Airbnb Predictive Model Report

*BUDT758T - Professor Jessica M Clark*

*Team 14: Vaidehi Deshpande, Raunak Ghawghawe, Xin Lan, Di Wang, Yannan Zhu*

### 

### **ORIGINAL WORK STATEMENT**

### We undersigned to certify that the actual composition of this proposal was done by us and is original work.

|  |  |  |
| --- | --- | --- |
|  | **Typed Name** | **Signature** |
| **Contact Author** | Raunak Ghawghawe | Raunak |
|  | Vaidehi Deshpande |  |
|  | Xin Lan | Xin Lan |
|  | Di Wang | Di Wang |
|  | Yannan Zhu | Yannan Zhu |

### 

### 

### **Team member names and contributions**

### Vaidehi Deshpande

### Model Creation and Code compilation

### Visualizations for model statistics

### Contributed to ? of the models

### Organize final R code

### Contributed to Final Report

### Raunak Ghawghawe

* Data cleaning

### Model Creation and Code compilation

### Visualizations for model statistics

### Contributed to cross validation, linear model, log model, ridge, lasso, knn, boosting, tuned boosting of the models

* Hyper parameters tuning
* Model evaluation and performance metrics
* Model selection

### Organize final R code

### Contributed to Final Report

### Xin Lan

### Model Creation and Code compilation

### Contributed to 2 (KNN and SVM) of the models

### Helped organize final R code

### Formatted and contributed to Final Report

### Di Wang

### Model Creation and Code compilation

### Contributed to 2 (CV and Trees) of the models

### Helped organize final R code

### Formatted and contributed to Final Report

### Yannan Zhu

### Created graphs demonstrating insights regarding features

### Helped organize final R code

### Formatted and contributed to Final Report

### 

### **Section 2: Business Understanding**

Airbnb.com is a home-sharing platform where homeowners can rent out full houses, apartments, bedrooms, or beds. Airbnb hosts make money from renting out their properties; renters provide feedback by posting public ratings and comments regarding their stay. Airbnb offers people an easy, relatively stress-free way to earn some income from their property. Guests often find Airbnb is cheaper, has more character, and is homier than hotels. Airbnb has also become an indispensable website for travel now.

We tried six different models for our prediction. They are KNN, TREE, SVM, logistic regression, ridge lasso, and cross-validation. The results of each model have different effects on different audiences. The result can provide some general insights into who could benefit from using these models and the potential business actions and value they can generate in the context of Airbnb.

For hosts, they can use the result for occupancy prediction. As A result of applying regression models like ridge lasso or SVM, hosts can predict optimal pricing for their listings based on factors such as location, property attributes, availability, and market demand. This can help hosts maximize their revenue while remaining competitive. Hosts also can use the results for occupancy prediction. Using models like logistic regression or decision trees, hosts can forecast the likelihood of booking their properties on specific dates. This information can be used to adjust pricing, offer promotions, or plan for marketing campaigns to increase occupancy rates.

For Airbnb company, they can use results for user segmentation. By employing clustering algorithms like KNN, Airbnb can group its users based on their preferences, behaviors, and demographics. This segmentation can then be used to personalize marketing strategies, improve targeted promotions, and enhance the overall user experience. They also can use results for customer satisfaction prediction. By utilizing regression models with cross-validation techniques, Airbnb can predict customer satisfaction scores based on factors such as reviews, ratings, and feedback. This information can be used to proactively address potential issues, improve service quality, and enhance customer retention. There are more strategies that can help Airbnb better, such as fraud detection, sentiment analysis, information tracking, and more.

The value of these predictions lies in empowering hosts and Airbnb's operations to make data-driven decisions. By leveraging these models, hosts can optimize their pricing strategies and increase their booking rates, while Airbnb can enhance customer satisfaction, prevent fraud, optimize business operations, and ultimately drive revenue growth for Airbnb and provide personalized experiences to its users. These actions can lead Airbnb to improve revenue, increase customer loyalty, provide more personalized and efficient services to both guests and hosts, differentiate itself in the market, and stay ahead of the competition.

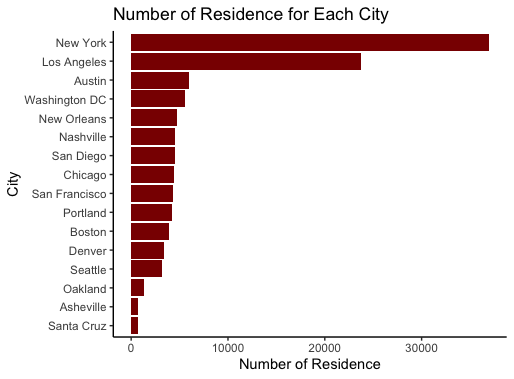
The data we use to give all information from project pages are from Airbnb data. We utilized data, clean data, and to build and train different models that will most accurately predict the success of future projects. We can learn different information from the different model results. Our goal is to find the best model that can help Airbnb or our target audience to improve their future processing.

