UNIVERSITY OF MARYLAND
ROBERT H. SMITH SCHOOL OF BUSINESS

# BUDT 758T
# Final Project Report
# Spring 2024

## Section 1: Team member names and contributions

- ***Gunjan Suriya:*** She was primarily responsible for the tasks of data cleaning of variables; text mining and sentiment analysis; creating derived binary variables for amenities and host verification variables and model training and prediction coding for logistic, LASSO, Ridge, Random Forest, Decision Trees and XGBoost models. She also performed hyperparameter tuning; report writing and handled a few weekly submissions. She was proactive in initiating regular project meetups and helped in task delegation.
- ***Nimisha Sharma :*** She handled various aspects of the project, including data cleaning, text mining, model implementation and tested model variations to tune hyperparameters, report writing, and generating graphs for Section 3. In addition to this, she extracted insights from textual data, implemented analytical models, and contributed to the comprehensive reporting of findings.
- ***Rajat Marathe :*** He was primarily responsible for tasks such as data cleaning, text mining, derived variable creation for amenities and host verifications, and coding for LASSO and Ridge Models. He also handled fitting curves for the LASSO model, training and validation accuracy coding for Ridge and LASSO models, and overall code compilation, including the creation of final CSV files.
- ***Tanmay Sakharkar :*** He focused on data cleaning for categorical variables and implemented XGBoost and Random Forest Models. He also performed text mining, developed cutoff value code, and created cutoff value versus performance metrics graphs. Additionally he crafted training and validation accuracy graphs and code for both the RF and XGBoost models.
- ***Tanvi Murumkar :*** She contributed to data cleaning of various text and numeric variables. In addition to this, she helped with text mining. Furthermore, she worked on report writing and editing, along with maintaining an overall coherence and clarity with the report. She also produced some of the graphs in Section 3 which explain the overall characteristic of the various features.

## Section 2: Business Understanding

In the dynamic hospitality industry, exceptional guest experiences are fundamental to success. With the emergence of home-sharing platforms such as Airbnb.com, hosts face the challenge of consistently meeting the expectations of their guests. However, amidst the multitude of factors influencing guest satisfaction, discerning the key drivers of perfect rating scores remain ever-changing.

Predictive modeling offers a promising avenue for tackling this challenge. This research aims to predict the likelihood of an Airbnb listing achieving a perfect rating score by harnessing advanced algorithms, feature engineering and data mining techniques. The report aims to uncover the underlying patterns that distinguish outstanding stays from the rest through a comprehensive analysis of various features.

This research report is an exploration into the predictive modeling landscape with regards to Airbnb rating scores. This study's methodologies, findings, and implications, seek to provide valuable insights for hosts, property management companies, and other stakeholders within the industry.

For individual hosts, the model enables proactive identification of areas for improvement and optimization of listing strategies. By understanding the factors that contribute to perfect rating scores, hosts can prioritize investments in property upgrades, refine communication strategies, and tailor guest experiences to meet and exceed guest expectations.

Property management companies can leverage the model to optimize operations across their portfolio of listings. By identifying high-performing properties and replicating successful strategies, management companies can drive excellence in guest experiences, improve overall satisfaction levels, and maximize returns on investment.

Additionally, the predictive model offers valuable insights for real estate investors seeking to enter or expand their presence in the short-term rental market. By evaluating the factors affecting achieving perfect rating scores, investors can make informed decisions regarding property acquisitions, renovations, and marketing strategies, ultimately maximizing profitability.

The implications of the research extend beyond individual hosts and businesses to encompass the broader hospitality industry. By facilitating the delivery of exceptional guest experiences, the predictive model contributes to the dynamic home-sharing sector with prospects of opening new avenues of making consumer expectations a success.

To conclude the report offers a thorough examination of the state of predictive modeling when it comes to Airbnb rating scores. By providing actionable insights and strategic recommendations, it aims to empower stakeholders to optimize guest satisfaction, drive business growth, and succeed in the dynamic and competitive hospitality market.

## Section 3: Data Understanding and Data Preparation

### *1) Feature Information Table*

| ID | Feature Name | Feature Type | Brief Description | Changes in Features | R Code Line Numbers |
|---|---|---|---|---|---|
| 1 | monthly_price | Numeric | price to rent the listing for a month | Replaced NA values by multiplying price & mean of the column = monthly price/ price | 122 - 132 |
| 2 | minimum_nights | Numeric | minimum nights you can book the listing | No changes made | |
| 3 | price | Numeric | price to rent the listing for one night | Replaced NA values with mean price | 114 |
| 4 | security_deposit | Numeric | Amount of security deposit required to rent the listing | Categorized into 5 categories based on value: Small, Medium, Moderate, Large and Unstated | 104 - 109 |
| 5 | bathrooms | Numeric | number of bathrooms | Missing values replaced with column mean | 89 |
| 6 | bedrooms | Numeric | number of bedrooms | Missing values replaced with column mean | 90 |
| 7 | beds | Numeric | number of beds | Missing values replaced with column mean | 91 |
| 8 | accommodates | Numeric | how many guests can stay in the listing | Missing values replaced with column mean | 88 |
| 9 | square_feet | Numeric | Listing's square feet | Missing values replaced with column mean | 103 |
| 10 | weekly_price | Numeric | Listing rent for a week | Replaced NA values by multiplying price & mean of the column = weekly price/ price | 131-132 |
| 11 | availability_30 | Numeric | Available days to rent in the next 30 days | No changes made | |
| 12 | availability_365 | Numeric | Available days to rent in the next 365 days | No changes made | |

