GR 5291 Final Project
Construct a Recommendation System for Airbnb
12/10/2018
Professor: Demissie Alemayehu

Group Member:
Di WU, Feifei YAN, Ge ZHANG, Hongbo ZHU, Lin WU, Jiansong CHEN, Jiaqi DONG, Junkai ZHANG, Shuoqiu ZHENG, Wanyi ZHENG, Yanzi SHEN, Yaxin WANG, Yinan WANG, Yiru (Yi) ZHONG, Zhaojun ZHANG

Introduction

(I) Background

Nowadays, when people travel around and need to find some places to take a rest, hotel is no longer the only choice. Via *Airbnb.com*, the travellers could easily search a place that caters for their requirements.

What is Airbnb? Airbnb is an online marketplace that allows people to rent out their spare rooms to the travellers. It was established in 2008 and has become increasingly popular since 2010. Through Airbnb, customers (guests) could find small family hotels run by private rooms owners (hosts). Airbnb offers two sides of services, one side for the hosts who would like to cash in on spare space, the other side for guests who seek good value accommodation. To make the selection more transparent, Airbnb itself owns a peer-review system, where hosts and guests could leave reviews for each other after a reservation and the guests can choose to offer a rating to the property according to their experiences. These ratings and reviews are perceived as the reflection of room/service conditions, and become a critical factor for the potential guests to consider when making final booking decision.

However, sometimes the ratings do not well serve its purpose: they might be distorted unconsciously or on purpose, and thus do not reflect the real situation. For instance, a newly listed property might only have one rating of score 100, which is obviously not representative. In addition, some ratings might be intentionally inflated or scored low by either a host or a guest, malicious rating and astroturfers for example, and thus could be trusted as a fair reference for booking.

(II) Objective

In order to help customers find places that better suit their needs, we work towards a new rating system that offers fair ratings by building machine learning models and customized text search by employing Natural Language Processing tools. Apart from providing guidance to customers, we intend to describe what factors should hosts take into account so as to improve their property ratings.

(III) Data Overview and Planned Analysis Methods

We obtain our data through *Inside Airbnb*, which gathers a large group of datasets that cover Airbnb properties and booking information in different countries around the world. And then we narrow down our scope by focusing on the New York City.

The dataset contains over 38,315 listing properties, and consists of variables in terms of price, customer review ratings, bedroom, clean fee, availability, number of minimum

nights etc. Based on our project objective, 28 variables are initially selected as the main predictors as below, and customer review scores rating as the response variable. In process of data cleaning, we convert categorical variables into dummy variables in our model. After transformation, we finally got 432 variables in total for model construction.

host_is_superhost	property_type	availability_30	availability_60	availability_90	host_has_profile_pic
bathrooms	bedrooms	weekly_price	monthly_price	deposit_price	cleaning_fee
minimum_nights	accommodates	amenities	instant_bookable	host_response_time	neighbourhood_cleansed
price	bed_type	host_identity_verified	guests_included	host_varification	reviews_per_month
availability_365	require_guest_phine_verification	extra_charge	house_rule		

In order to accurately predict the review score ratings, we conduct EDA for outlier, correlation and distribution check in advance. Then we build regression models to fit the actual ratings with our variables. We use 9 different algorithm methods: OLS, LASSO, RIDGE, Elastic Net, SVM, AdaBoost, Gradient Boosting, XGBoost and Random Forest. In addition to analysis on the numerical data, we also execute natural language analysis on guest reviews, which will incorporate the rating scores to give better recommendation.