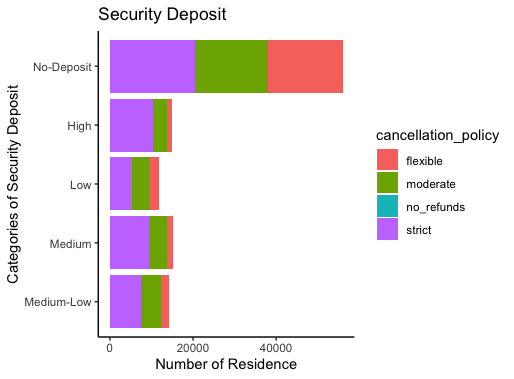
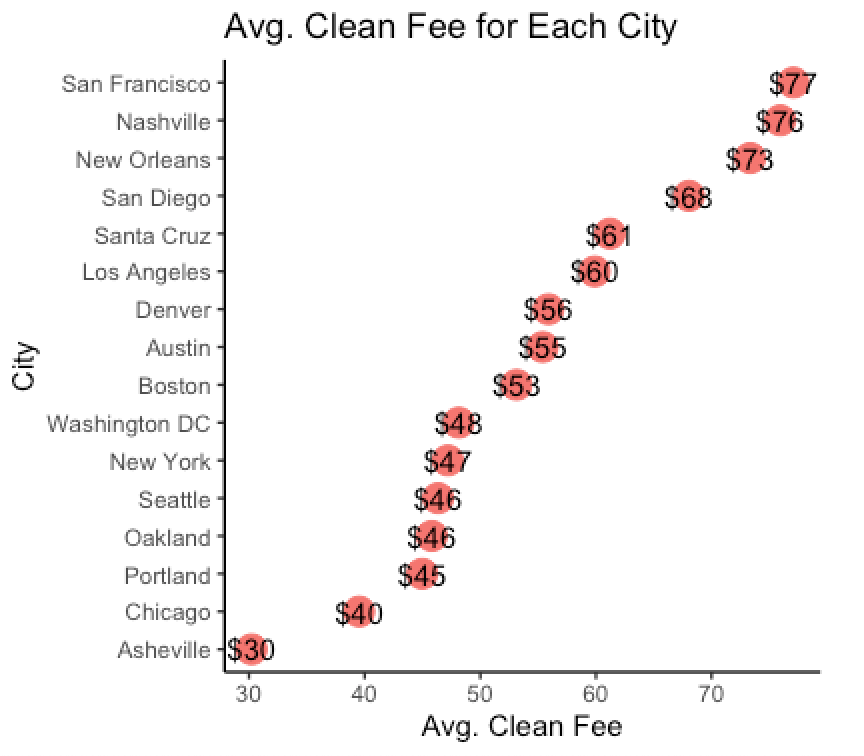
### **Section 3: Data Understanding and Data Preparation**

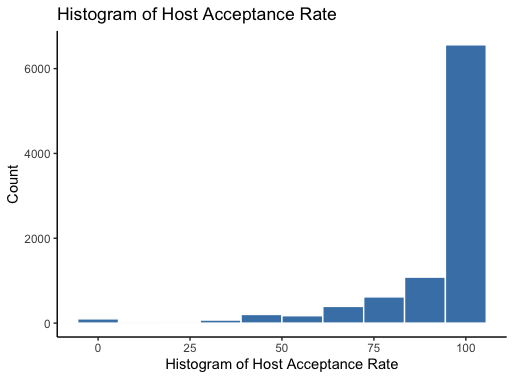
This section should include:

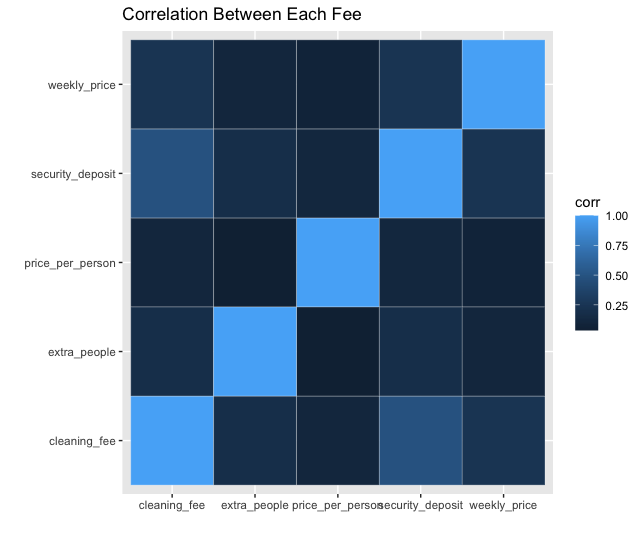
1) A table of the following format (for example):

