|  |  |
| --- | --- |
| Download Robert H Smith School Of Business Logo Png Transparent -  University Of Maryland PNG Image with No Background - PNGkey.com | **BUDT 758T**  **Final Project Report**  **Spring 2023** |

**Section 1: Team member names and contributions**

Akshar Yadav: Worked on grid search, and xgboost model.

Akshit Malik: Worked on logistic regression and feature engineering.

Shashank Patil: Worked on ridge and lasso model and feature engineering.

Shivam Nautiyal: Worked on random forest model and report writing.

Kalpana Sharma: Worked on cleaning the data and conducting exploratory data analysis.

**Section 2: Business Understanding**

A business case for the development and utilization of a predictive model to determine high booking rates for Airbnb listings. By analyzing historical data and using advanced data mining techniques, our team has created a model that predicts the probability of a listing achieving a high booking rate. This valuable tool can be leveraged by various stakeholders, including individual hosts, property management companies, and Airbnb itself, to optimize their listing strategies, make informed decisions, and enhance their overall revenue and user experience.

Target Audience: The predictive model is designed to cater to a wide range of users, including:

1. Individual Hosts: Homeowners who list their properties on Airbnb and seek to maximize their occupancy rates and rental income.
2. Property Management Companies: Organizations that manage multiple Airbnb properties and aim to optimize their portfolio performance.
3. Airbnb: The platform itself can utilize this model to provide personalized recommendations and insights to hosts, ultimately enhancing the overall user experience and driving higher engagement on their platform.

Value Proposition:

1. Improved Revenue Generation: Hosts can leverage the model to optimize their listing strategies and pricing decisions, increasing the probability of securing more bookings and generating higher rental income.
2. Enhanced User Experience: By predicting high booking rates, hosts can tailor their listings, amenities, and customer service offerings to attract more guests and provide a better experience, leading to positive ratings and reviews.
3. Resource Allocation: Property management companies can use the model to allocate their resources efficiently, focusing on properties with higher predicted booking rates and prioritizing their efforts accordingly.
4. Platform Optimization: Airbnb can integrate the model into its recommendation system, providing hosts with personalized insights and recommendations to improve their listing's booking potential. This can drive higher engagement, increase user satisfaction, and lead to higher overall revenue for the platform.

Business Actions Based on Model Output:

1. Pricing Strategy: Hosts can adjust their pricing based on the model's predictions. If a listing has a high probability of achieving a high booking rate, hosts could set slightly higher prices to maximize their revenue. Conversely, for listings with lower probabilities, hosts might consider offering discounts or adjusting pricing to attract more guests.
2. Listing Optimization: Hosts can modify their property listings based on the model's feature importance analysis. By identifying the key factors driving high booking rates, hosts can improve their property descriptions, add attractive amenities, or enhance the quality of listing photos to increase their chances of attracting guests.
3. Marketing and Promotion: Hosts can leverage the model's predictions to focus their marketing efforts on listings with higher predicted booking rates. This can include targeted advertising, social media campaigns, or partnerships with local tourist attractions to enhance visibility and attract potential guests.
4. Resource Allocation: Property management companies can use the model's predictions to allocate their resources effectively. By prioritizing properties with higher probabilities of achieving high booking rates, they can allocate maintenance, cleaning, and customer service resources, accordingly, optimizing operational efficiency.
5. Platform Recommendations: Airbnb can utilize the model's output to provide personalized recommendations to hosts. These recommendations could include pricing suggestions, listing improvements, or promotional opportunities tailored to each host's specific property and market.