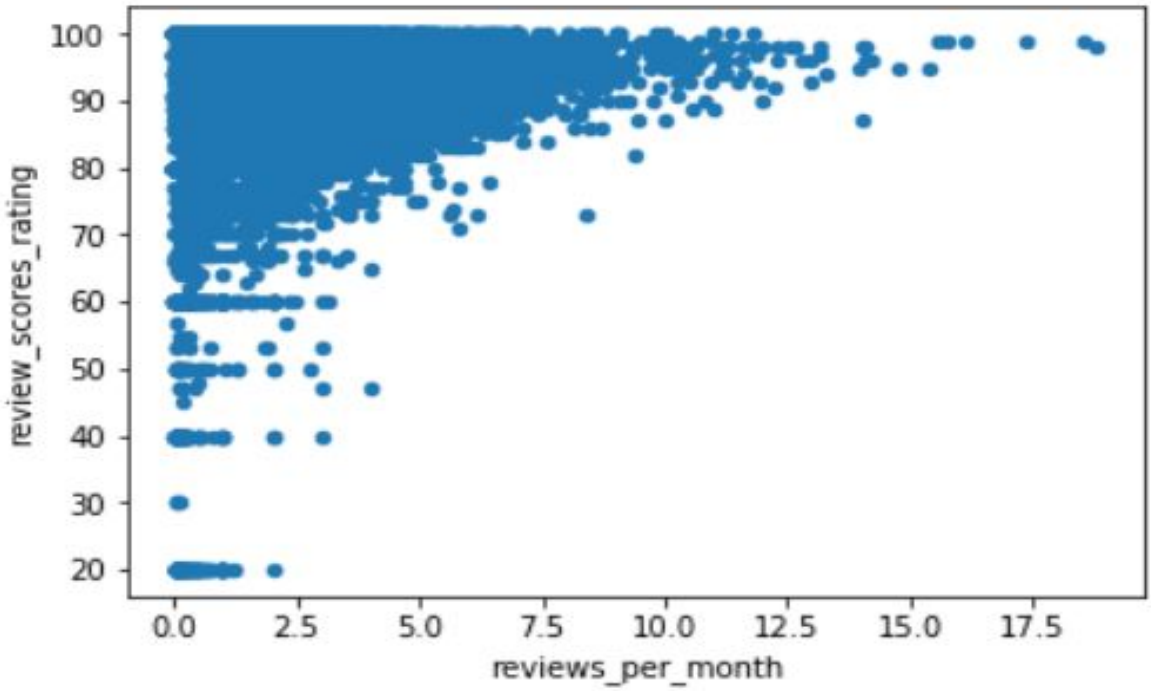
EDA

After acquiring data, we used summary statistics and visualizations to better understand data, and find clues about the tendencies of the data, its quality and to formulate assumptions and the hypothesis of our analysis.

When we check model assumptions, it is noticed that distribution for review score ratings is left skewed, which can be seen from the histogram and QQ plot. With summary statistics, skewness is -3.42 and kurtosis is 19.37. Also, the minimum, maximum, mean, standard deviation, and IQR for review score rating is 20, 100, 93.69, 8.55 and 9. There is no missing data in our dataset after removing all rows if the values of dependent variables are missing or replacing missing data with means if the values of independent variables are missing. We checked outliers using boxplot, scatter plot and z-score and identified some extreme points. To fix this problem, we use robust method in the model and only remove unreasonable outliers. A correlation matrix was plotted to calculate the

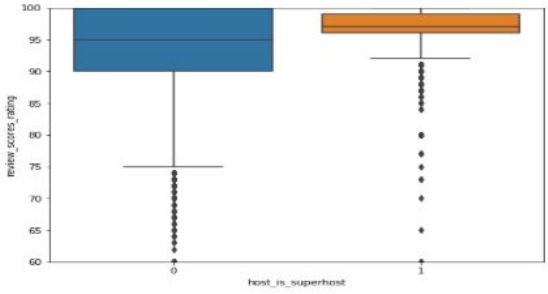
correlations between different variables and it is obvious that there are some variables highly correlated with each other. Regularization methods (Lasso, Ridge and Elastic Net) are used here to reduce correlations.

We draw several scatter plots to explore relationship between numerical predictors and independent variables. From the plot between review score rating and number of reviews per month, we can see that property lists with lowest review ratings only has one or two reviews, which makes the rating unreliable if the guest is biased or gives low ratings for only personally reasons. This justified the necessity of our project.

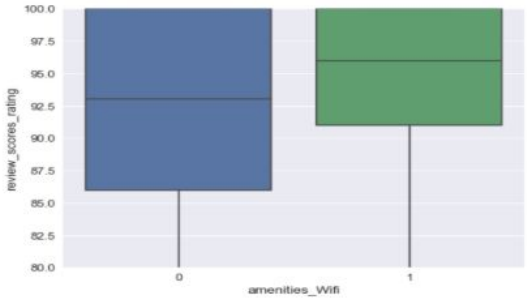


For dummy variables, there are some interesting findings. If a host is a Superhost based on some criteria set by Airbnb and has WIFI and dryer at home, then he or she is more likely to get a good review score. However, if smoking is allowed at home, then guests will probably give a lower rating which totally makes sense.