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Feature Name | Brief Description | R Code Line Numbers |
| 1 | cancellation\_policy | Added factors super\_strict\_30, super\_strict\_60 | 1-5 |
| 2 | cleaning\_fee | Original feature from dataset | 34,35 |
| 3 | has\_cleaning\_fee | When cleaning\_fee >0 then “Yes”, cleaning\_fee = 0 then “No” | 36, 44 |
| 4 | price | Original feature from dataset | 37,38 |
| 5 | bedrooms | Original feature from dataset | 39 |
| 6 | beds | Original feature from dataset | 40 |
| 7 | host\_total\_listings\_count | Original feature from dataset | 41,42 |
| 8 | price\_per\_person | the nightly price per accommodates | 43 |
| 9 | bed\_category | “bed” if the bed\_type is Real Bed and “other” otherwise | 45 |
| 10 | property\_category | 1.apartment if property\_type is Apartment, Serviced apartment, Loft.  hotel  2. if property\_type is Bed & Breakfast, Boutique hotel, Hostel.  condo  3.if property\_type is Townhouse, Condominium.  house  4. if property\_type is Bungalow, House.  5.other, otherwise | 46-56, 59 |
| 11 | bed\_type | Original feature from dataset | 57 |
| 12 | room\_type | Original feature from dataset | 58 |
| 13 | ppp\_ind | 1 if the price\_per\_person is greater than the median for the property\_category, and 0 otherwise | 64-66 |
| 14 | bathrooms | Original feature from dataset | 71 |
| 15 | host\_is\_superhost | Replace NAs in host\_is\_superhost with FALSE | 72-73 |
| 16 | extra\_people | Original feature from dataset | 74 |
| 17 | charges\_for\_extra | “YES” if extra\_people > 0 and “NO” if extra\_people is 0 or NA | 75 |
| 18 | host\_acceptance\_rate | Percent of stay requests the host accepts | 76 |
| 19 | host\_acceptance | “ALL” if host\_acceptance\_rate = 100%, “SOME” if host\_acceptance\_rate < 100%, and “MISSING” if it’s NA. | 77-79 |
| 20 | host\_response\_rate | Original feature from dataset | 80 |
| 21 | has\_min\_nights | Original feature from dataset | 84 |
| 22 | market | Original feature from dataset | 85 |
| 23 | host\_has\_profile\_pic | Original feature from dataset | 85 |
| 24 | host\_identity\_verified | Original feature from dataset | 94-95 |
| 25 | instant\_bookable | Original feature from dataset | 98 |
| 26 | is\_business\_travel\_ready | whether the listing is available for business travel (“Yes”) or not (“No”) | 96-97 |
| 27 | is\_location\_exact | whether the listing reports the exact location (“Yes”) or not (“No”) (usually for privacy purposes) | 101 |
| 28 | license\_status | whether the host has a hotelier license (“Yes”) or not (“Pending”) | 102-103 |
| 29 | monthly\_price | Original feature from dataset | 104-105 |
| 30 | require\_guest\_phone\_verification | whether the host requires a phone number to verify the guest's ID (Yes) or not (No) | 106 |
| 31 | require\_guest\_profile\_picture | whether the host requires the guest's profile picture (Yes) or not (No) | 107 |
| 32 | requires\_license | whether the listing is in a jurisdiction that requires the host to have a license (Yes) or not (No) | 108 |
| 33 | security\_deposit | Original feature from dataset | 109-110 |
| 34 | has\_security\_deposit | When security\_deposit equal to  1. -Inf-0 means "No-Deposit",  2. 0-100 means "Low",  3. 100-200 means "Medium-Low"  4. 200-400 means "Medium"  5. 400-Inf means "High" | 111-114 |
| 35 | state | Original feature from dataset | 115 |
| 36 | area | When square\_feet equal to  1. Inf-372 means "Small"  2. 372-750 means "Medium-Small"  3. 750-913 means "Medium"  4. 913-Inf means "Large" | 116-119 |
| 37 | weekly\_price | Original feature from dataset | 120-121 |
| 38 | host\_response | “ALL” if host\_response\_rate = 100%, “SOME” if host\_response\_rate < 100%, and “MISSING” if it’s NA. | 81-83 |
| 39 | property\_category\_medians | median of price per person for property category | 77 |
| 40 | hr\_no\_smoking | whether it’s a smoke allowed place | 150 |
| 41 | years\_since\_first\_review | time duration since first review | 136 |
| 42 | hr\_no\_pets | whether pets are allowed or not | 151 |
| 43 | host\_since\_years | calculated field showing the years since host owned the place | 142 |
| 44 | maximum\_nights | if maximum\_nights>30 then "short-term" else "long-term" | 143 |
| 45 | has\_summary | TRUE if place has summary else FALSE | 144 |
| 46 | has\_host\_about  hr\_no\_parties | TRUE if host allows parties else FALSE | 145 |
| 47 | has\_notes | TRUE if place has notes else FALSE | 146 |
| 48 | transit\_convenience | TRUE if place has transit convenience else FALSE | 147 |
| 49 | hr\_clean\_up | whether parties are allowed at the place | 153 |
| 50 | hr\_restricted\_checkin\_out | whether the place has restricted check out rules | 154 |
| 51 | hr\_quite\_hours | whether the place has noise curfew or quite hours etc | 156 |
| 52 | hr\_no\_extra\_guests | if the place has extra guests | 157 |
| 53 | hr\_remove\_shoes | if place allows to remove shoes inside the place | 159 |
| 54 | hr\_respect\_neighbours | if place respects neighbours | 160 |
| 55 | access\_full | if place has total full unrestricted total or unrestricted access | 164 |
| 56 | access\_parking | if place has parking accessible | 166 |
| 57 | access\_restricted | if place has limited restricted controlled specific | 169 |
| 58 | access\_keyless\_entry | if place has keyless entry | 171 |
| 59 | is\_commercial\_host | if place has commercial host | 177 |

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







3) (optional) Any additional comments about the data or the steps you undertook for data preparation.

