UNIVERSITY OF MARYLAND
ROBERT H. SMITH SCHOOL OF BUSINESS

BUDT 758T

Final Project Report

Spring 2023

1.0 Team Members and their Contributions

Name	Contributions
Aditi Patel	- Conducted exploratory data analysis (EDA) - Preprocessed the dataset, handling missing values and transforming variables - Explored feature selection techniques - Implemented and evaluated different models - Assisted in model comparison and selection
Meet Doshi	- Assisted in the incorporation of external dataset regarding income - Conducted data merging and preprocessing for external data - Participated in feature selection tasks - Assisted in determining parameters for text mining - Contributed to model evaluation and comparison

	- Contributed to the project report writing and documentation
Yash Makadia	- Participated in feature selection tasks - Assisted in determining parameters for text mining - Conducted advanced modeling techniques, such as Decision Tree and Random Forest. - Contributed to the project report writing and documentation - Assisted in data cleaning
Vidit Vaywala	- Conducted feature engineering tasks - Explored external datasets for potential integration - Assisted in preprocessing and cleaning the data - Assisted in model evaluation and comparison - Contributed to the project report writing and documentation
Deepthi Rao	- Assisted in the preprocessing and cleaning of data - Participated in model evaluation and comparison - Contributed to the project report writing and documentation

INDEX

2.0 Executive Summary

Our project focused on predicting the likelihood of an Airbnb listing receiving a perfect rating score. Airbnb.com is a widely recognized home-sharing platform that connects homeowners with renters worldwide. By analyzing a comprehensive dataset, we aimed to determine whether a listing would achieve a 100% perfect rating score or not.

To achieve our objectives, our team aimed to generate binary predictions with a high True Positive Rate (TPR) while keeping the False Positive Rate (FPR) below 10%. This approach allowed us to effectively identify listings with perfect rating scores while minimizing false positives.

The dataset provided for analysis included a range of informative columns, such as property access, accommodation details, amenities, availability metrics, cancellation policies, location information, pricing, and textual descriptions. These attributes offered valuable insights into the factors influencing perfect rating scores.

Throughout the project, we applied advanced data mining techniques and utilized feature engineering to enhance the performance of our predictive models. By leveraging machine learning algorithms, we evaluated multiple models to determine their accuracy in predicting perfect rating scores. Our primary goal was to identify the most effective model for this specific task.

After conducting comprehensive testing and analysis, we identified that the Ranger model exhibited the highest accuracy in predicting the success of Airbnb listings in terms of achieving a perfect rating score. The Ranger algorithm demonstrated exceptional performance and delivered precise predictions.

Our predictive model incorporated a variety of features, including availability metrics, property details, host-related information, cancellation policies, and other relevant attributes. These

factors played a significant role in determining the probability of a listing receiving a perfect rating score.

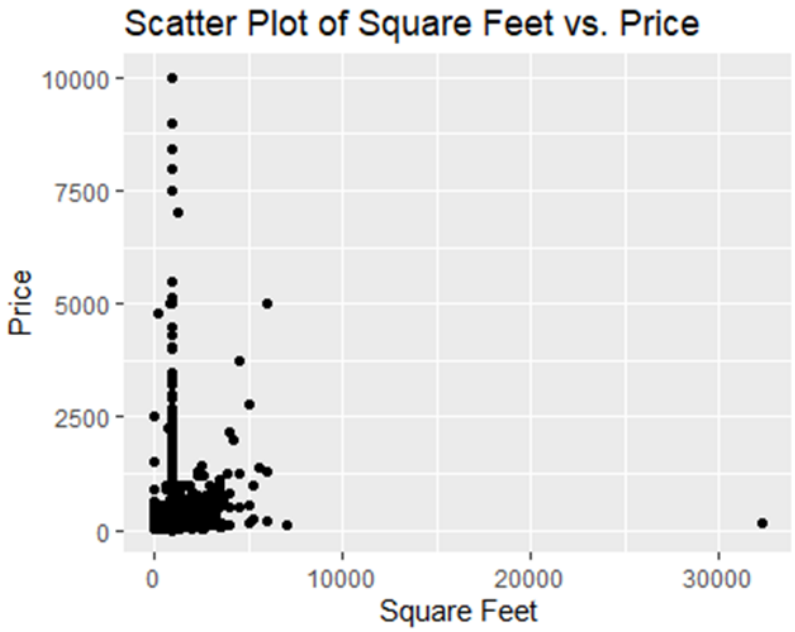
The Ranger model achieved an impressive accuracy rate of 76.79% on the test set with a TPR of ~42% and an FPR of ~8.6%, providing valuable insights into the factors that contribute to perfect rating scores on Airbnb.com. By adjusting the identified variables, our model offers hosts the opportunity to enhance their chances of achieving a perfect rating score. This, in turn, improves the overall guest experience and increases the potential for attracting more renters.

In conclusion, our data mining project successfully addressed the challenge of predicting perfect rating scores for Airbnb listings. By utilizing the Ranger model and incorporating key attributes, we achieved a high accuracy rate. Our findings and recommendations can be instrumental in assisting hosts to optimize their listings and provide exceptional experiences to Airbnb guests.

3.0 Data Understanding and Data Preparation

3.1 Data Exploration

1. **Square Feet - Histogram**



The presence of outliers in the scatter plot, with prices exceeding $5,000 per night despite square footage below 2,500, holds significant business value as it highlights properties commanding exceptionally high prices relative to their size.

Moreover, the sole observation with a square footage greater than 10,000 should be excluded from further analysis to ensure accuracy. From a technical standpoint, this outlier removal mitigates potential data distortions.