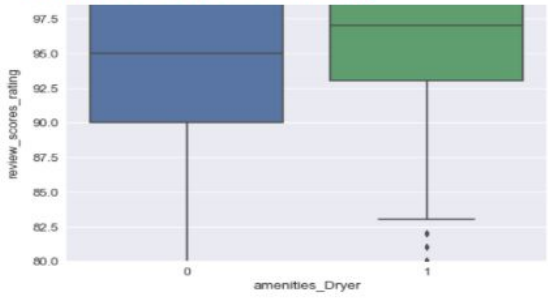
Host is Superhost vs Review Score Rating



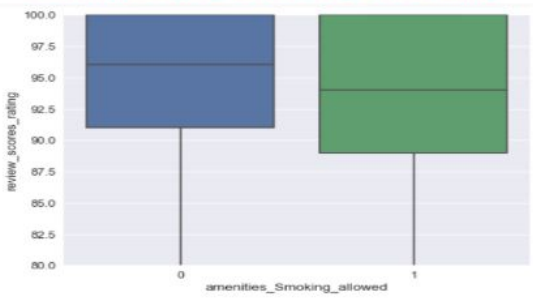
Wifi vs Review Score Rating



Dryer vs Review Score Rating

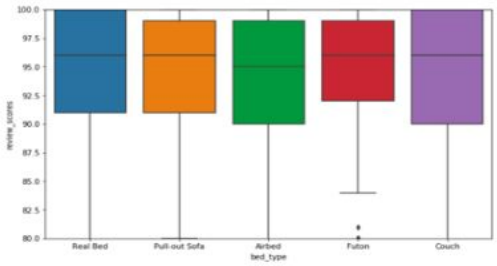


Smoking allowed vs Review Score Rating

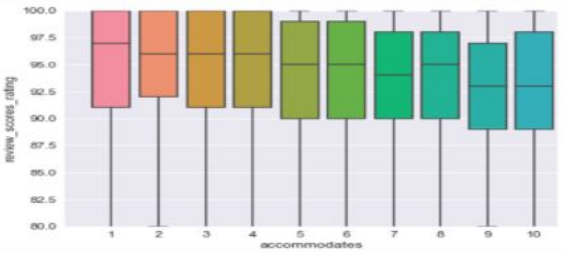


For categorical variables, we also draw boxplots. For bed types, we don't recommend airbed as it receives the lowest score on average. As the number of accommodates increases, customer satisfaction decreases and thus customers give lower ratings. It is the same with the No. of bedrooms as there will be more people living in the same property, but it is the opposite to the No. of bathrooms as it is more convenient for guests.

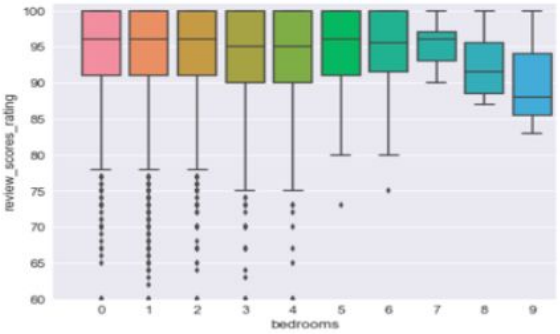
Bed type vs Review Score Rating



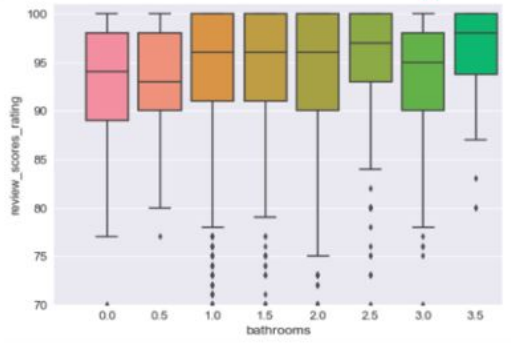
Accommodates vs Review Score Rating



Bedroom vs Review Score Rating



Bathroom vs Review Score Rating



Algorithms

As stated in the planned analysis, we use 9 machine learning algorithms to predict the review score rating, and then use the predicted rating as the basis of property ranking.