For predictions:

* We merged the labeled training data and unlabeled test data, after separating the target variables. This merged data frame was used for all part of the cleaning process.
* After cleaning the rows were reordered and the train and test data was split like it was originally.
* We used one hot encoding to reward the numerical and categorical variables by converting them to factors.
* This one hot data frame was used for all models.

For Graphs:

Chart 1 displays the distribution of residences across different cities, highlighting the dominance of New York and Los Angeles. These metropolitan areas boast substantial populations and expansive territories, warranting a separate analysis due to their concentrated nature.

Chart 2 presents the average cleaning fees per city, revealing that popular tourist destinations occupy the top positions. Conventionally, areas with a higher number of residences would be expected to have lower cleaning fees. However, contrary to this assumption, we observe that the average cleaning fee does not exhibit an inverse correlation with the number of residences, as indicated in Chart 1. We infer that in tourist cities, there is a significant demand for accommodations while the housing supply remains limited. Consequently, the level of cleaning fees appears to have minimal impact on these markets.

Chart 3 illustrates the ratio between each cancellation policy and its corresponding security deposit. When categorizing security deposits from "high" to "low," the strict policy holds a higher proportion, whereas the flexible policy exhibits a smaller proportion. Notably, the distribution of cancellation policies remains relatively consistent across all security deposit categories. However, in the case of the "No-Deposit" security policy, the ratios of flexible, moderate, and strict are remarkably similar, indicating an equal distribution among these categories.

Chart 4 displays the distribution of the Host Acceptance Rate, revealing that in the majority of cases, homeowners accept reservations from guests. However, a noteworthy observation is the presence of a 0% Host Acceptance Rate, indicating instances where the house has never been rented out. This peculiar behavior warrants further investigation to determine the underlying causes.

Chart 5 presents an analysis of the correlations among different costs. While the overall correlation is not strong, it is noteworthy that a significant correlation exists between the security deposit and cleaning fee. Additionally, there are moderate correlations between the fee for extra people and the cleaning fee, as well as between the security deposit and the fee for extra people, and the weekly fee and the cleaning fee. These observed correlations indicate potential relationships worth further exploration and analysis.

### **Section 4: Evaluation and Modeling**

1) Include a short (one-paragraph) description of the “winning” model, the variables included in the model, your estimated training and generalization performance, and how you decided that it was the winning model. Also list the line numbers in your R code where you generated the final predictions that you submitted for the contest.

**Winning model: Tuned Boosting Model**

The winning model in our analysis was the tuned boosting model. Boosting is a machine learning technique that combines multiple weak learners (models) to create a strong ensemble model. In our case, the boosting model was tuned to optimize its performance. The tuned boosting model demonstrated superior performance compared to other models that we implemented, such as linear regression, logistic regression, KNN, lasso regression, ridge regression, and simple boosting. Through the hyper parameter tuning process, the boosting model's parameters were optimized to achieve the best possible results. The tuned boosting model likely leveraged its ability to handle complex relationships and capture non-linear patterns in the data, resulting in improved predictive accuracy. It effectively learned from the training data and generalized well to unseen data, making it a reliable model for making predictions on new observations. The selection of the tuned boosting model as the winning model suggests that it outperformed the other models in terms of predictive power and overall performance. It could be considered a robust and effective model for the specific problem at hand, providing valuable insights and accurate predictions for the target variable.

**Variables included :** perfect\_rating\_score,accommodates,bathrooms,bed\_category,bed\_type,bedrooms,beds,cancellation\_policy,charges\_for\_extra,guests\_included,has\_cleaning\_fee, has\_min\_nights,has\_security\_deposit,host\_acceptance,host\_has\_profile\_pic,host\_identity\_verified,host\_is\_superhost,host\_response,host\_total\_listings\_count,instant\_bookable,is\_business\_travel\_ready,is\_location\_exact,license\_status,market,ppp\_ind,price,property\_category,require\_guest\_phone\_verification,require\_guest\_profile\_picture,requires\_license,room\_type,access\_full,access\_keyless\_entry,access\_restricted,access\_parking,amenities\_count,years\_since\_first\_review,host\_since\_years,maximum\_nights,has\_host\_about,has\_summary,transit\_convenience,has\_notes,hr\_no\_parties,hr\_no\_pets,hr\_no\_smoking,hr\_clean\_up,hr\_restricted\_checkin\_out,hr\_quite\_hours,hr\_no\_extra\_guests,hr\_remove\_shoes,hr\_respect\_neighbours

**Estimated training and generalization performance :**

**Estimated Training Performance:**

Max TPR at optimal cutoff: 0.3730769

Min FPR at optimal cutoff below 10%: 0.09964547

Optimal Cutoff: 0.06

Accuracy at Optimal cutoff: 0.7449575

Boosting Tuned AUC: 0.7804665

**Estimated Training Performance:**

Max TPR at optimal cutoff: 0.36

Min FPR at optimal cutoff below 10%: 0.09

Accuracy at Optimal cutoff: 0.74

**Line numbers for final predictions in R code : 740**

2) For each type of model that you try, please list:

**Linear Regression:**

1. Model Family - Linear Regression
2. R library - Base R Package

R Funciton - lm()

1. Performance:

"Max TPR at optimal cutoff: 0.327036199095023"

"Min FPR at optimal cutoff below 10%: 0.0928385724415032”

"Cutoff: 0.47 "

"Accuracy at cutoff: 0.736189364894149 "

"Linear AUC: 0.756184668002066 "

1. The methodology used is simple train/validation split, cross validation and also optimization of cut-offs to get the Maximum TPR & Minimum FPR.