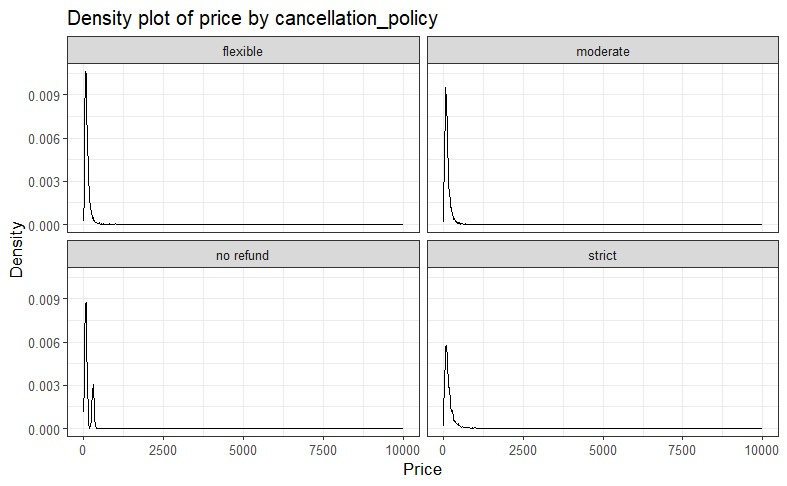
Top of Form

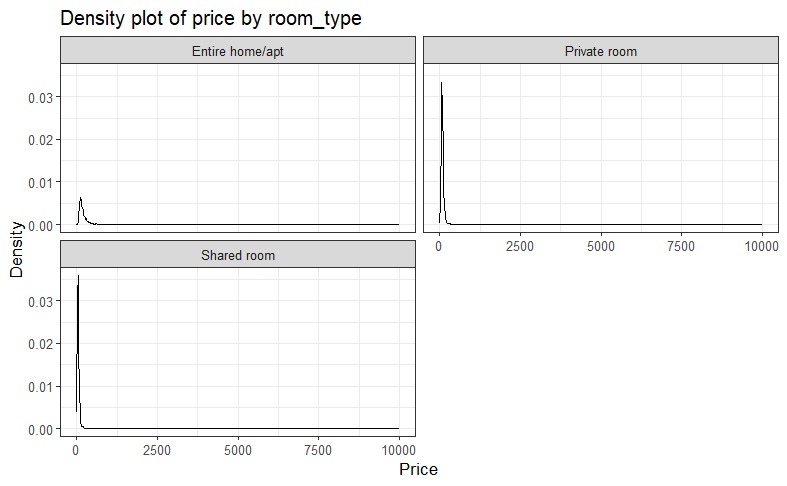
**Section 3: Data Understanding and Data Preparation**

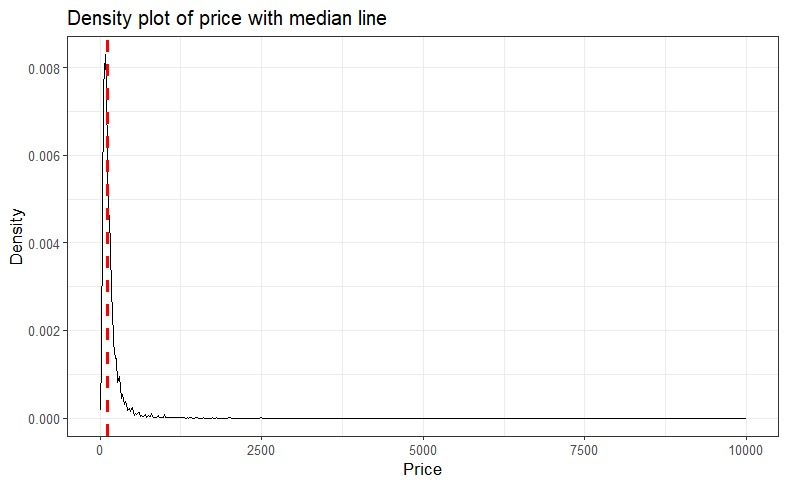
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Feature Name** | **Brief Description** | **R Code Line Numbers** |
|  | cancellation\_policy | It provides information on the rules and conditions related to refunds and penalties in case of cancellations by guests. | 105-108 |
|  | price | The "price" field in the Airbnb dataset represents the cost of renting a listing for a given period, typically per night. It provides information on the financial aspect of each listing, allowing users to compare and make decisions based on their budget. | 72 |
|  | room\_type | The "room\_type" field in the Airbnb dataset indicates the type of accommodation available for booking, such as "Entire home/apt," "Private room," "Shared room," or "Hotel room." | 200 |
|  | bedrooms | The "bedrooms" field in the Airbnb dataset indicates the number of bedrooms available in a listing. It provides essential information for guests to understand the accommodation's size and sleeping arrangements. | 138 |
|  | number\_of\_amenities | This is a derived column from the unstructured Amenities column that counts the total number of amenities in each property. | 178 |
|  | cleaning\_fee\_less\_than\_130 | This is a categorical derived column from cleaning\_fee column where we update the value to YES if cleaning\_fee is less than 130. | 591-592 |
|  | accommodates | The "accommodates" field in the Airbnb dataset indicates the maximum number of guests allowed to stay in a listing. It provides information on the capacity of the property, helping guests find suitable accommodations for their group size | 719 |
|  | bathrooms | This field provides information about the number of bathrooms available in each Airbnb listing, allowing guests to assess the level of convenience and comfort provided by the accommodation. | 137-138 |
|  | total\_years\_service | total\_years\_service in the Airbnb dataset represents the cumulative number of years a host has been active on the platform, indicating their experience and tenure as a host. | 599-604 |
|  | property\_category | The "property\_category" field in the Airbnb dataset categorizes the type of property being listed, such as houses, apartments, bedrooms, or beds. It provides information about the specific category to help differentiate and classify the properties available on the platform. | 91-100 |

**Graphs:**

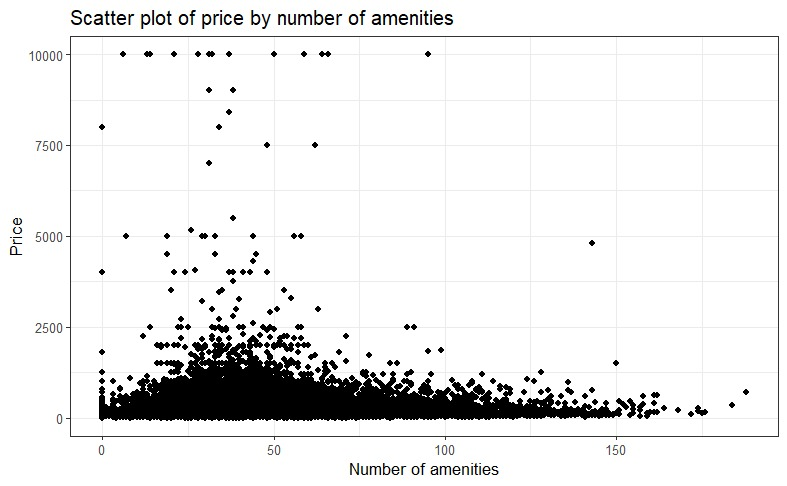
**Please find below the screenshot of plots of each feature listed in the table above:**