(I) RMSE

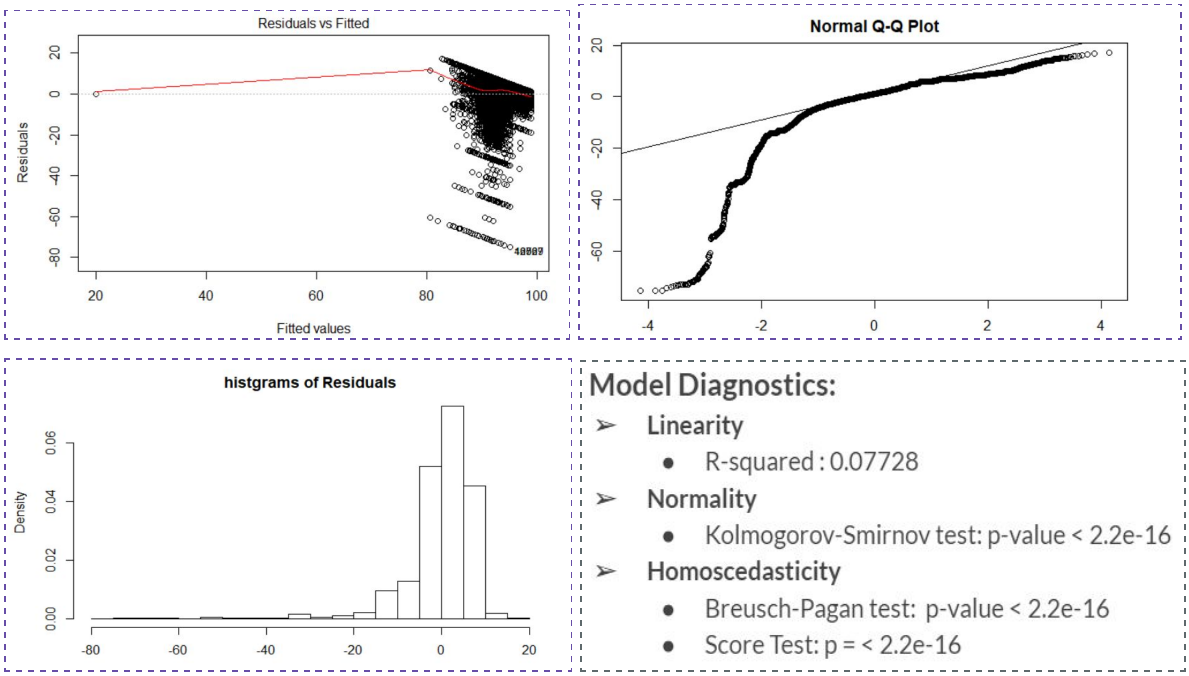
We use Rooted Mean Squared Error, a classic way to evaluate the accuracy in regression problems, to measure our model performance. RMSE is calculated by taking the square root of the mean of the squared errors as shown in below equation:

$$RMSE = \sqrt{\frac{\Sigma_{t=1}^{T}(y_t - \widehat{y_t})^2}{T}}$$

(II) OLS

Ordinary Least Squares (OLS) linear regression is a statistical technique used for the analysis and modelling of linear relationships between a response variable and several predictor variables. The goal of model selection is to choose a sparse model that adequately explains the data. Subset selection is one of the most frequently used model selection methods. It seeks to find a subset of the available predictor variables that can properly predict the response variable. Forward selection is a searching algorithms of subset selection that starts with intercept-only model. It adds one variable that improves the model's goodness-of fit using a certain criterion, and repeats this step until the optimal model is obtained. Here, we employ the Bayesian Information Criterion (BIC) as the criterion; that is, model with the lowest BIC score is finally selected for further analysis. By forwards subset selection with the smallest BIC, we finally get the model: review_scores_rating~ host_is_superhost + availability_365 + instant_bookable + amenities_Hair_dryer + host_varification_.reviews. + amenities_Dryer + property_type_Train + amenities_Wifi + amenities_Smoking_allowed.

Next, we check the model assumptions of linearity, normality, and homoscedasticity. We use the residuals vs fitted values plot to check the linear relationship assumptions. The plot indicates non-linear patterns and we can also get the same conclusion by R-squared. From the Normal Q-Q plot and histograms of residuals, we can conclude that the Normality assumption fails, and get the same conclusion by p-value of the Kolmogorov-Smirnov test. Moreover, P-value of Breusch-Pagan test and Score Test are both less than 0.05, which demonstrates that heteroscedasticity exists. In view of the above results, all assumptions are violated.