| | | | | | |
|---|---|---|---|---|---|
| 13 | host_listings_ count | Numeric | Total listings that the host has | No changes made | |
| 14 | availability_60 | Numeric | Available days to rent in the next 60 days | No changes made | |
| 15 | availability_90 | Numeric | Available days to rent in the next 90 days | No changes made | |
| 16 | host_acceptance_ rate | Numeric | percent of stay requests that the host accepts | Test: Values divided by 100 to adjust in range (0,1). Train and test: converted from % to numeric | 38-41 |
| 17 | host_response_ rate | Numeric | Stay request response percent by host | Replaced NA values with 0 | 92 |
| 18 | host_total_ listings_count | Numeric | Total listings the host has ever had | Replaced NA values with 0 | 93 |
| 19 | cleaning_fee | Numeric | how much cleaning fee is charged | Replaced NA values with 0 | 87 |
| 20 | extra_people | Numeric | additional charge for extra people in the rental | No changes made | |
| 21 | guests_included | Numeric | how many guests are included in the price of the rental | No changes made | |
| 22 | maximum_nights | Numeric | maximum nights you can book the listing for | No changes made | |
| 23 | latitude | Numeric | the number of degrees west of the prime meridian | No changes made | |
| 24 | longitude | Numeric | the number of degrees north of the equator | No changes made | |

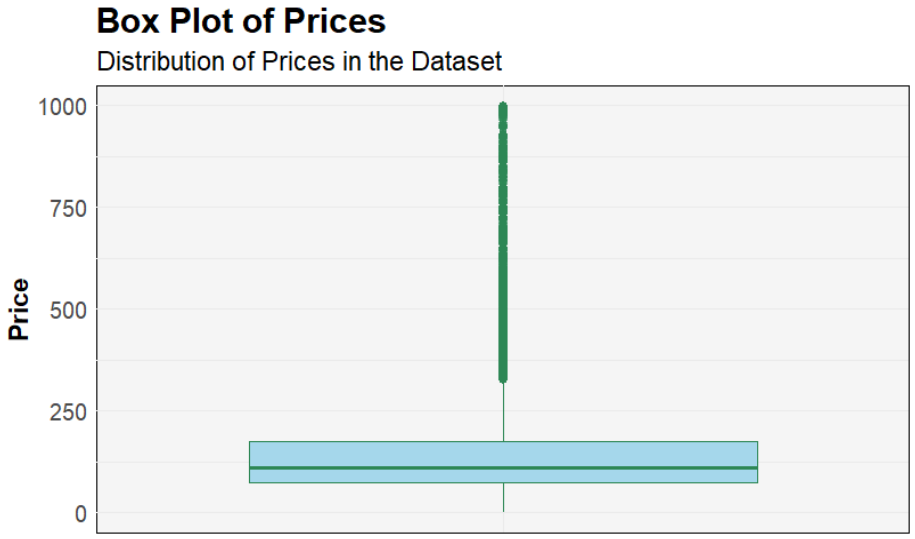
| | | | | | |
|---|---|---|---|---|---|
| 25 | transit | Text | free text field describing nearby transit options for the listing | preprocesses the text data in train_clean, tokenizes it with specific stop words : “stop_words_transit <- c("your", "s", "get", "also", "im", "go", "take"),” creates a vocabulary, generates a document-term matrix, and incorporates it into a combined dtm matrix. | 299-618 |
| 26 | neighborhood_ group | categoric al | coarse-grained neighborhood name of the listing | Missing values have been replaced with values from city | 102 |
| 27 | bed_type | categoric al | description of the bed | one-hot encoding for categorical features dataset, converting them into factors, setting reference levels and then converting factor levels to numeric values, effectively creating dummy variables for each category. | 162-222 |
| 28 | license | categoric al | whether the host has a hotelier license (t) or not (f) | Missing values have been replaced with values from no license | 95 |
| 29 | city | categoric al | the actual city that the listing is in | Missing values have been replaced with values from market | 100 |
| 30 | host_neighbourho od | categoric al | more fine-grained location of the host | Missing values have been replaced with values from host_location | 85 |
| 31 | jurisdiction_name s | categoric al | the legal jurisdiction that the listing falls under | Missing values have been replaced with values from smart_location | 86 |
| 32 | market | categoric al | Airbnb's definition of the market that the listing competes in | Missing values have been replaced with values from city | 99 |
| 33 | amenities | categoric al list | list of amenities available in the listing | Encoded categorical list features into binary matrices | 225-280 |
| 34 | host_verifications | categoric al list | ways the host has verified their identity | Encoded categorical list features into binary matrices | 225-280 |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | |

In addition, similar data cleaning methodologies to that of the “transit” feature have been used to other text features that are : description, host_about, access, features, neighborhood_overview, house_rules, space. Furthermore, the categorical list features have been cleaned through one-hot encoding. The features are as follows : room_type, host_response_time, property_type and cancellation_policy.