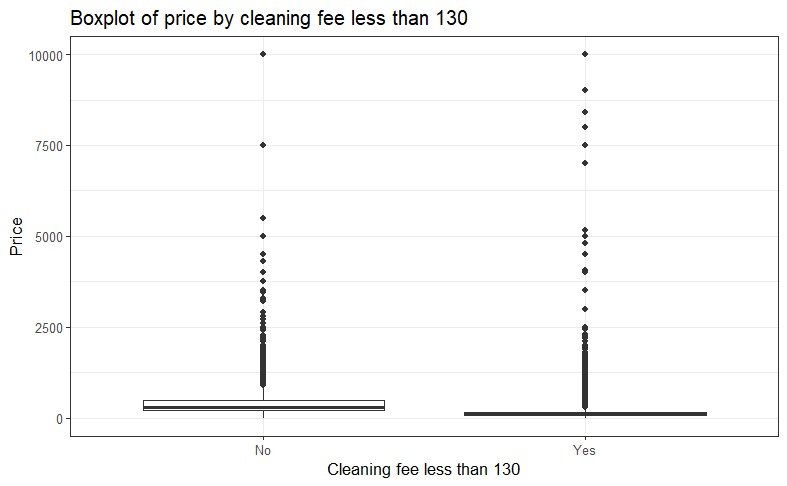
****

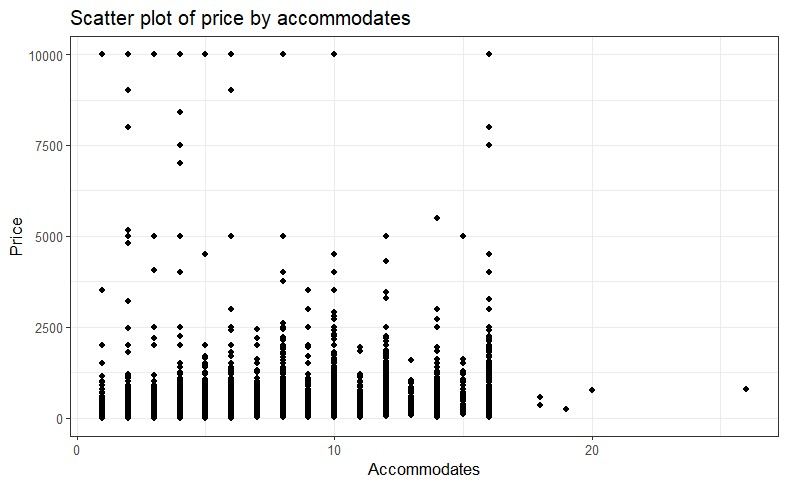
****

****

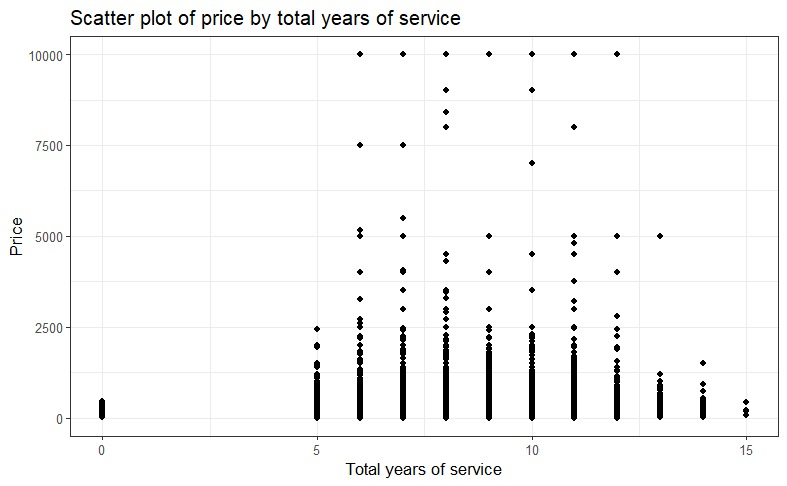
****

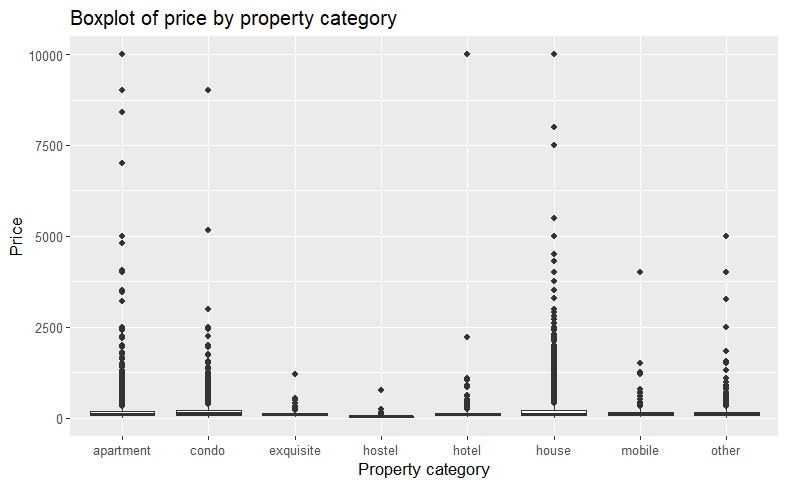
****

****

****







**Section 4: Evaluation and Modeling**

1. **Xgboost Model:**

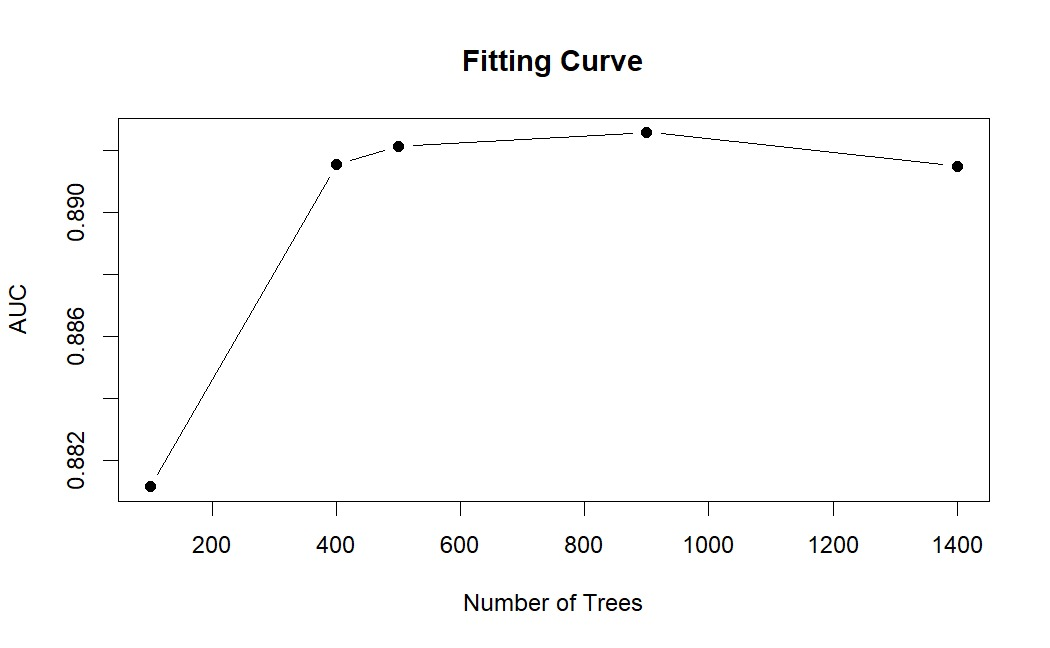
Xgboost is our winning model. The estimated training and generalization performance for xgboostregression is 0.9686 and 0.8934 respectively. xgboost is our winning model because we recorded the highest AUC for this