(III) Ridge/Lasso

Ridge regression produce more stable coefficients than OLS by adding L2 regularization (sum of squared coefficients) to the OLS function to penalize residuals. Accordingly, we do not need to consider multicollinearity among variables when preparing data for ridge regression, since all coefficients turn to be stable when the value of model parameter, lambda, increases. In the meanwhile, we choose the best lambda by cross validation. The RMSE of ridge regression is approximately 10.027, which is slightly better than RMSE of linear regression. The result makes more sense since ridge regression generalizes better due to smaller sensitivity to extreme variance in the data. Compared with Lasso, Ridge regression does not shrink the model but changes the weights of every coefficients. Thus, our model still contains all of the variables with non-zero coefficients.

Lasso poses L1 regularization to OLS function, so Lasso shrinks some coefficients to zero, which provides an alternative way to do variable selection and completely addresses the collinearity problem in OLS. Like ridge regression, all numerical variables has been scaled in order to prevent penalizing some coefficients more than others. By using cross-validation, we select the best tuning parameter and ultimately find the suitable variables. Barring lasso and ridge regression method, a new regularization method, namely elastic net, that combines L1 and L2 regularization to the OLS function performs much better at times. The RMSE of lasso and elastic net are 10.027 and 10.263 respectively. Compared all three method, lasso regression seems more robust and finally

achieve the lowest test RMSE.

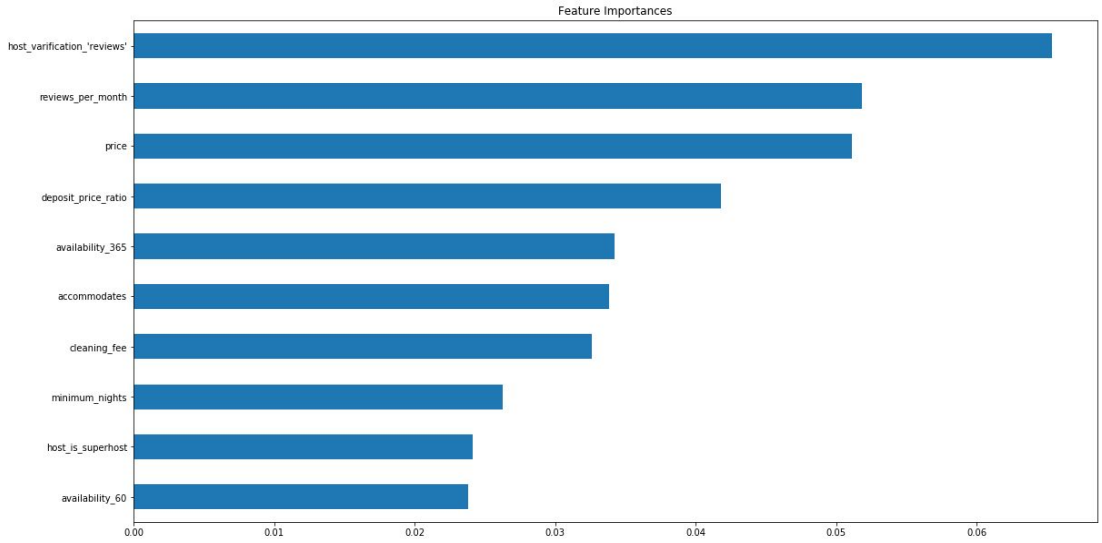
(IV) Gradient Boosting Machines (GBM)

Boosting algorithms are effective at controlling both bias and variance. It is a technique in which the predictors are made sequentially rather than independently. Boosting combines a collection of weak learners and improves the prediction accuracy. The logic of the technique is to learn from the mistakes of the previous predictors. That is, the observations have an unequal probability of appearing in subsequent model and ones with higher errors are given higher weights. Gradient Boosting is an example of boosting algorithm.

Like the assumption of linear regression that the sum of residuals is zero, the intuition behind gradient boosting is to make the residuals randomly distributed by leveraging the pattern in residuals and improve a weak prediction model. Once there is no pattern in residuals that could be modeled, we can stop to avoid overfitting. It is crucial to choose the stop criterion of the gradient boosting algorithm to deal with overfitting on training data. Here we use gradient boosting in a regression problem, which predicts the customer review ratings. The target of our algorithm is to minimize our loss function RMSE with predictions. Using gradient descent, we update predictions based on a learning rate and find the values when RMSE is minimum.