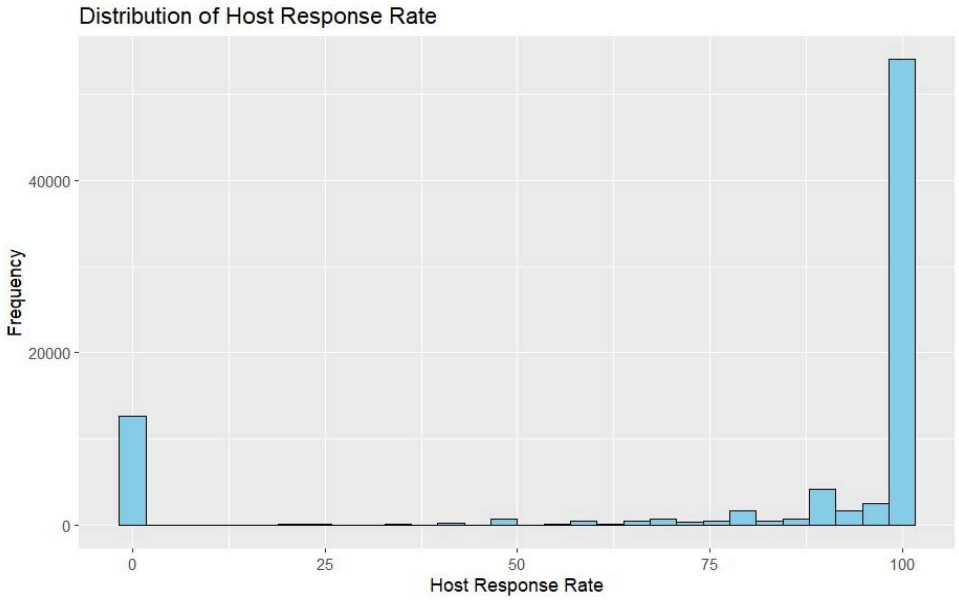
2) ***Graphs or tables demonstrating useful or interesting insights regarding features in the dataset.***

**Box Plot of Prices**



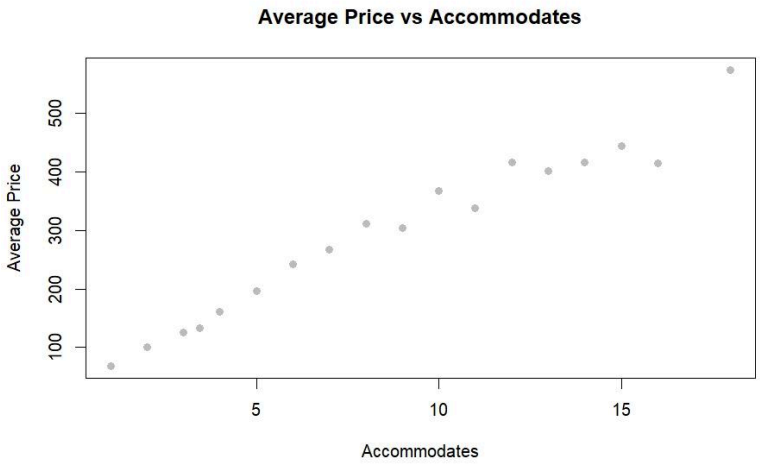

Majority of the prices range from $ 150 to around $ 225, as can be seen on the box plot. However, the dataset has many outliers ranging from $ 300 that go up to a $ 1000. This explains the right skewed nature of the price distribution.

**The distribution of host response rate**



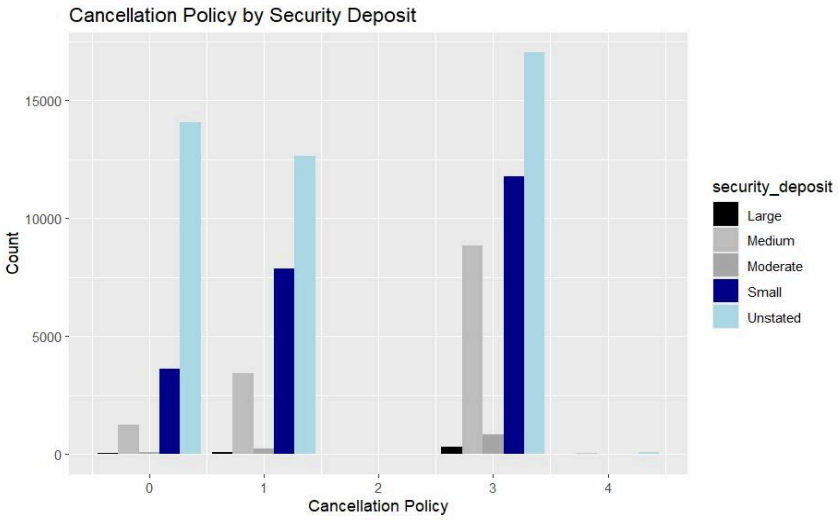

This histogram shows an interesting finding of the second highest frequency corresponding to 0% response rate. Usually, hosts are anticipated to provide prompt response to any inquiries to attract more bookings and provide better service.However, it appears that around 13000 hosts are not meeting these expectations which could potentially influence the guest perceptions further affecting the perfect rating score.

**Average Price vs Accommodates**



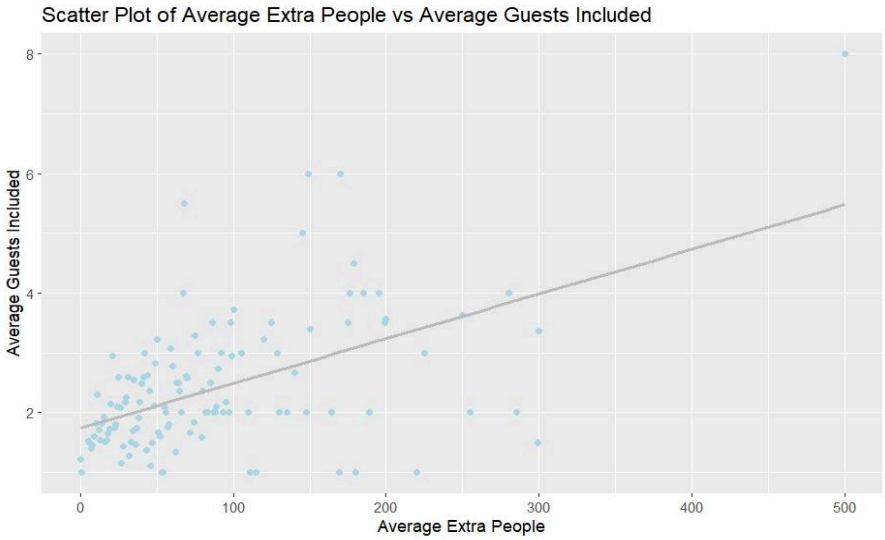

The scatter plot shows a positive correlation in between the average price and the number of people that it accommodates. It was interesting to see the gradual rise with larger accommodations suggesting a proportional relationship between these variables.