model. The R function used for xgboost model is xgboost() and it is part of the xgboost package in R. We tuned the nrounds parameter which is the number of rounds hyperparameter. We used dynamic values ranging from **100 to 1500**.

To estimate the generalization performance, we used a simple train/validation split and used 70% train data and 30% validation data. The line numbers of the R code are: **923-928**

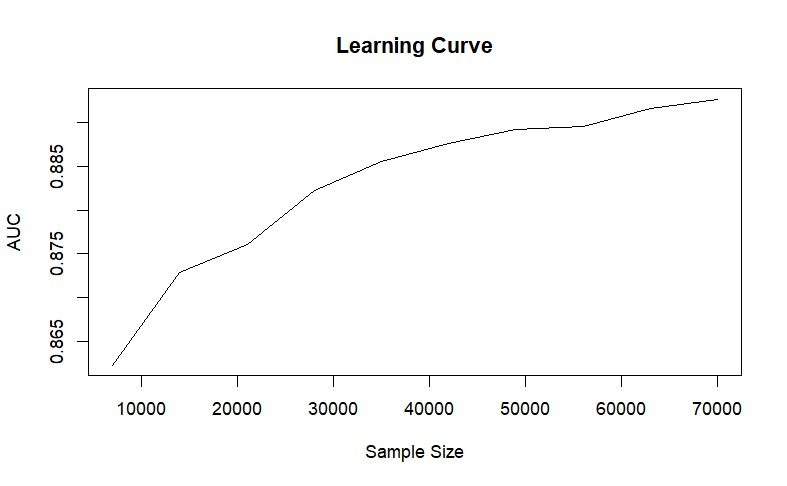
The best set of features used for xgboost regression is as mentioned below:

(accommodates, bedrooms, beds, one\_bed\_per\_person, cancellation\_policy, has\_cleaning\_fee, price, property\_category, bed\_category, bathrooms, charges\_for\_extra, host\_acceptance, host\_response, city\_name, host\_is\_superhost, is\_business\_travel\_ready, instant\_bookable, host\_identity\_verified, has\_min\_nights, require\_guest\_phone\_verification, require\_guest\_profile\_picture, requires\_license, room\_type, has\_security\_deposit, is\_location\_exact, guests\_included, host\_response\_time, offlinegovernmentid, governmentid, number\_of\_amenities, room\_type, has\_hotel\_license, has\_min\_weeks, availablity\_30\_perc, availablity\_60\_perc, availablity\_90\_perc, availablity\_365\_perc, has\_house\_rules, has\_weekly\_price, has\_monthly\_price, has\_access\_info, bathrooms\_per\_person, host\_and\_property\_same\_neighborhood, maximum\_nights, cleaning\_fee\_less\_than\_130, total\_years\_service, first\_review\_since, number\_of\_id\_verification\_options, maximum\_weeks, description\_length, avg\_price\_per\_night, price\_for\_extra, total\_price, cleaning\_fee, security\_deposit, name\_length, price\_groups, nearest\_attr\_dist, nearest\_attr\_count, host\_gender)

**Fitting curve of xgboost model:**

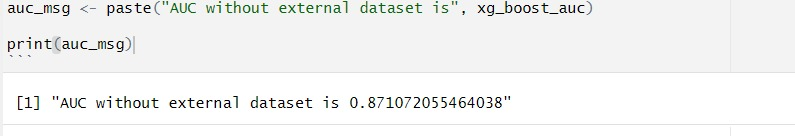
****

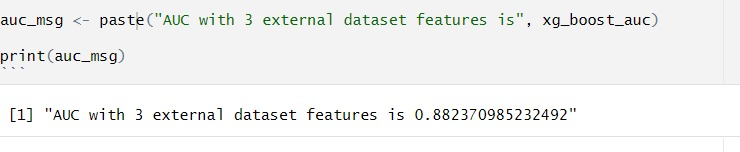
**Learning Curve for xgboost model:**

****

The insight we can draw from the above learning curve is regarding the correct sample size. As the sample size increases, the model can learn from more diverse examples, leading to improved accuracy. The learning curve can indicate the point at which adding more samples yields diminishing returns in terms of performance improvement.

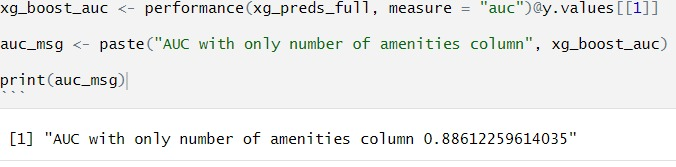
We have incorporated gender and attraction data sources apart from Airbnb. For attractions data source we counted the number of attraction sites within 2 miles of all the properties and calculated the distance for the nearest attraction. We used gender data sources to find the gender of the host in our Airbnb dataset. Please find below the screenshots of the change in AUC after adding these features:

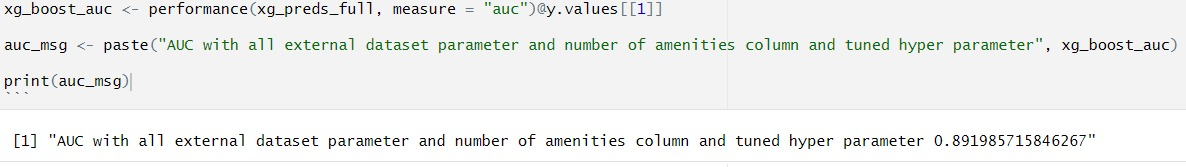




For the unstructured text field, we took the **amenities** column from the Airbnb dataset and created a new feature **number of amenities.** This feature was calculated by counting the number of comma-separated values in the amenities column. The line numbers of the R code are: **1141-1153**

Please find below the screenshots of the change in the AUC after adding this feature:

****

****

1. **Logistic Model**:

We first tried logistic regression to predict high\_booking\_rate. The R function for logistic regression is glm() and it is part of the base package in R. The estimated training and generalization performance for logistic regression is 0.849.

To estimate the generalization performance, we used a simple train/validation split and used 70% train data and 30% validation data. The best set of features used for logistic regression are as mentioned below:

(accommodates, bedrooms, beds, one\_bed\_per\_person, cancellation\_policy, has\_cleaning\_fee, price, property\_category, bed\_category, bathrooms, charges\_for\_extra, host\_acceptance, host\_response, city\_name, host\_is\_superhost, is\_business\_travel\_ready, instant\_bookable, host\_identity\_verified, has\_min\_nights, require\_guest\_phone\_verification, require\_guest\_profile\_picture, requires\_license, room\_type, has\_security\_deposit, is\_location\_exact, guests\_included, host\_response\_time, offlinegovernmentid, governmentid, number\_of\_amenities, room\_type, has\_hotel\_license, has\_min\_weeks, availablity\_30\_perc, availablity\_60\_perc, availablity\_90\_perc, availablity\_365\_perc, has\_house\_rules, has\_weekly\_price, has\_monthly\_price, has\_access\_info, bathrooms\_per\_person, host\_and\_property\_same\_neighborhood, maximum\_nights, cleaning\_fee\_less\_than\_130, total\_years\_service, first\_review\_since, number\_of\_id\_verification\_options, maximum\_weeks, description\_length, avg\_price\_per\_night, price\_for\_extra, total\_price, cleaning\_fee, security\_deposit, name\_length, price\_groups, nearest\_attr\_dist, nearest\_attr\_count, host\_gender)