The overall parameters of GBM can be grouped into three categories, tree-specific parameters, boosting parameters and miscellaneous parameters. The terminologies we introduce here are scikit-learn python specific. Firstly, we fix learning rate (0.15) and tune the number of estimators through grid search testing values from 100 to 180 in steps of 10. From our result, 160 is the optimal estimators for 0.15 learning rate. Then in the tree-specific category, our model tune min sample split, max depth, min sample leaf, and max features. It is important to note that min sample split, max depth and max features should be tuned using CV otherwise they can easily lead to overfitting. Our tuned model gives the final RMSE 9.5484.

GBM can be used to solve almost all objective functions that we can write gradient out, including regressions and rankings. It generally gives promising results if the tuning parameters are chosen wisely. However, GBMs are more sensitive to overfitting with noisy data and training generally takes longer since the trees are built sequentially. In addition, GBM provides the ordered feature importance. And the feature importance plot is shown as follows.



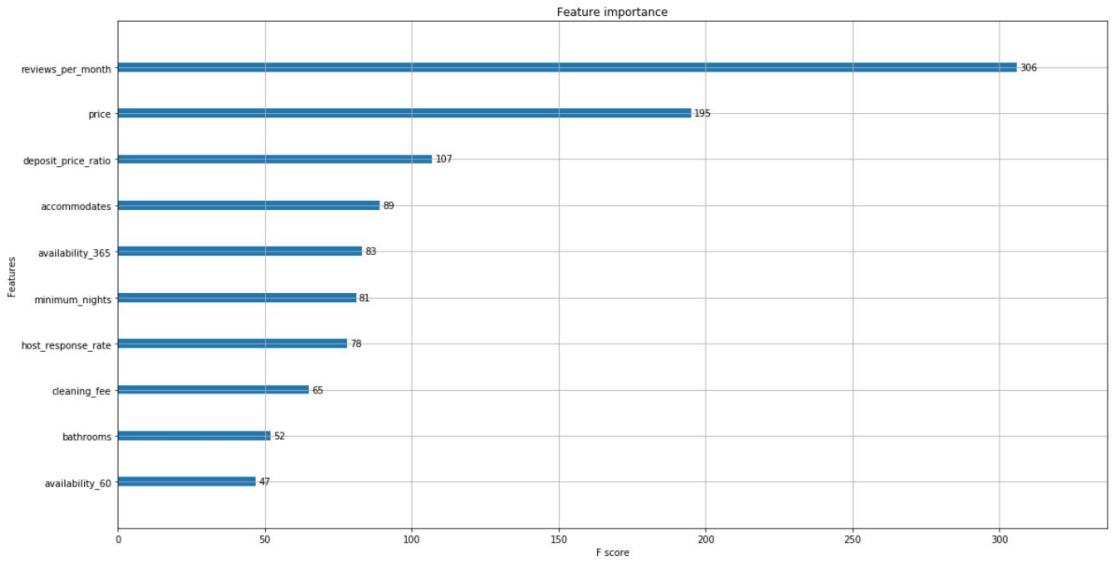
(V) XGBoost

XGBoost (eXtreme Gradient Boosting) is one of the most popular machine learning algorithm these days. It has several advantages over the gradient boosting algorithm, which has no regularization like XGBoost. And XGBoost implements parallel processing by using all cores of the computer and is blazingly faster as compared to GBM. Another important advantage is the algorithm makes splits up to the max depth and then starts pruning the tree backward which avoids missing a good split after a bad one. Besides, XGBoost has value adds in defining custom optimization objectives and evaluation criteria, handling missing values and built-in Cross-Validation. However, the drawback of XGBoost is that there are too many hyperparameters need to be tuned, such as number of boost round, learning rate, max depth and so on.

The whole parameter tuning process starts from the cross-validation number of boosting round by fixing the learning rate. Then the process focuses on the parameters have the highest impact on the model outcome, such as the max depth and min child weight, which defines the minimum sum of weights of all observations required in a child. Lastly, regularization parameters are tuned, which includes gamma, subsample,colsample bytree and alpha. After the whole process, the final model of XGBoost is tuned by using the training dataset and the final RMSE of XGBoost is 9.557016.

Similar to the GBM, XGBoost also can produce the feature importance plot to display the number of times a feature appears in a tree, in other words, a high number indicates that the feature has an important impact in the predictive model. Below is the feature importance of XGBoost. The further discussion of feature importance is saved for the

machine learning ensemble model.