## Cancellation Policy by Security Deposit




The unstated security deposits have the highest cancellations for all cancellation policies. Apart from this, there is a negative correlation between the cancellation policy and security deposit indicating that people might be willing to lose smaller security deposits.

## Average Extra people vs Average guests included

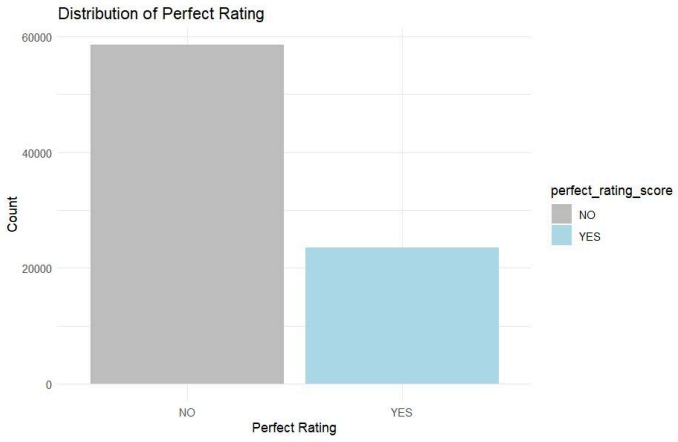



The rentals with an average of 1-4 guests seem to be clustered in 100$ range.However, the range from 100 to 300 we see varied distribution of the average guest included.This shows a diversity in additional charge for extra people in the rental which could be the influence of varying property types, location and amenities.

## Price vs Number of Bedrooms, Bathrooms & Beds

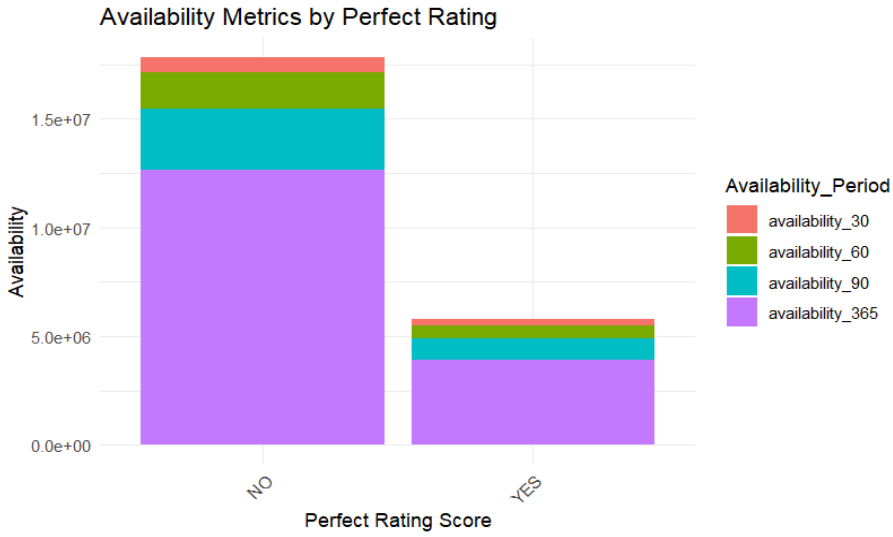

The above graph shows the relationship between price and bedrooms, beds and bathrooms. This shows a very surprising relationship between the variables. While one may assume that it would be a straightforward positive relationship, we can see that the number of beds has a very high count in comparison to the price. As price increases the number of beds does not show the same consistency as to when the price was low. Similarly, the relationship between price and bedrooms isn't very consistent.An increase in price shows a significant decrease in the number of bedrooms. On the other hand, the number of bathrooms has a consistent relationship with price.

## Distribution of perfect rating score



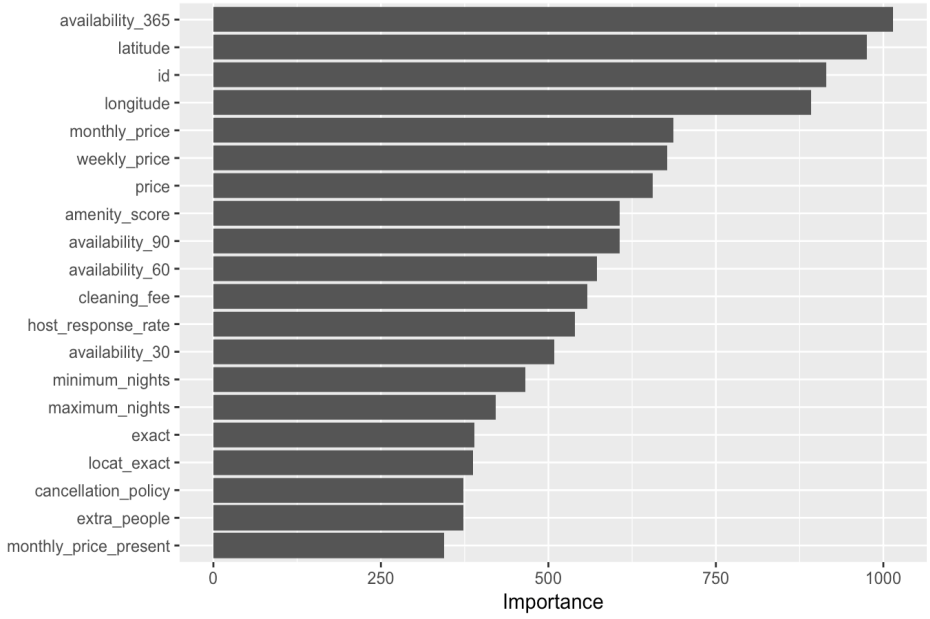

The perfect rating score has a negative count of around 59000 and positive count of 22000 which shows a significant imbalance. As the target variable, it is crucial to identify the positive instances whilst being able to minimize the misclassification of negative instances.