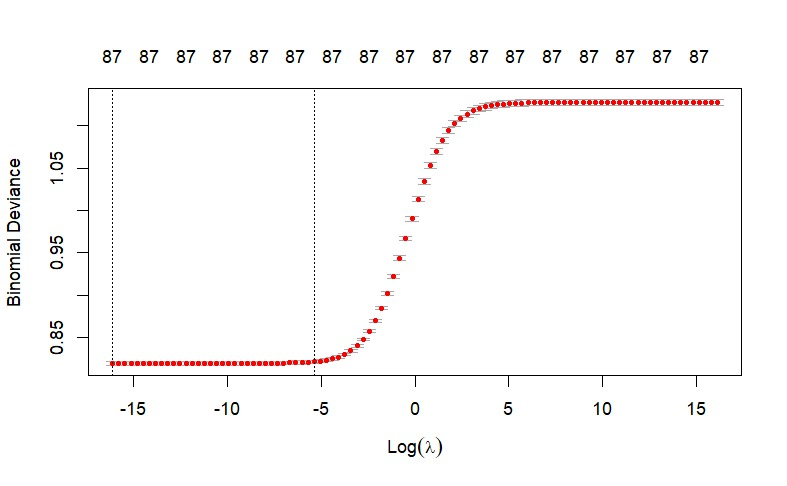
1. **Ridge Regression Model**

The R function we used for ridge regression cv.glmnet() and this function is part of the glmnet library in R. The estimated training and generalization performance for the ridge regression model is 0.8427 and 0.8398 respectively.

To estimate the generalization performance, we used a simple train/validation split and used 70% train - 30% validation data configuration with 5 folds (k=5). We tuned lambda (λ) hyperparameter. We used **100** values in the range of **10-7 to 107**. The line numbers of the R code are: **1186-1205**

The best set of features used for ridge regression is as mentioned below:

(accommodates, bedrooms, beds, one\_bed\_per\_person, cancellation\_policy, has\_cleaning\_fee, price, property\_category, bed\_category, bathrooms, charges\_for\_extra, host\_acceptance, host\_response, city\_name, host\_is\_superhost, is\_business\_travel\_ready, instant\_bookable, host\_identity\_verified, has\_min\_nights, require\_guest\_phone\_verification, require\_guest\_profile\_picture, requires\_license, room\_type, has\_security\_deposit, is\_location\_exact, guests\_included, host\_response\_time, offlinegovernmentid, governmentid, number\_of\_amenities, room\_type, has\_hotel\_license, has\_min\_weeks, availablity\_30\_perc, availablity\_60\_perc, availablity\_90\_perc, availablity\_365\_perc, has\_house\_rules, has\_weekly\_price, has\_monthly\_price, has\_access\_info, bathrooms\_per\_person, host\_and\_property\_same\_neighborhood, maximum\_nights, cleaning\_fee\_less\_than\_130, total\_years\_service, first\_review\_since, number\_of\_id\_verification\_options, maximum\_weeks, description\_length, avg\_price\_per\_night, price\_for\_extra, total\_price, cleaning\_fee, security\_deposit, name\_length, price\_groups, nearest\_attr\_dist, nearest\_attr\_count, host\_gender)



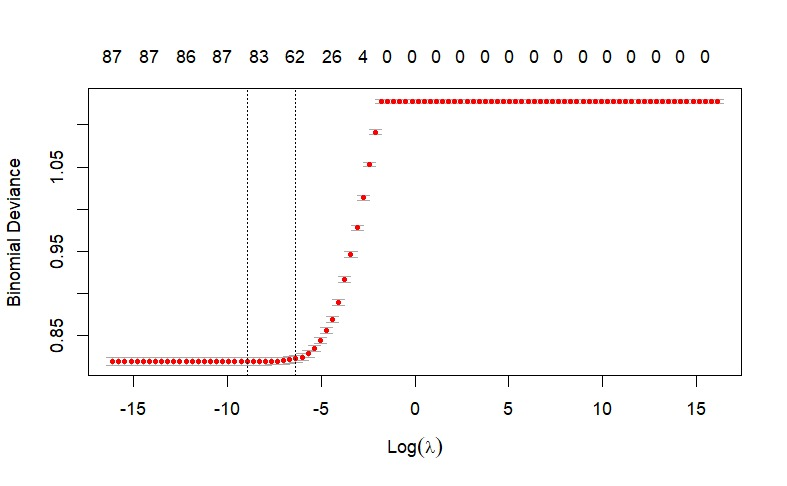
1. **Lasso Regression Model**

The R function we used for lasso regression cv.glmnet() and this function is part of the glmnet library in R. The estimated training and generalization performance for the lasso regression model is 0.842 and 0.8398 respectively.