Lastly, the concentration of most observations near the origin, with prices below $2,500 and square footage below 5,000, underscores the importance of analyzing this variable. Understanding the dynamics within this range is vital for both business and technical analysis, as it provides valuable insights into market behavior and aids decision-making. Thus, the outliers,

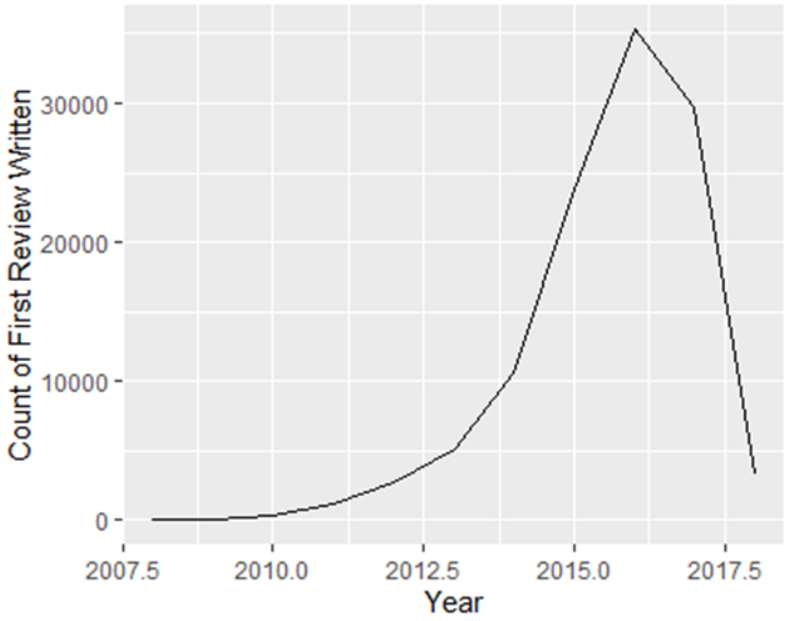
exclusion of extreme values, and the concentration near the origin make the price and square footage variable crucial for subsequent analysis.

2. **Price Category by Count - Bar Graph**

The bar chart shows a significant difference in the number of listings across price categories. The "Medium" price category has the most entries, indicating that it is a popular choice among users. The "High" price category, on the other hand, has the lowest count, indicating that it may cater to a more niche market.

The bar chart shows that there are many observations in both the "Low" and "Very High" price categories. This means that the market has a significant number of listings that cater to both budget-conscious individuals looking for low-cost options and those looking for high-end accommodations.

3. No Listings over the Year

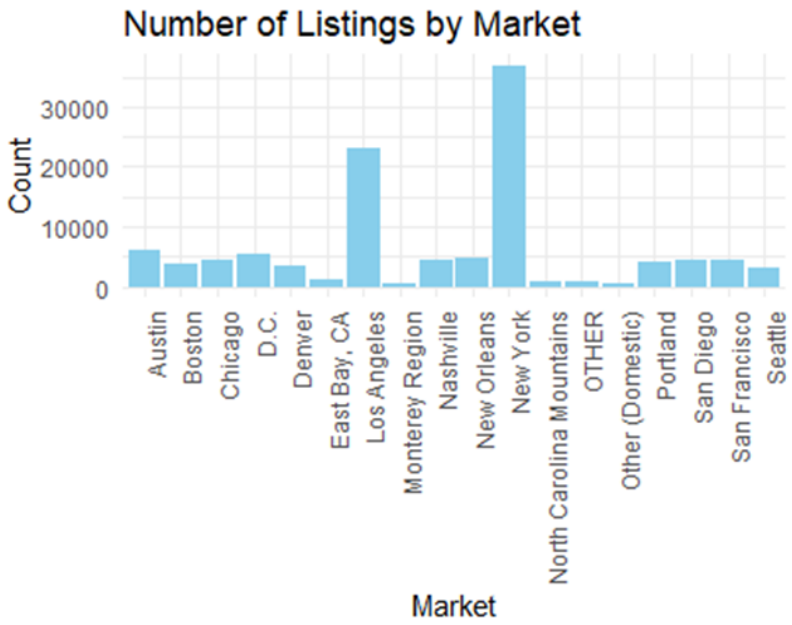


The line chart shows that the number of first reviews increased significantly in 2015 and 2016, with around 25,000 and 35,000 reviews, respectively. This suggests that a growing number of users are joining the platform and leaving their first reviews during this time period.

The sharp drop in first reviews in 2017, with only 15,000 reviews, followed by another drop to 5,000 reviews in 2018, indicates a significant drop in user engagement or a potential shift in user behavior. This decline could be attributed to a variety of factors, including changes in platform popularity, changes in user preferences, or external market conditions.

The declining trend in first reviews from 2017 to 2018 raises concerns about the platform's overall growth and user activity. It may be worthwhile to investigate the causes of this decline in order to identify potential areas for improvement, such as improving the onboarding process or implementing strategies to encourage users to leave reviews.

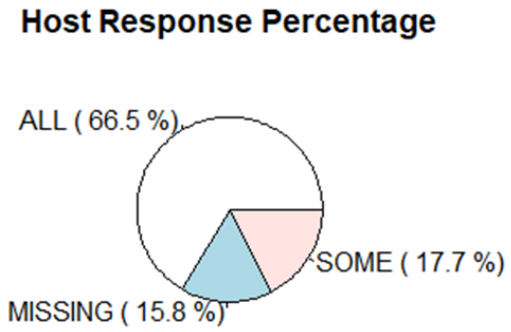
4. **Market Analysis - Bar chart**



The bar chart shows that New York has the most listings of any market, with approximately 35,000 listings. This indicates that Airbnb accommodations are prevalent in New York, making it a popular travel destination.

Los Angeles is also notable for having many listings, with approximately 22,000 properties. This suggests that Los Angeles is another popular destination for Airbnb stays, drawing many visitors. Places like North Carolina Mountains have barely around 300 listings as there are not many houses in that area.