**Average Metrics by Perfect Rating**




Due to the significant imbalance in the perfect rating score, it is hard to determine which availability variables hold the highest significance. It is better to focus on the positive relationship rather than the distribution of the availability variables.

**Feature Importance plot (for final model)**




The importance plot shows that the latitude and longitude rank second and third in importance among the features. This is interesting because we expected these variables to be significant only if we had done a geospatial analysis, especially considering that we used the "smart location" variable, which involved generating jurisdiction names through text mining. It's noteworthy that

both the monthly and weekly price gained significance despite being derived from the price variable. Surprisingly, the availability of the listing for the entire year is a significant factor which can help understand customer perspective while rating airbnbs.

**Additional Data Preparation steps**

During the project's initial phases, extensive Exploratory Data Analysis (EDA) was conducted to comprehend the dataset thoroughly. In order to avoid chances of overfitting, we kept 10,000 rows (held-out data) of train data separately and treated it as unseen data for the model by not using it for training purposes. For the remaining 82,067 rows, we split it into a 80-20 training-validation split.

Owing to some general understanding of the data, we extracted unique amenities and host verification methods from the respective columns and turned them into individual columns assigning them 1-0 values based on their presence in the listing data point. This helped us increase our TPR.

By delving into every aspect of the dataset, we gained a deeper understanding of its underlying patterns and dynamics, ultimately enhancing the accuracy and effectiveness of our predictive model.

## Section 4: Evaluation and Modeling

Winning Model: 'Random Forest' was our winning model. To estimate the generalization performance for this model, 10,000 rows from the dataset were held-out to be used as validation data which helped avoid the issue of overfitting and also helped tune hyperparameters like cutoff values, mtry and num.trees. The TFIDF dataframe was generated using text mining with the train data and test data. Numeric columns along with the TFIDF columns were incorporated for this model as it showed the highest TPR compared to the FPR threshold of 10%.
The line numbers for model training and estimated its generalized performance are lines 1057 to 1157.
The final model was trained using the entire dataset provided and the predictions and classifications were generated. Changes in the cutoff value were made based on a loop code which would test all cutoff values for the predictions ranging from 0.1 to 0.9 with 0.01 intervals. This loop would then plot a graph showcasing the FPR, Accuracy, and TPR for all cutoff values. The code also included a section wherein we would give a desired FPR and the Optimal cutoff would be stored in a variable. The cutoff value stored in the variable is then used as the cutoff to make classifications.
Different values for num.trees were tried and tested for hyperparameter tuning (500, 1000, 1500 ,2000, 2500), The model started with 500 and ended with 2500 num.trees which provided the highest TPR, and since the number of trees was adequately high, the generalization performance improved significantly.

**Model Details** :

- Type of Model : Random Forest
- R library : Ranger
- R Function:

```
tr_x <- tr_x[, common_cols]
va_x <- va_x[, common_cols]

rf_mod <- ranger(x = tr_x, y = as.factor(tr_y),
                 mtry = 50, num.trees = 2500,
                 importance = "impurity", probability = TRUE)

# Predict probabilities of class 1 (YES) for the validation set
preds_rf <- predict(rf_mod, data = va_x, type = "response")$predictions[, 2]

# Convert probabilities to class labels (YES or NO)
class_preds <- ifelse(preds_rf > 0.464, "YES", "NO")
valid_classifications <- as.factor(class_preds)
```

- Training performance of random forest model:
  - TPR: 0.9928237

    FPR: 0.0002926544
- Generalization performance of random forest model on validation data:
  - TPR: 0.3016186

    - FPR: 0.09157865
- Generalization performance of random forest model on holdout data:
    - TPR: 0.3201104
    - FPR: 0.09252218
- Hyperparameters tuned: mtry= 50, num. trees =2500, threshold = 0.464
- Important features: availability_365, id, latitude, longitude, monthly_price, weekly_price, price, amenity score, availability_60, availability_90

| **Model Type** | **Training performance** | **Generalization performance on validation data** | **Generalization performance on holdout data** | **Best-performing features** |
|---|---|---|---|---|
| Logistic Regression | TPR: 0.1348<br>FPR: 0.0417 | TPR: 0.1302<br>FPR: 0.0455 | TPR: 0.1294<br>FPR: 0.0451 | Interaction terms: accommodates*bathrooms, bed_type*price; Derived terms: amenity score and host_verification_score |
| Ridge | TPR: 0.2337<br>FPR: 0.0649 | TPR: 0.2355<br>FPR: 0.0694 | TPR: 0.2318<br>FPR: 0.0629 | Amenities such as : Wide clearance to shower and toilet, Paid parking off premises, Free Parking on Street, Paid Parking Off Premises, Cleaning before checkout |
| LASSO | TPR: 0.2318<br>FPR: 0.0651 | TPR: 0.2373<br>FPR: 0.0693 | TPR: 0.2387<br>FPR: 0.0630 | Amenities such as : Wide clearance to shower and toilet, Pocket wifi, Accessible-height bed, Paid parking off premises, Free Parking on Street |
| Decision Trees | TPR: 0.7739<br>FPR: 0.5157 | TPR: 0.7779<br>FPR: 0.5118 | TPR: 0.2929<br>FPR: 0.1621 | Amenities such as : Wide clearance to shower and toilet, Paid Parking Off Premises |
| Random Forest | TPR: 0.9928<br>FPR: 0.0002 | TPR: 0.3016<br>FPR: 0.09158 | TPR: 0.3201<br>FPR: 0.0925 | availability_365,id,latitude,longitude,monthly_price,weekly_price,price,amenity score,availability_60, availability_90 |
| XGBoost | TPR: 0.8456<br>FPR: 0.0297 | TPR: 0.8473<br>FPR: 0.0283 | TPR: 0.3639<br>FPR: 0.1249 | availability_365, latitude, id, longitude, monthly_price, weekly_price, price |