To estimate the generalization performance, we used a simple train/validation split and used 70% train - 30% validation data configuration with 5 folds (k=5). We tuned the lambda (λ) hyperparameter. We used **100** values in the range of **10-7 to 107**. The line numbers of the R code are: **1213-1233**

The best set of features used for lasso regression is as mentioned below:

(accommodates, bedrooms, beds, one\_bed\_per\_person, cancellation\_policy, has\_cleaning\_fee, price, property\_category, bed\_category, bathrooms, charges\_for\_extra, host\_acceptance, host\_response, city\_name, host\_is\_superhost, is\_business\_travel\_ready, instant\_bookable, host\_identity\_verified, has\_min\_nights, require\_guest\_phone\_verification, require\_guest\_profile\_picture, requires\_license, room\_type, has\_security\_deposit, is\_location\_exact, guests\_included, host\_response\_time, offlinegovernmentid, governmentid, number\_of\_amenities, room\_type, has\_hotel\_license, has\_min\_weeks, availablity\_30\_perc, availablity\_60\_perc, availablity\_90\_perc, availablity\_365\_perc, has\_house\_rules, has\_weekly\_price, has\_monthly\_price, has\_access\_info, bathrooms\_per\_person, host\_and\_property\_same\_neighborhood, maximum\_nights, cleaning\_fee\_less\_than\_130, total\_years\_service, first\_review\_since, number\_of\_id\_verification\_options, maximum\_weeks, description\_length, avg\_price\_per\_night, price\_for\_extra, total\_price, cleaning\_fee, security\_deposit, name\_length, price\_groups, nearest\_attr\_dist, nearest\_attr\_count, host\_gender)



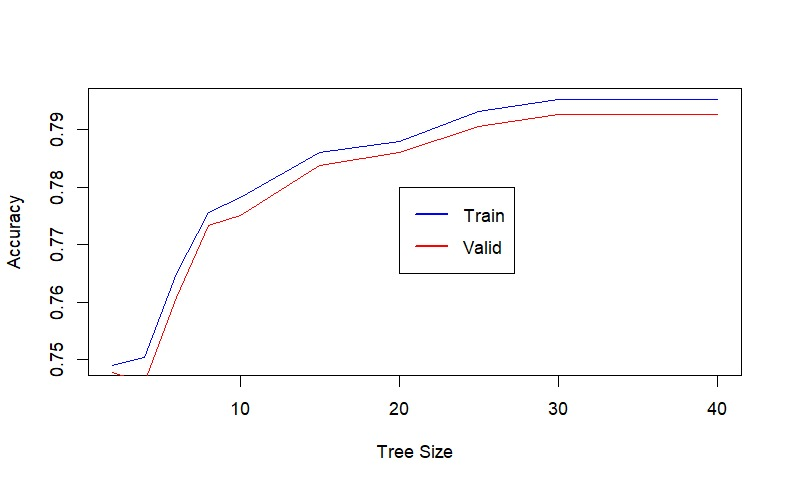
1. **Decision Tree Model**

The R function we used for the decision tree is tree.control(), tree(), prune.tree() and this function is part of the tree and ROCR library in R. The estimated training and generalization performance for the pruned decision tree is 0.8237 and 0.819 respectively.

To estimate the generalization performance, we used a simple train/validation split and used 70% train - 30% validation data configuration. We tuned the tree size hyperparameter. We used **11** values between **2 to 40** which are random values. The line numbers of the R code are: **843-892**

The best set of features used for the decision tree model is as mentioned below:

(accommodates, bedrooms, beds, one\_bed\_per\_person, cancellation\_policy, has\_cleaning\_fee, price, property\_category, bed\_category, bathrooms, charges\_for\_extra, host\_acceptance, host\_response, city\_name, host\_is\_superhost, is\_business\_travel\_ready, instant\_bookable, host\_identity\_verified, has\_min\_nights, require\_guest\_phone\_verification, require\_guest\_profile\_picture, requires\_license, room\_type, has\_security\_deposit, is\_location\_exact, guests\_included, host\_response\_time, offlinegovernmentid, governmentid, number\_of\_amenities, room\_type, has\_hotel\_license, has\_min\_weeks, availablity\_30\_perc, availablity\_60\_perc, availablity\_90\_perc, availablity\_365\_perc, has\_house\_rules, has\_weekly\_price, has\_monthly\_price, has\_access\_info, bathrooms\_per\_person, host\_and\_property\_same\_neighborhood, maximum\_nights, cleaning\_fee\_less\_than\_130, total\_years\_service, first\_review\_since, number\_of\_id\_verification\_options, maximum\_weeks, description\_length, avg\_price\_per\_night, price\_for\_extra, total\_price, cleaning\_fee, security\_deposit, name\_length, price\_groups, nearest\_attr\_dist, nearest\_attr\_count, host\_gender)