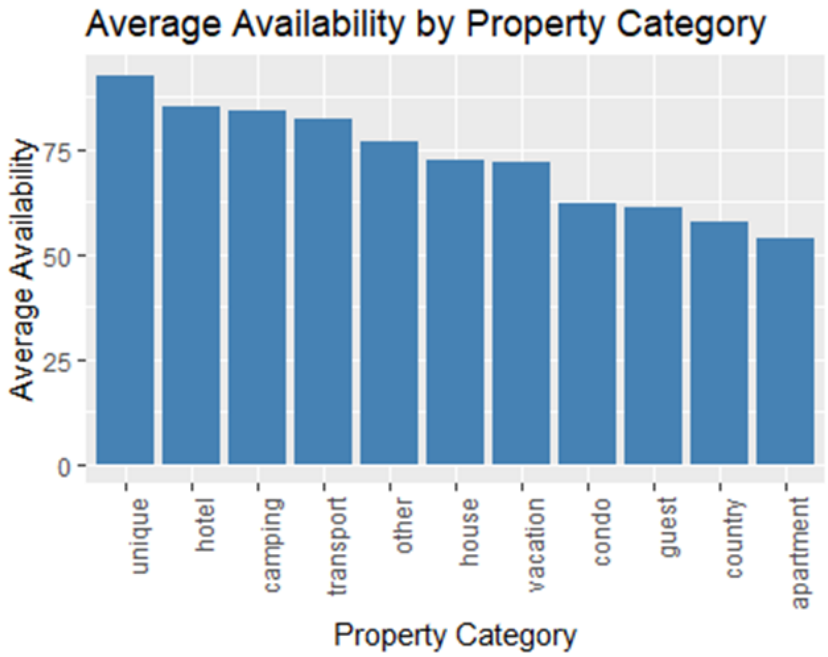
5. **Pie chart for Host Response Rate Analysis**



Most host responses (66.5%) fall into the category of "All," indicating that a significant portion of hosts responds to inquiries or messages from guests in a timely manner. This suggests that hosts communicate in a timely and effective manner, which can have a positive impact on the overall guest experience.

A significant proportion of host responses (17.7%) are labeled as "Some," implying that some hosts may not consistently respond to all inquiries or messages within the time frame specified. This variation in response rates could be attributed to host preferences or other factors. It emphasizes the importance of taking host responsiveness into account as a potential predictor when developing a predictive model for the perfect rating score.

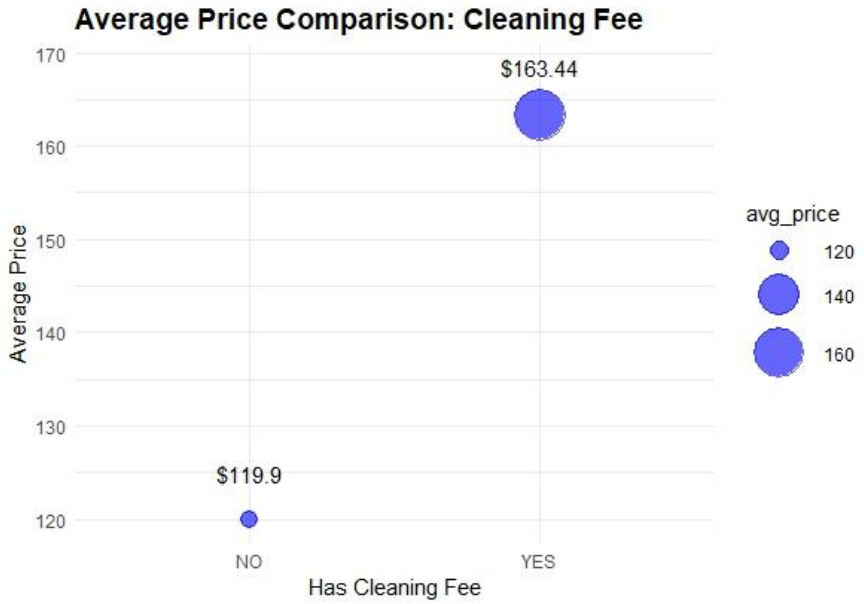
6. **Average Availability in days for different property categories – Bar Chart**



When compared to other property categories, unique properties have the highest average availability. This suggests that guests may prefer to stay in unusual accommodations such as treehouses, yurts, or boats. These types of properties frequently provide a unique and memorable experience for travelers, which could explain their increased availability.

Apartments, country living, and guest houses have lower average availability when compared to other property types. This could imply that these types of accommodations are more popular with travelers, resulting in higher demand and, as a result, lower availability. Apartments are a popular choice for longer stays or for travelers looking for a home-like setting. Country living and guest houses may appeal to guests seeking a peaceful and rural retreat.

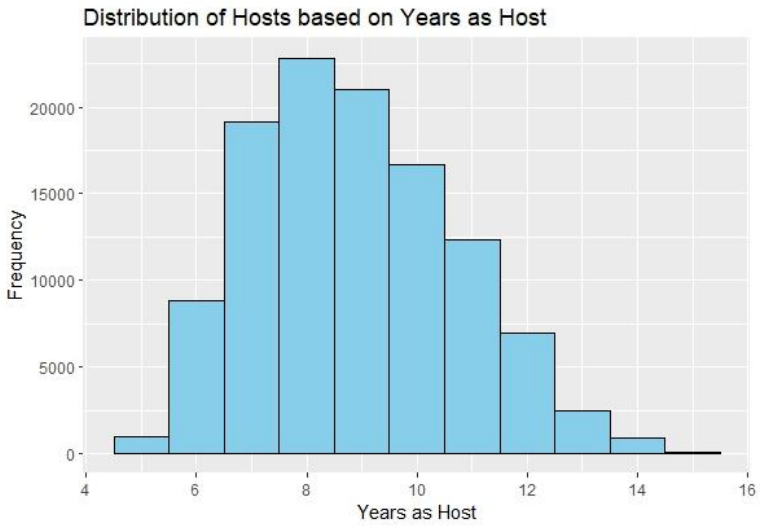
7. **Bubble Chart showing the average price for properties with and without cleaning price**



Cleaning fees increase the average price of a property to $163.44. This implies that these properties may provide additional services or amenities, or that they may be subjected to more thorough cleaning and maintenance. Guests may perceive these properties to be of higher quality and are willing to pay a higher price for the added convenience and cleanliness.