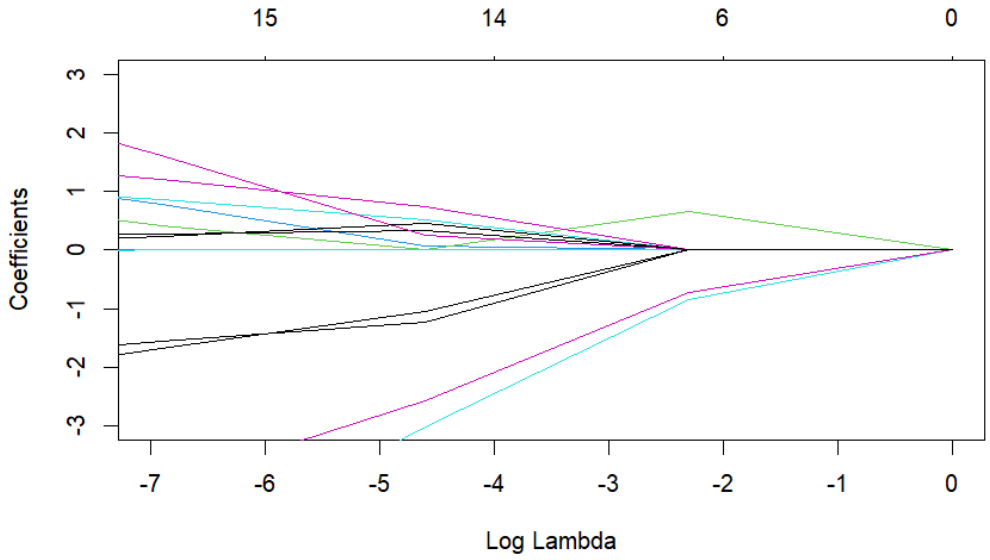
| **Model type** | **R-function/libraries** | **Hyperparameters tuned** | **Line Numbers in code** |
|---|---|---|---|
| Logistic Regression | Libraries used : caret, tidyverse<br><br>Function: See Code 1 | None | 212 - 311 |
| Ridge | Libraries used: glmnet, vip<br><br>Function:<br><br>ridge_model <- cv.glmnet(tr_ridge, as.factor(tr_y_ridge), alpha = 0, lambda = grid, family = "binomial") | Lambda values: Grid search over lambda values from 10^10 to 10^-2 with 100 values in between. alpha = 0 | 383 - 475 |
| LASSO | Libraries used: glmnet, vip<br><br>Function:<br><br>lasso_model <- cv.glmnet(tr_lasso, as.factor(tr_y_lasso), alpha = 1, lambda = grid, family = "binomial") | Lambda values: Grid search over lambda values from 10^10 to 10^-2 with 100 values in between. alpha = 1 | 487 - 567 |
| Decision Trees | Library used: tree<br><br>Function:<br><br>default_tree <- tree( perfect_rating_score~ ., data = data.frame(tr_tree))<br>pruned_tree_5=prune.tree(default_tree, best = 3) | Ran the model for different values of 'best' from 2 to 100 (prune.tree function) | 597 - 669 |
| Random Forest | Library used: Ranger<br><br>Function: rf_mod <- ranger(x = tr_x, y = as.factor(tr_y_full),<br>mtry = 50, num.trees = 2500,<br>importance = "impurity",<br>probability = TRUE) | Ran the model with different tree values (1000, 1500, 2000, 2500, 3000, 3500) | 1057 - 1157<br>1519 - 1544 |

| | | | |
|---|---|---|---|
| XGBoost | Library used: xgboost<br><br>Function:<br><br>xgb_model <- xgboost(data = as.matrix(tr_xg), label = tr_y_xg, nrounds = 1000, objective = "binary:logistic") | Ran the model for different nrounds ranging from 500 to 1500 in breaks of 500 | 1605 - 1699 |

