forest model has the highest R2 score as 13.57%, we can conclude that it is the best-performance model. This is understandable since the reality relationship seem to be nonlinear. In general, our best model Random Forest explains 13.57% of the variation in the target variable around its mean for any given test dataset.
Overall, we see that ensemble methods are promising, but require a considerable amount of computational time as well to tune for the best parameters. With further and more comprehensive tuning, it is very likely that the Median Absolute Error could be decreased for an ensemble method.

5. Discussion

5.1 Strengths and Weaknesses of The Proposed Methodology

Linear regression is the simplest regression model that assuming a linear relationship between target variable and response variables. However, as the relationship in real dataset is high-possible of not being perfectly linear, this model would not fit the data well. Ridge Regression improves Linear Regression a little but still not as good as expected.
Compared to Linear Regression, Linear SVR performs well no matter the data is linear or not. However, it is more suitable for classification instead of regression.

Random forest model reduces the probability of overfitting by averaging the prediction results of a diverse set of trees. It is always compared with the decision tree - the total bias of random forest is higher, as it trains independent trees on random samples of data. However, the final prediction result is the average of all individual trees, which alleviate the increased bias by reducing variance. Therefore, the accuracy of the random forest is always slightly higher than the decision tree.

KNN model is non-parametric, which means it does not make an assumption about the underlying data distribution pattern, and this reduces the bias due to the model selection compared to parametric methods. However, this model can be problematic since we have a group of features. The high dimensionality data not only prolong the computation time, but also lower the accuracy of KNN. Also, this model is very sensitive to outliers.

5.2 Improvements and Limitations

The results are fairly well because the price depends on various factors that are far more than the response variables we used, but it is not good-enough since the best model only explain 13.57% of the variance in given test dataset. There are some more can be added as response variables such as the number of rooms, number of bathrooms, number of subway lines in 300m, pet-friendly or not, etc. Also, as from some predictions, we conclude the regression to be linear, we can try more polynomial regression models that may fit the problem better and may get more accurate results.

6. Conclusion

6.1 Summary of the problem, findings and limitations.

This project using different models including Linear Regression, Ridge Regression, Linear SVR, Random Forest, and KNN to predict prices for airbnb in NYC. Through comparing RMSE, MAE, and R2-score in testing dataset, we conclude that Random Forest outperforms others. However, as R2 of 13.57%, our best model could not predict the price very precisely.

6.2 Future work directions:

- Try out deep learning models such as neural networks to improve performance.
- Use better quality data which includes review contents. And use models with natural language processing (NLP) to process reviews
- In addition to predicting base prices, a sequence model could be created to calculate daily rates using data on seasonality and occupancy.
- Tailor the model to deal with new listings/hosts who lack the information such as availability and number of reviews.

References

"Predicting Airbnb prices with machine learning and deep learning"Laura Lewis, May 22 https://towardsdatascience.com/predicting-airbnb-prices-with-machine-learning-and-deep-learning-f46d44afb8a6
"Airbnb Pricing Predictions" https://airbnb-pricing-prediction.herokuapp.com/
Friedrichs, F., & Igel, C. (2005). Evolutionary tuning of multiple SVM parameters. Neurocomputing, 64, 107-117.
Zhang, A. X., Noulas, A., Scellato, S., & Mascolo, C. (2013, September). Hoodsquare: Modeling and recommending neighborhoods in location-based social networks. In Social Computing (SocialCom), 2013 International Conference on (pp. 69-74). IEEE. http://arxiv.org/pdf/1308.3657.pdf
Tang, E., & Sangani, K. (n.d.). Neighborhood and price prediction for San Francisco Airbnb Listings
Kalehbasti, Pouya Rezazadeh,Nikolenko, Liubov & Rezaei Hoormazd. Airbnb Price Prediction Using Machine Learning and Sentiment Analysis (2019) 1907.12665.arXiv.cs.LG

Appendix - Evaluation result of models

	RMSE	MAE	R2_testing
Linear Regression	233.579178	46.508230	0.094903
Ridge Regression	233.579188	46.508723	0.094903
Random Forest	228.254708	33.221637	0.135696
Linear SVR	237.890616	28.451702	0.061182
KNN	230.745922	31.090909	0.116727