Properties without cleaning fees, on the other hand, have a lower average price of $119.9. This could imply that these properties take a more self-service approach, with guests expected to clean their own accommodations before checking out. These properties may be appealing to budget-conscious travelers who value cost-cutting options.

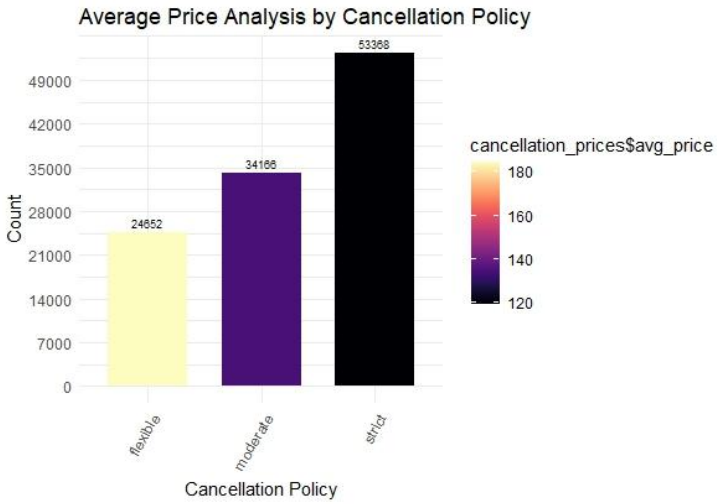
8. **Distribution of Number of Hosts based on the number of years of hosting the property on Airbnb**



We can see that the pattern follows close to a normal distribution with the most hosts lying in the 7-11 years of hosting.

This variable will help us in the model to determine whether newer or experienced hosts are more likely to get a perfect rating score.

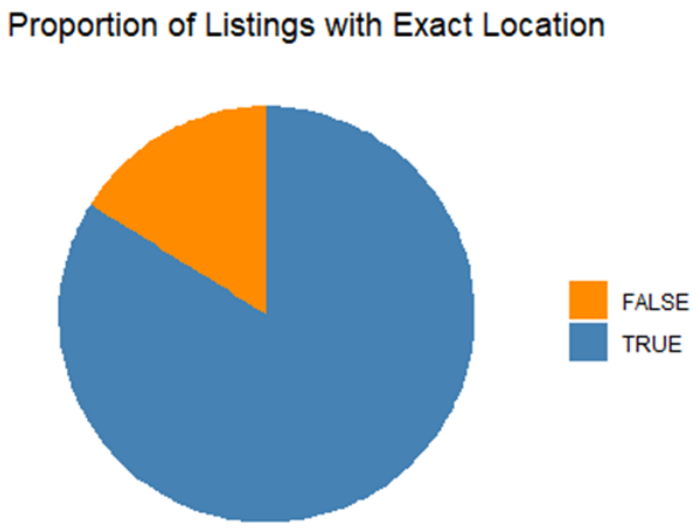
9. **Average Price Analysis by Cancellation Policy and Count of Listings**



We can see that most of the listings (53368) have a cancellation policy as Strict and the lowest average price of around $120 whereas the listings with the flexible cancellation policy have the lowest count (24652) but with an average price of $180.

This is consistent with the real-life situation that hosts that are able to keep the cancellation policy as flexible, they would have to charge a higher price in order to cover the costs if the customer cancels at the last minute.

10. Proportion of Listings with the exact location



We can see that almost 80% of locations have an exact location in the dataset.

This suggests that travelers usually look for places that have the exact location so they can plan their trips in a more efficient manner.

3.2 Feature Engineering

During the initial data exploration phase, we used the data to train our logistic model. We tried to understand the various columns in the dataset, and we noticed that there were many columns that could not be used directly but were important for the model to work efficiently. This led to the need to reprocess the dataset. We cleaned the parameters in the current dataset and created new parameters from the existing ones.

Below is our table of the final list of the parameters that were there and created eventually to be used in our project.

ID	Feature Name	Description	R Code Line Numbers
Existing Parameters			
1	security_deposit	Security deposit amount in the listings	38-39
2	host_since	Date when the host joined Airbnb	40
3	first_review	Date of the first review received by the property	47
4	price	Price of the listing	50
5	property_category	Categorization of the property type	55-66
6	cancellation_policy	Categorization of cancellation policy	79-81
7	cleaning_fee	Cleaning fee charged for the listing	82-83
8	square_feet	Square footage of the property	85
9	beds	Number of beds in the listing	86
10	extra_people	Additional fee for extra people	87
11	guests_included	Number of guests included in the booking	89
12	host_identity_verified	Indicator variable for host identity verification	90
13	host_is_superhost	Indicator variable for host superhost status	91

14	host_response_rate	Response rate of the host	100
15	instant_bookable	Indicator variable for instant bookable listings	102
16	is_location_exact	Indicator variable for precise location	103
17	require_guest_phone_verification	Indicator variable for guest phone verification	104
18	requires_license	Indicator variable for the requirement of a license	105
19	room_type	Categorization of the room type	106
20	host_acceptance_rate	Acceptance rate of booking requests by the host	107-108
21	bathrooms	number of bathrooms in the listing	109-110
22	market	Airbnb's definition of the market that the listing competes in	111
New Parameters			
23	years_as_host	Number of years the host has been active on Airbnb	41-42
24	availability_avg	Average availability of the property over different time periods	43-46
25	property_first_used	Number of years the property has been used since the first review	48-49
26	availability_30_ratio	Ratio of availability for the next 30 days	51
27	average_nights_available	Average number of nights available for booking	52
28	booking_flexibility	Categorization of booking flexibility based on minimum and maximum nights	53-54
29	price_per_person	Price per person for the listing	67
30	ppp_ind	Indicator variable for price per person higher than the category median	68-69
31	price_category	Categorization of price range	75-78

32	has_cleaning_fee	Indicator variable for the presence of cleaning fee	84
33	charges_for_extra	Indicator variable for the presence of charges for extra people	88
34	has_min_nights	Indicator variable for minimum nights requirement	92
35	host_response	Categorization of host response rate	101
36	host_acceptance	Categorization of host acceptance rate	108

Date Columns

Years as Host and Property First Used: These variables provide insights into the host and property's experience and longevity, respectively. They were derived by calculating the number of years since the host's registration and the property's first review, respectively. These metrics help assess the reliability, reputation, and maintenance of the host and property.