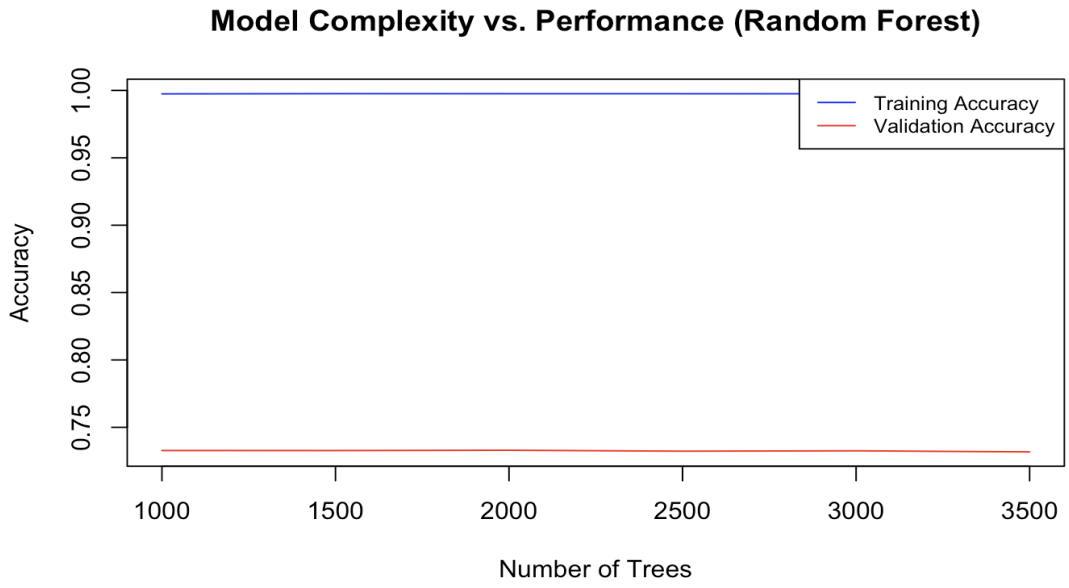
Code 1: logmodel <- glm(perfect_rating_score~ host_response_rate + host_listings_count + price + room_type + bedrooms + beds + cancellation_policy + amenity_score + cleaning_fee + host_verification_score + guests_included + availability_30 + accommodates*bathrooms + extra_people + maximum_nights + bed_type*price + availability_365 + minimum_nights , data = tr_log, family = "binomial")