(VI) AdaBoost

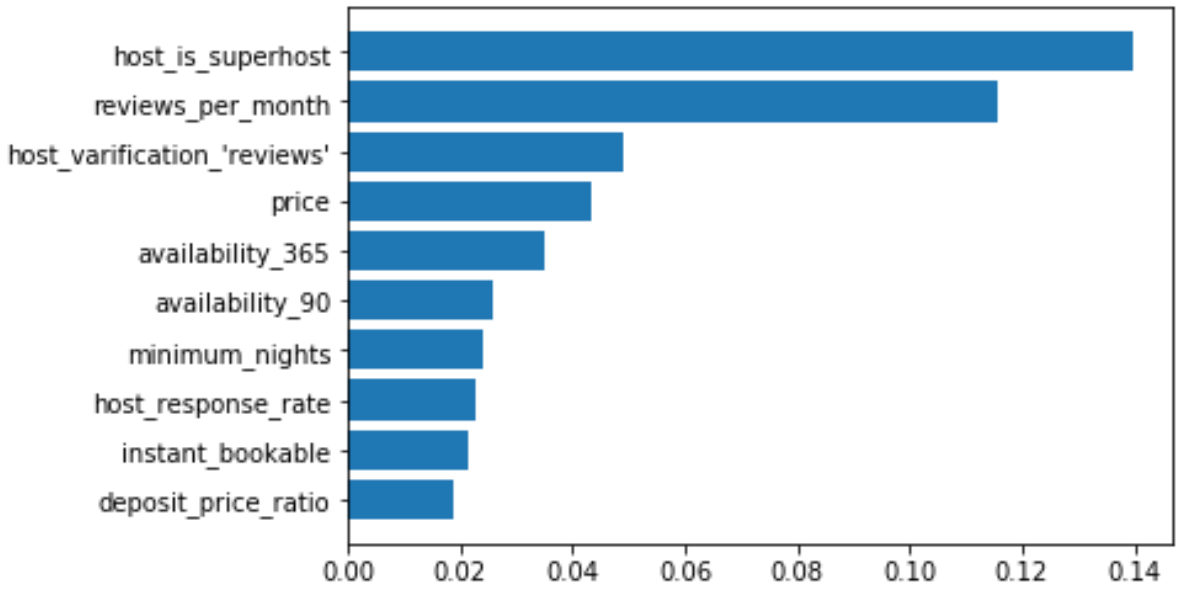
Developed by Yoav Freund and Robert Schapire, AdaBoost is another type of boosting algorithms that can be applied to both classification and regression. AdaBoost is more sensitive to outliers and noisy data, while it is more resistant to overfitting than many other machine learning algorithms.

AdaBoost utilizes multiple iterations to construct a single composite strong learner by iteratively adding weak learners. It starts by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset while the weights of instances are adjusted according to the error of the current prediction. Therefore, the result is a classifier with higher accuracy than the weak learners' classifiers.

(VII) Random Forest

Random forest is also a widely used machine learning method. Typically we use random forest to solve classification problems, while this time we will perform random forest regression. The basic idea of using random forest to solve regression problems is similar to random forest classification. We train a bunch of regression trees as weak learners and take the average of them to predict our response.

Random forest can easily deal with high dimensional data without feature selection. It is also very easy to tune parameters. The randomness in different weak learners prevents the model from over fitting. Feature importance generated by random forest is shown in the below graph:



(VIII) Support Vector Regression

Support vector machines (SVM) are supervised learning models that widely used for classification and regression analysis. Based on Airbnb circumstance, a version of SVM for regression, support vector regression (SVR), is used to analysis and predict data. Unlike tree base models to apply weak learners in the model, SVR is to find a hyperplane, which is a function, that minimizes error and maximizes the margin at the same time. In addition, SVM is not restricted on normality or linearity of underlying distribution, thus this model is appropriate for this Airbnb data analysis.