Specific parameters of validation setup are :k= 5, for cross validation

1. Best set of features: perfect\_rating\_score,accommodates,bathrooms,bed\_category,bed\_type,bedrooms,beds,cancellation\_policy,charges\_for\_extra,guests\_included,has\_cleaning\_fee,has\_min\_nights,has\_security\_deposit,host\_acceptance,host\_has\_profile\_pic,host\_identity\_verified,host\_is\_superhost,host\_response,host\_total\_listings\_count,instant\_bookable,is\_business\_travel\_ready,is\_location\_exact,license\_status,market,ppp\_ind,price,property\_category,require\_guest\_phone\_verification,require\_guest\_profile\_picture,requires\_license,room\_type,access\_full,access\_keyless\_entry,access\_restricted,access\_parking,amenities\_count,years\_since\_first\_review,host\_since\_years,maximum\_nights,has\_host\_about,has\_summary,transit\_convenience,has\_notes,hr\_no\_parties,hr\_no\_pets,hr\_no\_smoking,hr\_clean\_up,hr\_restricted\_checkin\_out,hr\_quite\_hours,hr\_no\_extra\_guests,hr\_remove\_shoes,hr\_respect\_neighbours
2. Line number: 293

**Logistic Regression :**

1. Model Family - Generalized Linear Models
2. R library - Base R Package

R Function - glm()

1. Performance:

"Max TPR at optimal cutoff: 0.345361990950226 "

"Min FPR at optimal cutoff below 10%: 0.0928385724415032 "

"Cutoff: 0.47 "

"Accuracy at cutoff: 0.736189364894149 "

"Logit AUC: 0.759967488404376 "

1. The methodology used is simple train/validation split, cross validation and also optimization of cut-offs to get the Maximum TPR & Minimum FPR.

Specific parameters of validation setup are :k= 5, for cross validation

1. Best set of features: perfect\_rating\_score,accommodates,bathrooms,bed\_category,bed\_type,bedrooms,beds,cancellation\_policy,charges\_for\_extra,guests\_included,has\_cleaning\_fee,has\_min\_nights,has\_security\_deposit,host\_acceptance,host\_has\_profile\_pic,host\_identity\_verified,host\_is\_superhost,host\_response,host\_total\_listings\_count,instant\_bookable,is\_business\_travel\_ready,is\_location\_exact,license\_status,market,ppp\_ind,price,property\_category,require\_guest\_phone\_verification,require\_guest\_profile\_picture,requires\_license,room\_type,access\_full,access\_keyless\_entry,access\_restricted,access\_parking,amenities\_count,years\_since\_first\_review,host\_since\_years,maximum\_nights,has\_host\_about,has\_summary,transit\_convenience,has\_notes,hr\_no\_parties,hr\_no\_pets,hr\_no\_smoking,hr\_clean\_up,hr\_restricted\_checkin\_out,hr\_quite\_hours,hr\_no\_extra\_guests,hr\_remove\_shoes,hr\_respect\_neighbours
2. Line number: 299

**Ridge Regression:**

1. Model Family - Reguralized Regression
2. R library - glmnet

R function - glmnet()

1. Performance:

"Max TPR at optimal cutoff: 0.0917901938426454 "

"Min FPR at optimal cutoff below 10%: 0.0917901938426454 "

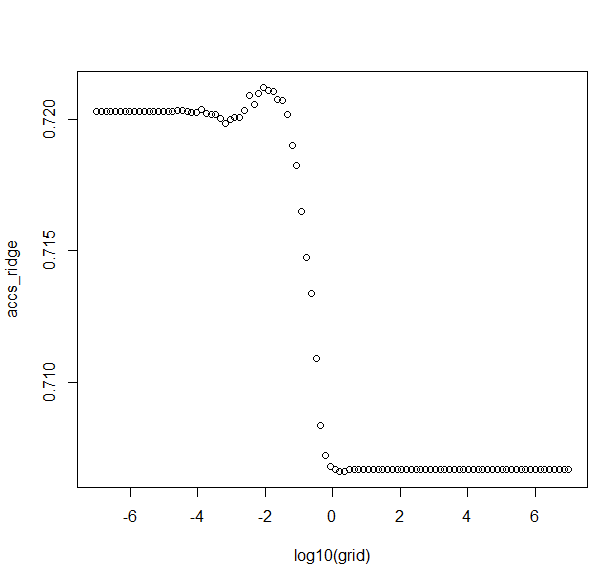
"Optimal Cutoff: 0.49 "

"Accuracy at Optimal cutoff: 0.668078013002167 "

"Ridge AUC: 0.501914507871763 "

1. The methodology used is simple train/validation split, cross validation, optimization of cut-offs to get the Maximum TPR & Minimum FPR, and grid search for optimization of lambda..