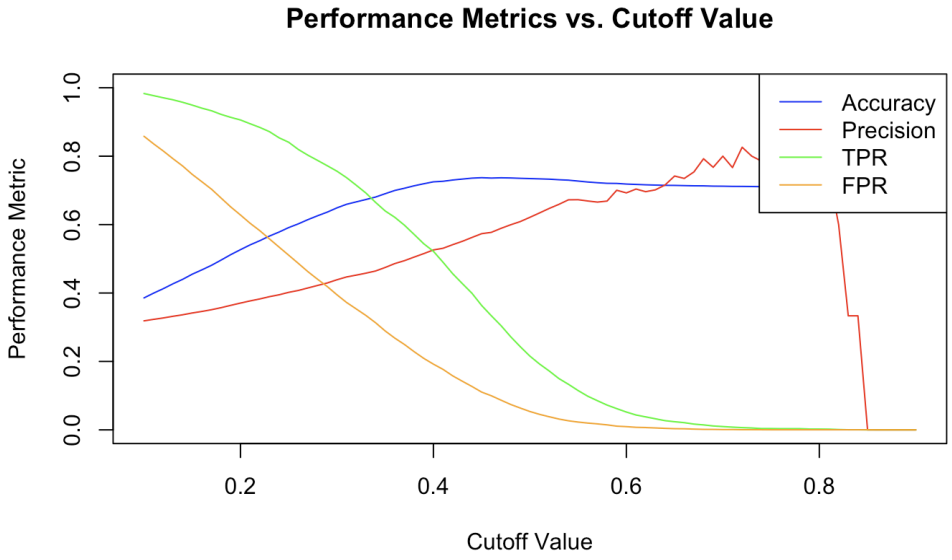
**Fitting Curves:**

**LASSO model:**

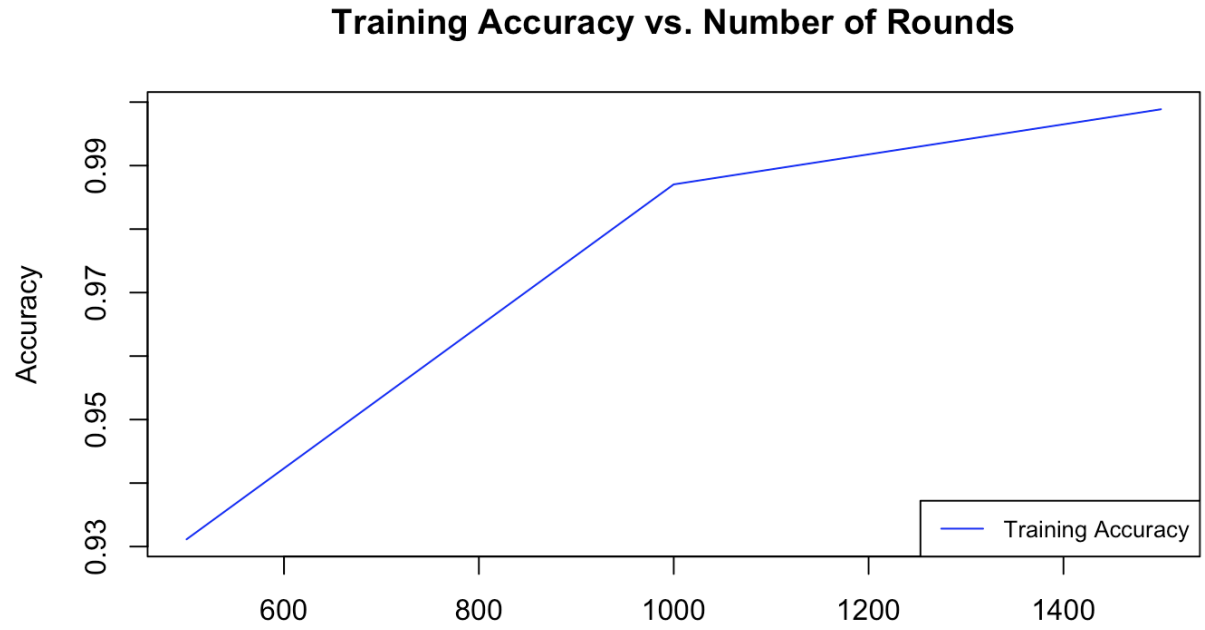


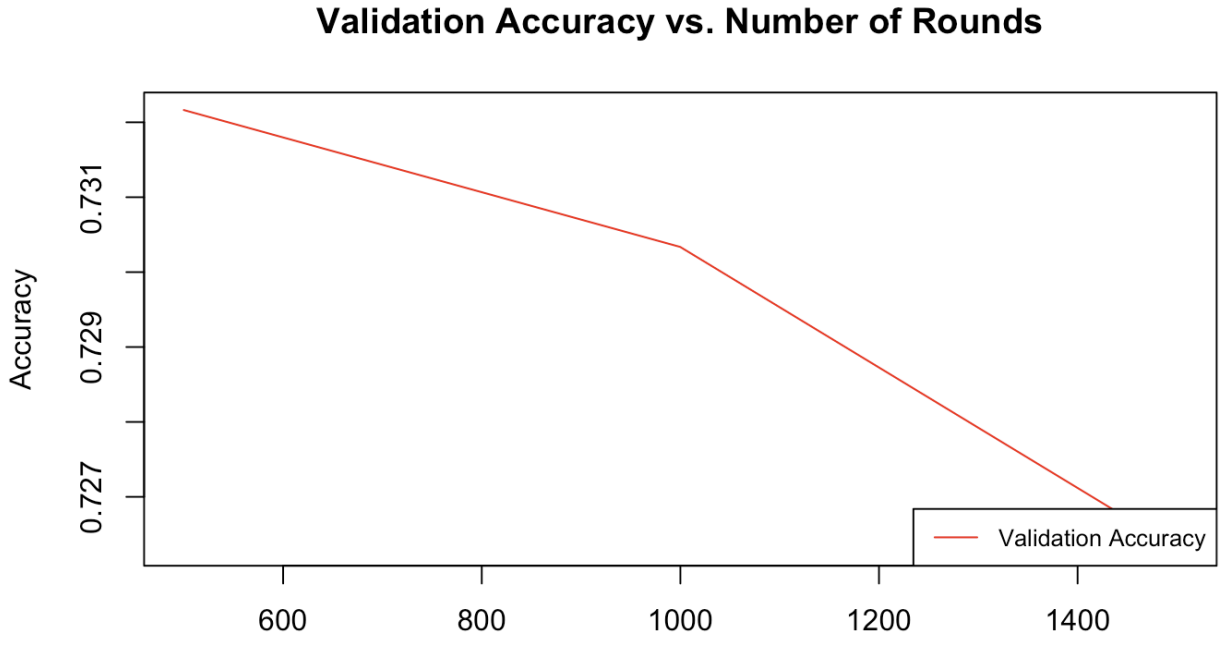
**Random Forest Model:**



**XGBoost Model:**



## Section 5: Reflection/takeaways

Throughout the journey of this project, our group experienced numerous highs and lows, including the difficulties that came with having members of the group work together for the first time. This gave us a huge opportunity for reflection and takeaways. The group effectively managed tasks within the designated timeline, starting from the contest submission to the final report. We were able to keep each other motivated and remained committed, thus enabling us to maintain a consistent position on the leaderboard.The group had clarity on tasks and maintained the right direction. We managed to coordinate well through an open communication channel; helping us allocate tasks to utilize our strengths as team members.

The main challenge we faced was when we reached a stage where we had tried many different models but our true positive rate was not increasing. To rectify this, we had to go back to feature engineering and evaluate our choices of data cleaning. This was very time consuming and often led to no results but we tried to visualize data from a fresh perspective which helped us modify a few features and increased our model performance in later submissions. Apart from this, accommodating everyone's time schedules sometimes became a hassle but we overcame it through regular meetings after and in between classes to push for the best results in the project.

If given the chance to do the project differently, we would designate more time to data cleaning and feature engineering. Whilst trying out different variations of the model, we noticed that adjusting different thresholds or altering random splits was not necessarily helping us achieve a higher TPR while aiming for a FPR lower than 10%.

If given the opportunity to work on the project for another few months, we would like to work with modifications in data cleaning, feature engineering and text mining. Due to the time constraints, we were unable to try out more variations hence we would like to direct our focus towards deriving new variables such as host friendliness based on sentiment analysis. Other than the variations, we also would like to explore other supervised learning models and see what could be the possible difference in the performance. During our check in with Professor Jahani, he had mentioned the possibility of joining a compilation of the AirBnB reviews dataset. With more time, we would have also taken the time to explore these datasets and observe their effect on our TPR and FPR. Intrigued by the latitude and longitude variables, we would be interested in looking at spatial interpolation in Random Forest models as we do not know much about georeferencing through this classifier. We would also rely more on our own knowledge about what factors could affect the target variable and take decisions accordingly instead of relying more on models for feature selection.

For the future students who would be working on this project, our advice would be to maintain team spirit and keep encouraging each other because as one tries different tedious models, it can get frustrating and may not always yield the result we'd like. Maintaining patience and a strong sense of collaboration along with dedicated brainstorming will be beneficial. Creating a good timeline while coordinating schedules will reduce conflicts and help meet deadlines.The dataset has a set of numerous features and feature engineering will add new variables but reducing the features or using lasso for feature selection will help in not overfitting the model.Taking more time to do text mining can help in reducing down to the important words and will be beneficial

utilizing the sentiment analysis. This is a great learning opportunity which will help strengthen concepts but also help learn different aspects of teamwork. It is also a great opportunity to learn from your group members strengths and a variety of perspectives.