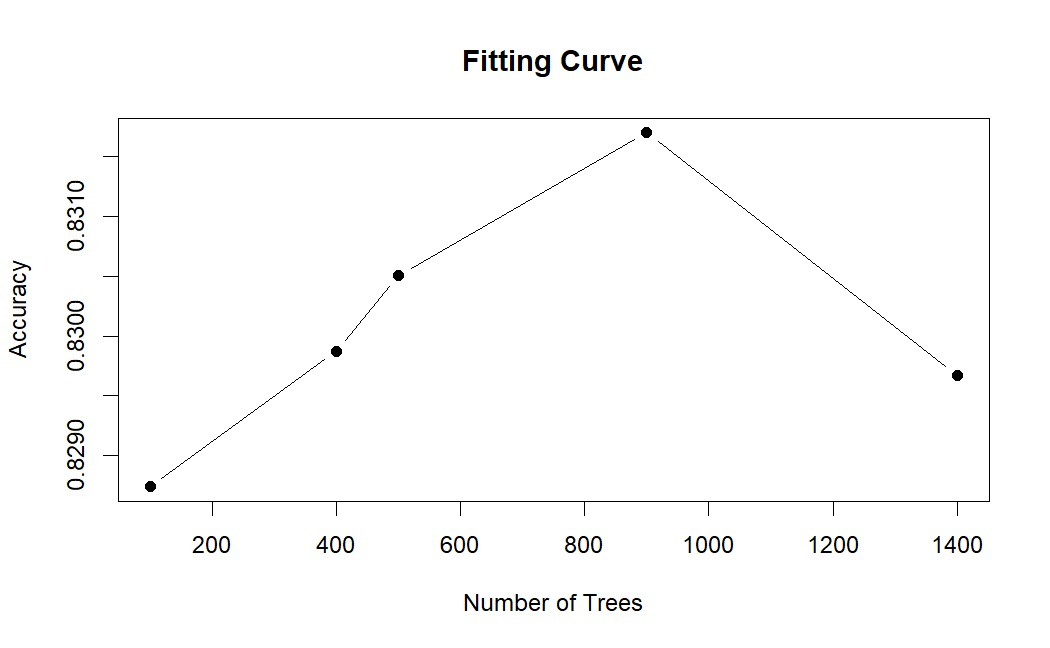
1. **Random Forest Model**

The R function we used for the random forest is ranger() and this function is part of the randomforest library in R. The estimated training and generalization performance for the random forest model is 0.9986 and 0.8819 respectively.

To estimate the generalization performance, we used a simple train/validation split and used 70% train - 30% validation data configuration. We tuned the mtry and number of trees hyperparameter. We used **all** values from **5 to 15** andthe number of trees from 500 to 1000 which are random values. The line numbers of the R code are: **1239-1248**

The best set of features used for the random forest model is as mentioned below:

(accommodates, bedrooms, beds, one\_bed\_per\_person, cancellation\_policy, has\_cleaning\_fee, price, property\_category, bed\_category, bathrooms, charges\_for\_extra, host\_acceptance, host\_response, city\_name, host\_is\_superhost, is\_business\_travel\_ready, instant\_bookable, host\_identity\_verified, has\_min\_nights, require\_guest\_phone\_verification, require\_guest\_profile\_picture, requires\_license, room\_type, has\_security\_deposit, is\_location\_exact, guests\_included, host\_response\_time, offlinegovernmentid, governmentid, number\_of\_amenities, room\_type, has\_hotel\_license, has\_min\_weeks, availablity\_30\_perc, availablity\_60\_perc, availablity\_90\_perc, availablity\_365\_perc, has\_house\_rules, has\_weekly\_price, has\_monthly\_price, has\_access\_info, bathrooms\_per\_person, host\_and\_property\_same\_neighborhood, maximum\_nights, cleaning\_fee\_less\_than\_130, total\_years\_service, first\_review\_since, number\_of\_id\_verification\_options, maximum\_weeks, description\_length, avg\_price\_per\_night, price\_for\_extra, total\_price, cleaning\_fee, security\_deposit, name\_length, price\_groups, nearest\_attr\_dist, nearest\_attr\_count, host\_gender)



**Section 5: Reflection/takeaways**

1) What did your group do well?

Our group was able to successfully preprocess and clean the data, perform feature selection, and train several models to predict the high booking rate for Airbnb listings. We also implemented a robust validation strategy and selected the best-performing model for submission in the contest. We collaborated effectively and communicated regularly, allowing us to divide the workload and make progress efficiently.

2) What were the main challenges?

One of the main challenges we faced was dealing with the large number of features in the dataset. We had to carefully choose which features to include and apply different feature selection techniques to reduce the dimensionality of the data. Additionally, some features had a high degree of missing values, which required us to impute or drop those instances.

Another challenge was achieving high performance on the test set, which had a different distribution of instances than the training set. We had to carefully balance the bias-variance trade-off and avoid overfitting on the training set.

3) What would your group have done differently if you could start the project over again?

If we could start the project over again, we would have focused more on exploratory data analysis to gain deeper insights into the data and the relationship between the features and the target variable. This would have helped us identify more effective feature engineering and selection techniques, and potentially discover new features that we missed during the initial preprocessing phase.

4) What would you do if you had another few months to work on the project?

If we had more time, we would experiment with more advanced models such as ensemble methods, deep learning, or gradient boosting to improve our predictions. We would also explore more advanced feature engineering techniques, such as creating new variables from combinations of existing variables or using external data sources to enrich the dataset.

Additionally, we would investigate the effect of hyperparameters on model performance and use more advanced hyperparameter optimization techniques, such as Bayesian optimization, to find the optimal hyperparameters for each model.

5) What advice do you have for a group starting this project next year?

We recommend investing time in exploratory data analysis to gain a deeper understanding of the data and identify effective feature engineering and selection techniques.

It's also crucial to implement a robust validation strategy, including cross-validation and hyperparameter tuning, to avoid overfitting and maximize model performance on the test set.

Finally, we recommend collaborating effectively and communicating regularly to ensure that everyone is on the same page and to facilitate progress efficiently.