Specific parameters of validation setup are :k= 5, for cross validation, set of lambda = 10^-7 to 10^7

1. Best set of features: perfect\_rating\_score,accommodates,bathrooms,bed\_category,bed\_type,bedrooms,beds,cancellation\_policy,charges\_for\_extra,guests\_included,has\_cleaning\_fee,has\_min\_nights,has\_security\_deposit,host\_acceptance,host\_has\_profile\_pic,host\_identity\_verified,host\_is\_superhost,host\_response,host\_total\_listings\_count,instant\_bookable,is\_business\_travel\_ready,is\_location\_exact,license\_status,market,ppp\_ind,price,property\_category,require\_guest\_phone\_verification,require\_guest\_profile\_picture,requires\_license,room\_type,access\_full,access\_keyless\_entry,access\_restricted,access\_parking,amenities\_count,years\_since\_first\_review,host\_since\_years,maximum\_nights,has\_host\_about,has\_summary,transit\_convenience,has\_notes,hr\_no\_parties,hr\_no\_pets,hr\_no\_s
2. Line number: 303
3. Hyperparameters tuned lambda( 10^-7 to 10^7)
4. 

**Lasso Regression:**

1. Model Family - Reguralized Regression
2. R library - glmnet

R function - glmnet()

1. Performance:

"Max TPR at optimal cutoff: 0.0916761687571266 "

"Min FPR at optimal cutoff below 10%: 0.0916761687571266 "

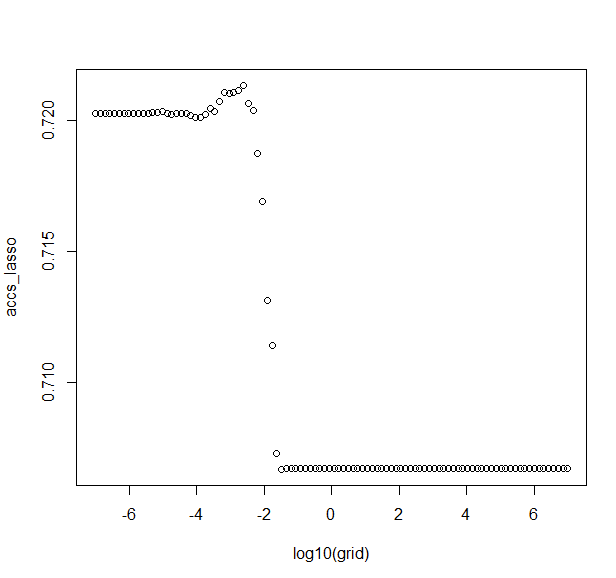
"Optimal Cutoff: 0.48 "

"Accuracy at Optimal cutoff: 0.667677946324387 "

"Lasso AUC: 0.501899455216881 "

1. The methodology used is simple train/validation split, cross validation, optimization of cut-offs to get the Maximum TPR & Minimum FPR, and grid search for optimization of lambda..

Specific parameters of validation setup are :k= 5, for cross validation, set of lambda = 10^-7 to 10^7

1. Best set of features: perfect\_rating\_score,accommodates,bathrooms,bed\_category,bed\_type,bedrooms,beds,cancellation\_policy,charges\_for\_extra,guests\_included,has\_cleaning\_fee,has\_min\_nights,has\_security\_deposit,host\_acceptance,host\_has\_profile\_pic,host\_identity\_verified,host\_is\_superhost,host\_response,host\_total\_listings\_count,instant\_bookable,is\_business\_travel\_ready,is\_location\_exact,license\_status,market,ppp\_ind,price,property\_category,require\_guest\_phone\_verification,require\_guest\_profile\_picture,requires\_license,room\_type,access\_full,access\_keyless\_entry,access\_restricted,access\_parking,amenities\_count,years\_since\_first\_review,host\_since\_years,maximum\_nights,has\_host\_about,has\_summary,transit\_convenience,has\_notes,hr\_no\_parties,hr\_no\_pets,hr\_no\_s
2. Line number: 303
3. Hyperparameters tuned lambda( 10^-7 to 10^7)
4. 

**kNN:**

1. Model Family - Instance-Based Learning Family
2. R library - class

R function - knn()

1. Performance:

"Max TPR at optimal cutoff: 0.0917420814479638 "

"Min FPR at optimal cutoff below 10%: 0.0917420814479638 "

"OptimalCutoff: 0.46 "

"Accuracy at Optimal cutoff: 0.669244874145691 "

"kNN AUC: 0.505852116622516”

1. The methodology used is simple train/validation split, cross validation, optimization of cut-offs to get the Maximum TPR & Minimum FPR, and grid search for optimization of k..