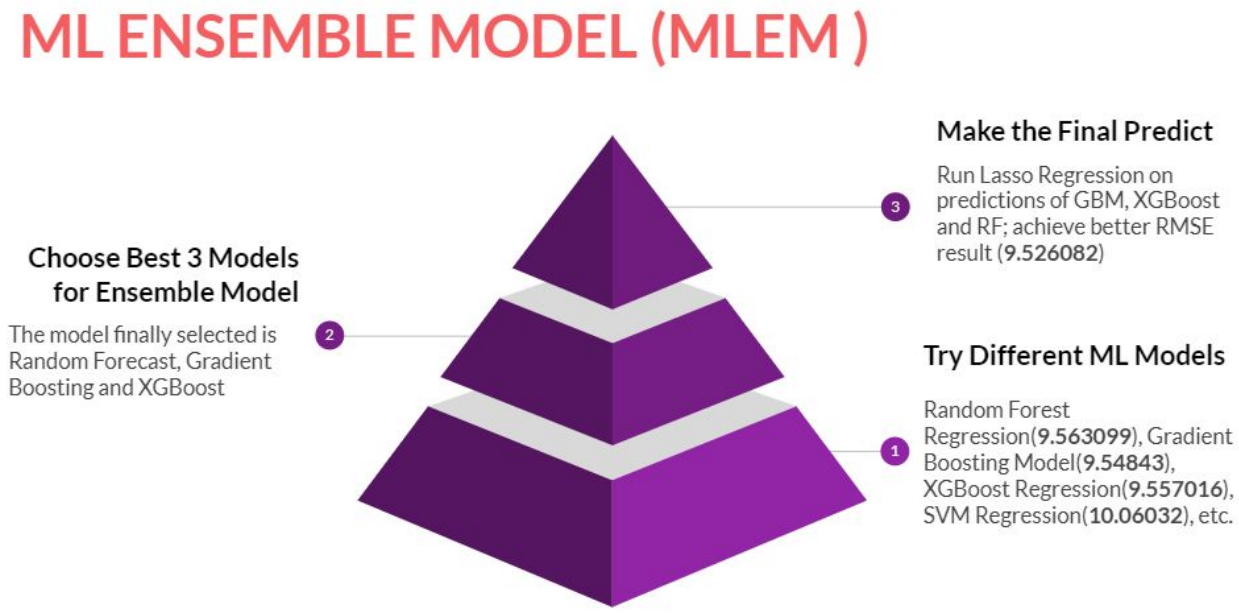
There are three types of hyperplane can be chose from, linear kernel, polynomial kernel and radial basis kernel. For hyperplane choosing, cross validation is used and the final choice for Airbnb case is radial basis kernel. The final RMSE of test data is 10.0603.

Machine Learning Ensemble Model

When we try to predict the target variable using other dependent variables, the causes of the difference between actual and predicted values are noise, variance and bias. Note that noise is an irreducible error, hence ensemble is applied to reduce variance and bias. Ensemble is a technique that collects predictors to give a final prediction.

Our group decide to use an ensemble model to give the final predicted review scores. First, we tried various machine learning models, including all the above machine learning models. Then, we select the top 3 algorithms based on the model performance, RMSE. The

machine learning models we used include GBM, XGBoost and Random Forest Regression. The last step is to use the prediction results from these models as the input of LASSO regression, which actually gave a better RMSE result in the test dataset.



NLP

Conclusion

(I) Summary

To conclude, we highlight a few key observations that we made in our experiments. We have trained 9 models with attribute of RMSE and rank them in decreasing order: Elastic Net, OLS, SVM, RIDGE, LASSO, AdaBoost, Random Forest, XGBoost and Gradient Boosting. In this case, Gradient Boosting has best testing result with lowest RMSE. In order to improve the result, we have constructed the ensemble model with GBM, XGBoost and Random Forest and achieve a better RMSE. By using this model, we have gotten scorings for every single listing house from Airbnb dataset. In addition, we have also perform NLP on the reviews of housing using LDA models. Moreover, to improve the model, we would imply more attributes to compare the models rather than using only

RMSE.

(II) Improvement

(i) Data: We currently focus on the listing properties in new york city. In the following work, we are going to expand our scope to large area in United States. For example, boston, washington.DC and other cities with a mass of listing properties.

(ii) Methods: In the model construction part, we totally used 9 methods, including both supervised and unsupervised machine learning algorithms. The model with best performance is Gradient Boosting, and the RMSE is 9.5484. In the following work, we are going to keep seeking models with great performance in prediction accuracy.

(iii) Shiny app: On account of the limitation of time, we save Shiny App for the future improvement. The Shiny App will serve as the data visualization platform that shows the searching results based on the predictions from our machine learning ensemble model and LDA model.

Appendix