Availability Categories

Availability Average and Availability 30 Ratio: These variables represent the availability of the property. Availability Average is the average availability over different time periods, while the Availability 30 Ratio indicates the availability for the next 30 days. These metrics assist in understanding how frequently the property is available for booking and provide information about short-term availability.

Booking Categories

Booking Flexibility: This variable categorizes the booking flexibility based on the minimum and maximum nights allowed. It helps users understand the property's booking policies, whether they are flexible or not.

Price Categories

Price per Person, Price per Person Indicator, and Price Category: These variables focus on the pricing aspect of the listings. Price per Person calculates the price per person for the listing, allowing fair comparisons between different properties. The price per Person Indicator identifies if the price per person is higher than the median within its property category. Price Category categorizes the price range into different categories, enabling users to filter properties based on their budget preferences.

Other Categories

Has Cleaning Fee and Charges for Extra People: These indicators provide information about additional fees associated with the property. Has Cleaning Fee identifies if the property has a cleaning fee, while Charges for Extra People indicates whether extra charges apply for accommodating more guests.

Has Minimum Nights: This indicator variable informs whether the property has a minimum night requirement. It helps users filter properties based on their desired length of stay.

Host Categories

Host Response and Host Acceptance: These variables categorize the host response rate and host acceptance rate, respectively. They provide insights into the host's communication and interaction with guests, helping users assess the host's responsiveness and acceptance level.

A good number of columns had NAs and information that was not meaningful, this was especially so for categorical columns. To remove them one can either remove the entire row or impute with appropriate values. Removal of rows leads to loss of information in other columns. For this reason, we decided to impute values like ‘None’ for categorical nulls, and 0 for nulls in rewards.

Dummy Variable Encoding

In order to effectively utilize certain models such as Ranger, Decision Trees, and Logistic Regression, it was necessary to transform the predictor variables into binary numerical values (1s

and 0s). To achieve this, we applied dummy variable encoding to convert the predictors into the desired format, enabling the models to perform optimally.

Incorporation of External Dataset

During the feature selection process, we explored the potential benefits of incorporating an external dataset that provided information about the income of individuals in different zip codes across the USA. The aim was to leverage this additional feature in models such as Ranger, Decision Trees, and Logistic Regression. To utilize the external dataset, we calculated the mean average income of individuals based on their zip codes. However, after thorough experimentation, we observed that incorporating this external dataset did not lead to improved model accuracy. As a result, we made the decision not to include it in our feature selection process. Despite our efforts to incorporate the external dataset, we determined that its inclusion did not yield significant improvements in the performance of our models. Therefore, we decided not to proceed with its integration into our feature selection pipeline.

4.0 Evaluation and Modeling

4.1 Ranger Random Forest - WINNER MODEL

The Ranger Model is a random forest machine learning algorithm used to predict whether an Airbnb listing will receive a perfect rating score. The model is trained on 32 variables, including price, booking flexibility, availability ratios, property information, host characteristics, and more. The model uses 800 trees with a maximum of 20 variables considered at each split and impurity-based variable importance.

The estimated training and generalization performance of the model is evaluated using a validation dataset, where the accuracy of the model is calculated based on the number of correctly predicted perfect rating scores. The model achieved an accuracy of around 77%, with a true positive rate (TPR) of 42% and a false positive rate (FPR) of 8.6%. This was better than the Baseline Model which had a had accuracy of 70.33%. Other models which include Logistic

Regression, Ridge, Lasso, and Decision Trees were also able to achieve a better accuracy and performance as compared to the Baseline Model but Ranger performing the best in all of them.

The decision to choose the Ranger Model as the winning model was based on its performance compared to other models in the project. The Ranger Model had the highest accuracy and TPR compared to other models, indicating its ability to correctly identify listings with perfect rating scores. Additionally, the model's interpretability and variable importance analysis provided insights into the factors that contribute to a perfect rating, making it a useful tool for Airbnb hosts looking to improve their listings.

Model function: ranger

Estimated Training Performance:
TPR: 0.922
FPR: 0.0010
Accuracy: 0.9765

Estimated Generalization Performance:
TPR: 0.4230
FPR: 0.0865
Accuracy: 0.7679

To evaluate the performance of our model, we utilize metrics such as accuracy, True Positive Rate (TPR), and False Positive Rate (FPR). These metrics are employed in the code to assess how well our model is performing.

Set of features used:

price_category,booking_flexibility,availability_30_ratio,square_feet,average_nights_available, property_first_used,availability_avg,years_as_host, accommodates, bedrooms, ppp_ind, beds, cancellation_policy, has_cleaning_fee, charges_for_extra, host_identity_verified, market,

host_is_superhost, host_response, instant_bookable, has_min_nights, price, guests_included, property_category, security_deposit, is_location_exact, host_acceptance, bathrooms, require_guest_phone_verification, requires_license, room_type
Line Numbers in R code: 302-338

List of Hyperparameters tuned and their values: mtry - 5, 6, 8, 10, 15, 20, 22
num.trees - 100, 500, 800, 1000, 1500, 2000

4.2 Logistic Regression

Logistic Regression is a popular and widely used model for binary classification. It models the relationship between the independent variables and the probability of the target variable. The logistic regression model in the code achieves comparable accuracy to Random Forest but slightly lower TPR and higher FPR. Logistic Regression is a simpler model compared to Random Forest, but it may struggle to capture complex relationships in the data. Nonetheless, it still performs reasonably well and can be a good choice if interpretability is a priority.