Specific parameters of validation setup are :k= 5, for cross validation, set of k = 1:50

1. Best set of features: perfect\_rating\_score,accommodates,bathrooms,bed\_category,bed\_type,bedrooms,beds,cancellation\_policy,charges\_for\_extra,guests\_included,has\_cleaning\_fee,has\_min\_nights,has\_security\_deposit,host\_acceptance,host\_has\_profile\_pic,host\_identity\_verified,host\_is\_superhost,host\_response,host\_total\_listings\_count,instant\_bookable,is\_business\_travel\_ready,is\_location\_exact,license\_status,market,ppp\_ind,price,property\_category,require\_guest\_phone\_verification,require\_guest\_profile\_picture,requires\_license,room\_type,access\_full,access\_keyless\_entry,access\_restricted,access\_parking,amenities\_count,years\_since\_first\_review,host\_since\_years,maximum\_nights,has\_host\_about,has\_summary,transit\_convenience,has\_notes,hr\_no\_parties,hr\_no\_pets,hr\_no\_s
2. Line number: 365
3. Hyperparameters tuned k(1-50)

**Boosting:**

1. Model Family - Machine Learning Ensemble Methods
2. R library - gbm

R function - gbm()

1. Performance:

"Max TPR at optimal cutoff: 0.360972850678733 "

"Min FPR at optimal cutoff below 10%: 0.360972850678733 "

"Optimal Cutoff: 0.52 "

"Accuracy at Optimal cutoff: 0.744824137356226 "

"Boost AUC: 0.780323666837424 "

1. The methodology used is simple train/validation split, cross validation, optimization of cut-offs to get the Maximum TPR & Minimum FPR, and tuning of hyper parameters Specific parameters of validation setup are :k= 5
2. Best set of features: perfect\_rating\_score,accommodates,bathrooms,bed\_category,bed\_type,bedrooms,beds,cancellation\_policy,charges\_for\_extra,guests\_included,has\_cleaning\_fee,has\_min\_nights,has\_security\_deposit,host\_acceptance,host\_has\_profile\_pic,host\_identity\_verified,host\_is\_superhost,host\_response,host\_total\_listings\_count,instant\_bookable,is\_business\_travel\_ready,is\_location\_exact,license\_status,market,ppp\_ind,price,property\_category,require\_guest\_phone\_verification,require\_guest\_profile\_picture,requires\_license,room\_type,access\_full,access\_keyless\_entry,access\_restricted,access\_parking,amenities\_count,years\_since\_first\_review,host\_since\_years,maximum\_nights,has\_host\_about,has\_summary,transit\_convenience,has\_notes,hr\_no\_parties,hr\_no\_pets,hr\_no\_s
3. Line number: 433

**Boosting Tuned:**

1. Model Family - Machine Learning Ensemble Methods
2. R library - gbm

R function - gbm()

1. Performance:

"Max TPR at optimal cutoff: 0.373076923076923 "

"Min FPR at optimal cutoff below 10%: 0.0996454738832427 "

"Optimal Cutoff: 0.06 "

"Accuracy at Optimal cutoff: 0.744957492915486 "

"Boosting Tuned AUC: 0.780466469208631 "

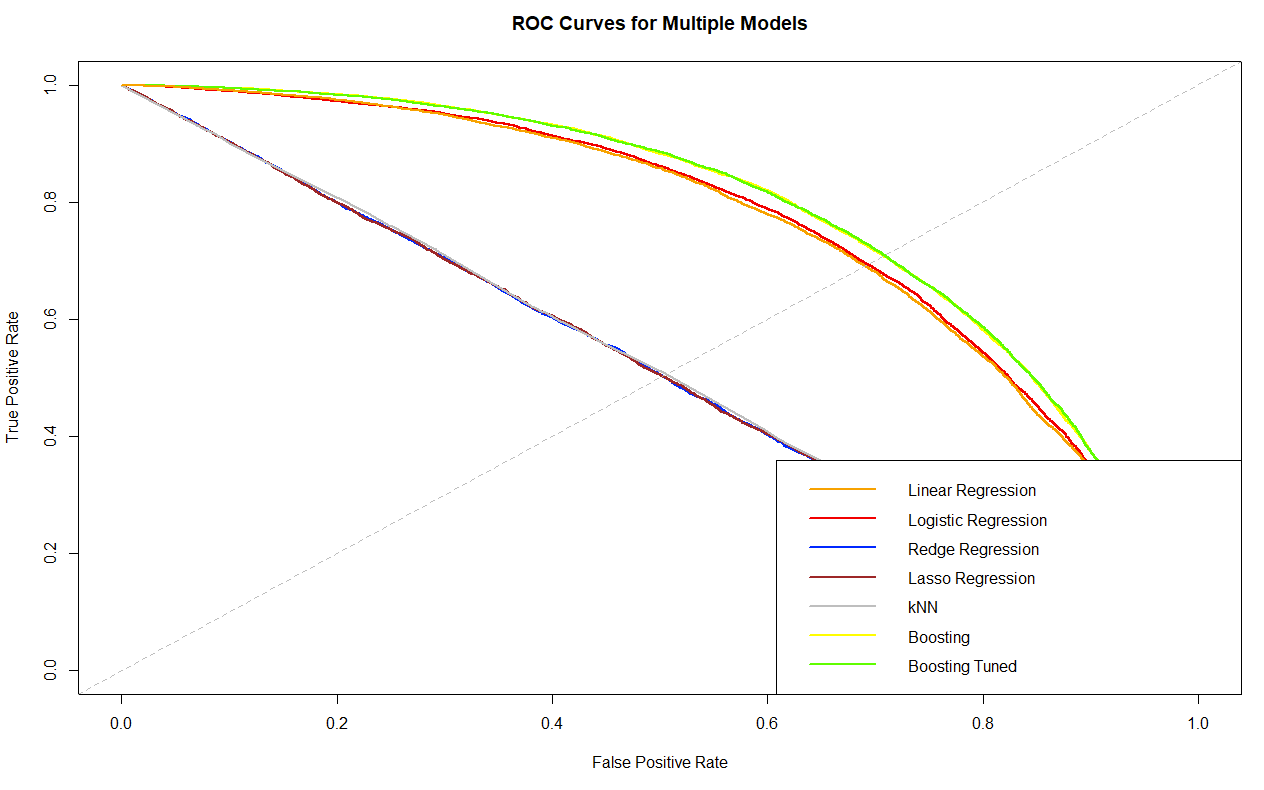
1. The methodology used is simple train/validation split, cross validation, optimization of cut-offs to get the Maximum TPR & Minimum FPR, and tuning of hyper parameters Specific parameters of validation setup are :k= 5
2. Best set of features: perfect\_rating\_score,accommodates,bathrooms,bed\_category,bed\_type,bedrooms,beds,cancellation\_policy,charges\_for\_extra,guests\_included,has\_cleaning\_fee,has\_min\_nights,has\_security\_deposit,host\_acceptance,host\_has\_profile\_pic,host\_identity\_verified,host\_is\_superhost,host\_response,host\_total\_listings\_count,instant\_bookable,is\_business\_travel\_ready,is\_location\_exact,license\_status,market,ppp\_ind,price,property\_category,require\_guest\_phone\_verification,require\_guest\_profile\_picture,requires\_license,room\_type,access\_full,access\_keyless\_entry,access\_restricted,access\_parking,amenities\_count,years\_since\_first\_review,host\_since\_years,maximum\_nights,has\_host\_about,has\_summary,transit\_convenience,has\_notes,hr\_no\_parties,hr\_no\_pets,hr\_no\_s
3. Line number: 447
4. Hyperparameters tuned: n.trees(500,1000,2000), interaction.depth(3,4,5),shrinkage(0.1,0.01), n.minobsinnode(10,20)

3) The ROC curve was generated to compare the performance of several models in predicting the perfect rating score. The models under consideration included linear regression, logistic regression, ridge regression, lasso regression, kNN, basic boosting, and tuned boosting.

### The ROC curve visually illustrates the trade-off between the true positive rate (TPR) and the false positive rate (FPR) for each model. The TPR represents the proportion of actual positive cases correctly identified by the model, while the FPR represents the proportion of actual negative cases incorrectly classified as positive.

### After analyzing the ROC curve, it was observed that the tuned boosting model outperformed the other models in terms of overall predictive accuracy. It achieved a higher TPR and lower FPR compared to the rest. Additionally, the linear regression, logistic regression, and basic boosting models also demonstrated competitive performance, with similar TPR and FPR values.

### Based on these results, it can be concluded that the tuned boosting model is the most promising choice for accurately predicting the target variable. However, further analysis and consideration should be given to the specific requirements and context of the problem before finalizing the model selection.



### 

### 

### **Section 5: Reflection/takeaways**

Our team demonstrated good communication and collaboration throughout the project. We hold regular meetings and group chats to discuss progress, clarify goals and address any challenges. One of our challenges is holding regular meetings because each member has a different schedule. Assigning tasks posed another challenge for us, as each member of the group had a different level of proficiency in R, resulting in inconsistent time for everyone to complete the task, but the team members can negotiate well.

If our group could start the project over again, we will do two things differently. The first is to put more emphasis on comprehensive planning and set goals at the initial stages. The second is model selection and evaluation, instead of solely relying on a select model, we would consider experimenting with multiple models and evaluating their performance using appropriate validation techniques. This would allow us to compare different algorithms and choose the most suitable one based on the specific requirements of the project.

If we have another few months to work on the project, we would focus on several key areas to improve the model's performance and overall effectiveness. We would spend time enhancing the model's training set. We would also thoroughly clean and preprocess the data, ensuring that it is in the optimal format for modeling. We also would explore new features that could potentially improve the model's predictive power and would fine-tune the chosen model by experimenting with different hyperparameters to optimize its performance.

The advice we have for the group who will start this project next year includes three points. The first is to start with a clear understanding of project objectives and scope. This means that before diving into the project, the group should have a well-defined understanding of what they aim to achieve and the boundaries within which they will work. It involves clearly defining the goals, desired outcomes, and deliverables of the project. The second is to stay organized by setting realistic timelines and milestones. This includes establishing effective channels of communication and regular meeting schedules, estimating time frames that make sure all deadlines are met, and regular monitoring and adjustment which means if something unexpected happens, the team can make changes in time. The third is to allocate tasks based on individual strengths and expertise. For example, for our project, we have different experiences with R. The best way for us to collaborate is to divide the work based on our abilities.

After running several classification models for predicting Airbub airlines, we have found out that Tuned Boosting Model has the best performance. According to our data analysis, we have found out that

The most necessary takeaway from this project is that team spirit is crucial, and a good team spirit is essential to achieve successful projects. Each group member needs to be assigned by the reasonable distribution of tasks. Also, a perfect team production requires the cooperation of each member. Even if we have different levels of R or other software, we are able to solve problems together in completing tasks, and improve our professional skills. Everyone is able to contribute to the group work, to seek common ground while reserving differences, to respect each other, to reach agreement on the basis of negotiation. It is important that everyone has the opportunity to express themselves.