Model Function: glm

Estimated Training Performance:
TPR: 0.3934
FPR: 0.103
Accuracy: 0.7491

Estimated Generalization Performance:
TPR: 0.4006
FPR: 0.0994
Accuracy: 0.7522

To evaluate the performance of our model, we utilize metrics such as accuracy, True Positive Rate (TPR), and False Positive Rate (FPR). These metrics are employed in the code to assess how well our model is performing.
Set of features used:

price_category,booking_flexibility,availability_30_ratio,square_feet,average_nights_available, property_first_used,availability_avg,years_as_host, accommodates, bedrooms, ppp_ind, beds, cancellation_policy, has_cleaning_fee, charges_for_extra, host_identity_verified, market, host_is_superhost, host_response, instant_bookable, has_min_nights, price, guests_included, property_category,security_deposit, is_location_exact, host_acceptance, bathrooms, require_guest_phone_verification, requires_license, room_type

Line Numbers in R code: 342-376

4.3 Decision Trees

Decision Trees are intuitive models that partition the data based on a set of rules. They are prone to overfitting but can be controlled using parameters like maximum depth and minimum sample split. The decision tree model in the code achieves a relatively lower accuracy compared to Random Forest and Logistic Regression. It may not capture the complexity of the data as effectively and could be sensitive to overfitting. While Decision Trees have their merits, in this case, other models seem to offer better performance.

Model Function: rpart

Estimated Training Performance:
TPR: 0.3478
FPR: 0.0867
Accuracy: 0.7473

Estimated Generalization Performance:
TPR: 0.3319

FPR: 0.0933
Accuracy: 0.7361

To evaluate the performance of our model, we utilize metrics such as accuracy, True Positive Rate (TPR), and False Positive Rate (FPR). These metrics are employed in the code to assess how well our model is performing.

Set of features used:

price_category,booking_flexibility,availability_30_ratio,square_feet,average_nights_available, property_first_used,availability_avg,years_as_host, accommodates, bedrooms, ppp_ind, beds, cancellation_policy, has_cleaning_fee, charges_for_extra, host_identity_verified, market, host_is_superhost, host_response, instant_bookable, has_min_nights, price, guests_included, property_category,security_deposit, is_location_exact, host_acceptance, bathrooms, require_guest_phone_verification, requires_license, room_type

Line Numbers in R code: 379-414

List of Hyperparameters tuned and their values:
cp - 0.001, 0.0001, 0.0002, 0.0003, 0.0005
Maxdepth - 5, 6, 7, 8, 9, 10

4.4 Lasso Regression

Lasso Regression is another linear regression model that incorporates L1 regularization. It performs variable selection and forces some coefficients to zero, effectively performing feature selection. The lasso regression model in the code achieves a similar accuracy to Ridge Regression but slightly lower TPR and higher FPR. Like Ridge Regression, Lasso Regression is a linear model and may have limitations in capturing complex relationships. It could be a viable choice if feature selection is important, but other models seem to offer better overall performance.

Model Function: glmnet

Estimated Training Performance:
TPR: 0.3868
FPR: 0.1003
Accuracy: 0.7491

Estimated Generalization Performance:
TPR: 0.3956
FPR: 0.0963
Accuracy: 0.7529

To evaluate the performance of our model, we utilize metrics such as accuracy, True Positive Rate (TPR), and False Positive Rate (FPR). These metrics are employed in the code to assess how well our model is performing.

Set of features used:

Price_category,booking_flexibility,availability_30_ratio,square_feet,average_nights_available,property_first_used,availability_avg,years_as_host, accommodates, bedrooms, ppp_ind, beds, cancellation_policy, has_cleaning_fee, charges_for_extra, host_identity_verified, market, host_is_superhost, host_response, instant_bookable, has_min_nights, price, guests_included, property_category,security_deposit, is_location_exact, host_acceptance, bathrooms, require_guest_phone_verification, requires_license, room_type

Line Numbers in R code: 493-558

List of Hyperparameters tuned and their values: We did a grid search from 10^-7 to 10^7 to get the lambda with the highest accuracy and used that lambda value (0.0003430469) for Lasso.

4.5 Ridge Regression

Ridge Regression is a linear regression model that incorporates L2 regularization to handle multicollinearity and prevent overfitting. It performs variable selection and shrinks the coefficients of less influential variables. The ridge regression model in the code achieves a similar accuracy to Random Forest and Logistic Regression but slightly lower TPR and higher FPR. Ridge Regression is a linear model, and its performance may be limited compared to more flexible models like Random Forest. In this case, other models seem to offer better performance, but Ridge Regression can still be considered if interpretability and simplicity are key concerns.

Model Function: glmnet

Estimated Training Performance:
TPR: 0.3824
FPR: 0.0988
Accuracy: 0.7523

Estimated Generalization Performance:
TPR: 0.3901
FPR: 0.0948
Accuracy: 0.7523

To evaluate the performance of our model, we utilize metrics such as accuracy, True Positive Rate (TPR), and False Positive Rate (FPR). These metrics are employed in the code to assess how well our model is performing.

Set of features used:

price_category,booking_flexibility,availability_30_ratio,square_feet,average_nights_available, property_first_used,availability_avg,years_as_host, accommodates, bedrooms, ppp_ind, beds, cancellation_policy, has_cleaning_fee, charges_for_extra, host_identity_verified, market,

host_is_superhost, host_response, instant_bookable, has_min_nights, price, guests_included, property_category,security_deposit, is_location_exact, host_acceptance, bathrooms, require_guest_phone_verification, requires_license, room_type

Line Numbers in R code: 416-491

List of Hyperparameters tuned and their values: We did a grid search from 10^-7 to 10^7 to get the lambda with the highest accuracy and used that lambda value (0.001747528) for Ridge.

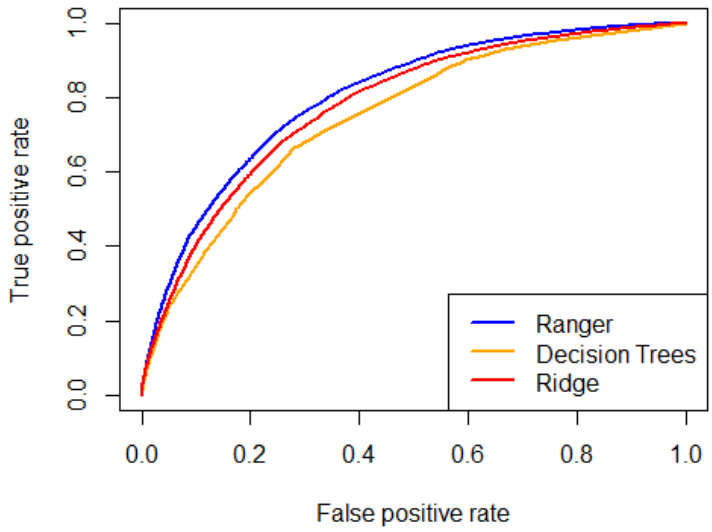
Other Libraries:

In addition to these specific libraries, the code also uses some common libraries such as tidyverse, caret, dplyr, pROC, Metrics, mlr, ggplot2, plotly, randomForest, glmnet, tidytext, and ranger for general data manipulation, visualization, and evaluation purposes.

4.6 Model Evaluation

4.6.1 ROC Curve

ROC Curve for Ranger, Decision Tree, and Ridge Regression:

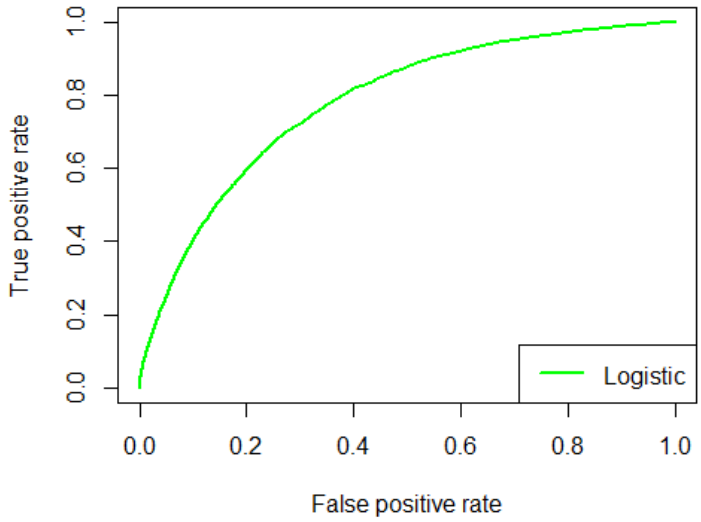


Line Numbers in R code: 572-590

The ROC curves depicted in the figure provide a visual representation of the trade-off between the true positive rate (TPR) and false positive rate (FPR) for different thresholds in the random forest, decision tree, and ridge classifiers. By examining the ROC curves, it becomes evident that they do not intersect with each other. This characteristic allows us to determine if one classifier outperforms the others by consistently achieving higher TPR values for all possible FPR values. Consequently, the area under the ROC curve, which indicates the overall performance of the classifier, is also expected to be greater.

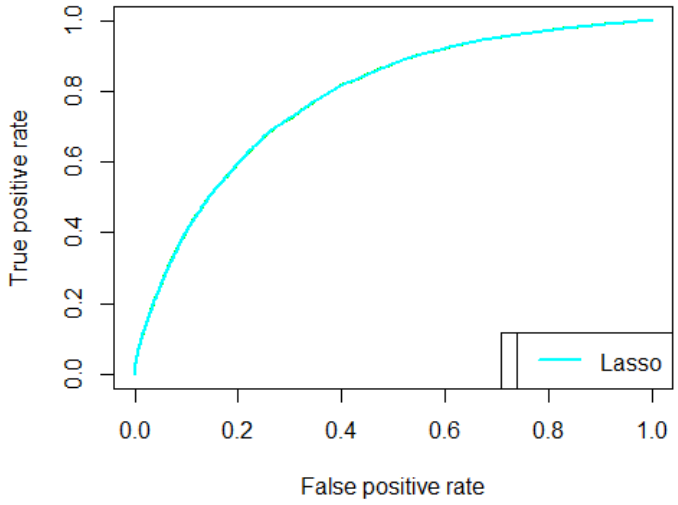
In our analysis, the Random Forest (Ranger) classifier demonstrates superior performance compared to the Lasso and Logistic Regression models, as evident from the ROC curves presented in other figures. The Random Forest classifier consistently exhibits higher TPR values, indicating its ability to accurately identify positive instances while maintaining a low rate of false positives. This performance advantage positions the Random Forest classifier as the top-performing model in our study.

Logistic Regression ROC Curve:



Line Numbers in R code: 560-565

Lasso Regression ROC Curve:

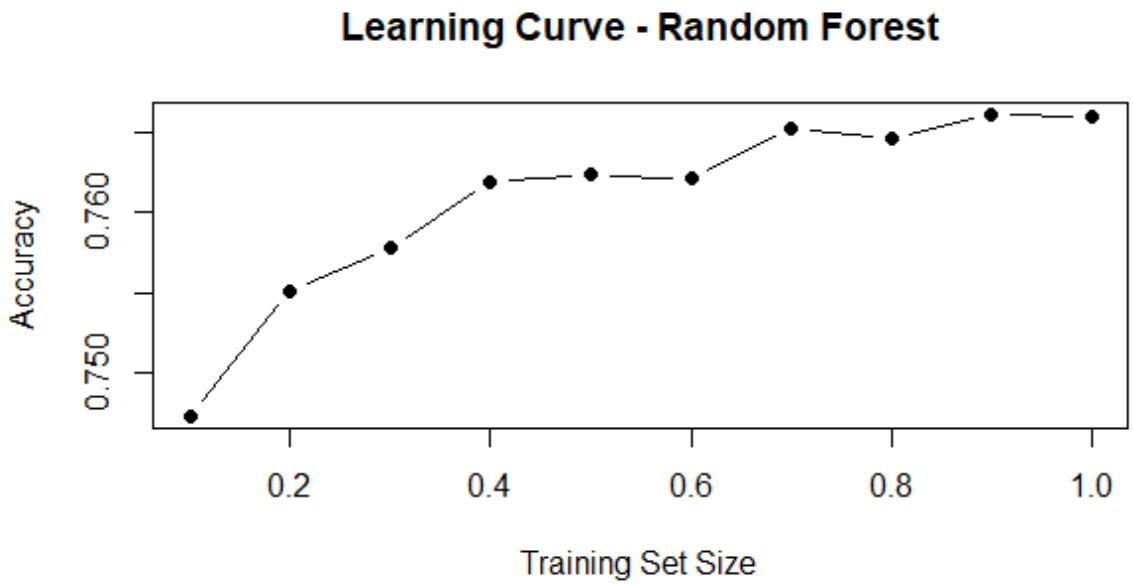


Line Numbers in R code: 566-571

4.6.2 Optimizing Cutoff Values for Enhanced Performance

To select the optimal cutoff value, we closely examined the ROC curve and experimented with different threshold values. Our objective was to identify a cutoff value that would yield a true positive rate (TPR) and false positive rate (FPR) below 10%, while still maintaining a satisfactory level of accuracy. By carefully analyzing the ROC curve and iteratively adjusting the cutoff value, we aimed to strike a balance between minimizing the rate of false positives and maximizing the rate of true positives. This process allowed us to identify a cutoff value that met our criteria for both TPR and FPR, ultimately enabling us to achieve a high level of accuracy while effectively controlling the false positive rate.

4.6.3 Learning Curve on Training Set



The learning curve analysis conducted on the training dataset reveals a steady enhancement in accuracy with the inclusion of additional training samples. The initial surge in accuracy occurs when the training size reaches 0.7, denoting a 70% - 30% partition between the training and validation sets. However, the gain in accuracy becomes negligible beyond this threshold. Consequently, we have opted to adopt a 70-30 split for segregating our data into training and validation sets based on these findings.

5.0 Conclusion

Key takeaways

We, as a group, have compiled some key takeaways from our project that we believe are essential to include in the project report:

Data Preprocessing: Throughout the project, we learned the significance of thorough data preprocessing and feature engineering. We emphasize the importance of cleaning the dataset, handling missing values effectively, and appropriately transforming variables, for example in the case of handling the NA values and imputing them with either mean, median or a static value based on the type and nature of the column. These steps are crucial to ensure the reliability and accuracy of the models we built.

Feature Selection: Although our code utilized various available features for modeling, we strongly recommend exploring feature selection techniques. By identifying the most relevant variables, we can potentially improve the performance of our models, reduce overfitting issues, and enhance interpretability. This additional step would greatly benefit the overall project.

Model Comparison: Our project involved comparing multiple models to evaluate their performance. This approach allowed us to gain a better understanding of how different algorithms perform in the given task. We believe it is important to consider various model types and select the one that provides the best balance between accuracy, TPR, and FPR. This comprehensive analysis ensures we make informed decisions regarding model selection.

Model Evaluation Metrics: In our project, we primarily employed accuracy, true positive rate (TPR), and false positive rate (FPR) as evaluation metrics for our models. While accuracy is commonly used, we also suggest considering additional metrics, such as precision, recall, and F1 score, depending on the specific requirements of the problem at hand. These additional metrics

provide a more comprehensive evaluation of model performance and should be included in the final report.

Documentation and Version Control: During the course of our project we realized the importance of the Documentation and Version Control. It allowed us to track previous iterations of our work, enabling us to identify shortcomings and make continuous improvements to our model throughout the project duration.

By including these key takeaways in our project report, we aim to provide a realistic and informative summary of our findings, methodologies, and recommendations.

Reflections

If we start the project again then we could have done the below things differently.

More EDA: Conduct a comprehensive exploratory data analysis for deeper insights into the data, including visualizations, correlation analysis, and statistical tests.

Hyperparameter Tuning: Fine-tune the models by performing hyperparameter tuning using techniques like grid search, random search, or Bayesian optimization.

Future scope of the project

Use of Advanced Modeling Techniques: Extend the project to explore advanced models like GBMs, SVMs, or deep learning models for capturing intricate patterns and nonlinear relationships.

Time-Series Analysis: Apply time-series analysis techniques to predict future ratings by considering temporal dependencies and incorporating lagged variables or time-based features.

User Segmentation: Explore user segmentation based on demographics or preferences and build separate prediction models for each segment to provide personalized rating predictions.

Online Deployment: Deploy the model as a predictive service or integrate it into an existing platform for real-time predictions and continuous monitoring.

Challenges Faced

During the course of our project, we encountered several challenges that required careful consideration and problem-solving. These challenges are outlined below:

Incorporation of External Data Set: We discovered an external dataset containing income information for individuals across different zip codes in the USA. Our initial intention was to merge this dataset with our existing data to potentially enhance the performance of our models. However, despite our efforts, we did not observe any improvement in our model's accuracy after incorporating this external dataset.

Determining Parameters in Text Mining: During the project, we encountered a significant challenge related to determining the optimal parameters for text mining, specifically in utilizing various text variables in our models. The amenities column was one such variable that required careful consideration in terms of preprocessing and utilization. Despite our efforts to incorporate the amenities column into our models, we found that it did not contribute to an increase in the true positive rate (TPR) or a decrease in the false positive rate (FPR). Consequently, we made the decision not to include the amenities column in our final model. This challenge required us to carefully evaluate and experiment with different approaches for handling text variables, considering their impact on model performance. While the amenities column did not prove beneficial in this context, we gained valuable insights into the importance of selecting relevant and impactful features for our predictive models.

Feature Engineering: While conducting feature selection, we encountered difficulties in finding a set of features that provided satisfactory performance for our models. We had to undergo multiple rounds of trial and error to identify the most relevant variables. This process required us to create new variables and explore different feature engineering techniques in order to achieve improved model performance.

Overcoming these challenges required careful analysis, experimentation, and a systematic approach to ensure the reliability and accuracy of our models. By addressing these obstacles, we were able to refine our methodologies and derive meaningful insights from our project.