Data Mining - Final Project Report

BUDT758T DATA MINING AND PREDICTIVE ANALYTICS
Professor Jessica Clark
Project Team 8
Date: 16th May 2023

Group Members:
Tania Sinhasan
Nishank Shah
Rituparna Desai
Preksha Jagtap
Nimrah Mehmooda

Table of Contents

Section 1: Team member names and contributions

After engaging in a comprehensive discussion about the diverse characteristics of the datasets, potential models to employ, and preprocessing steps, we decided to distribute the different models among ourselves to determine their respective performance levels.

Half of the team dedicated their efforts to feature engineering and data cleaning tasks, which involved comprehensive data preprocessing techniques. This group meticulously handled missing values, outliers, and inconsistencies to ensure the dataset's integrity and quality. Additionally, they employed their expertise to create new features that could potentially provide valuable insights and improve the model's predictive performance. By addressing these crucial steps, this team aimed to enhance the dataset's usability and prepare it for further analysis.

Simultaneously, the other half of the team focused on feature analysis and selection, leveraging their skills to identify the most influential features for model training. They conducted an in-depth exploration of the dataset, examining factors such as correlation, impact on model performance, and statistical significance through techniques like determining the p-value. By conducting thorough feature analysis, this team aimed to identify the most informative features that would contribute significantly to the model's accuracy and interpretability.

Here is the revised allocation:

1. Tania Sinhasan - xgboost model + Feature engineering
2. Nishank Shah - Random Forest model + Feature selection
3. Rituparna Desai - xgboost model + Feature engineering
4. Preksha Jagtap - Random Forest model + Feature selection
5. Nimrah Mehmooda - Logistic Regression model + Feature engineering

We aimed to conduct a thorough evaluation and comparison of these models to determine their performance and suitability.

Section 2: Business Understanding

The rapidly growing popularity of Airbnb.com as a homesharing platform presents opportunities and challenges for homeowners, vacation rental management businesses, potential investors, and even Airbnb's competitors. This data mining project focuses on Contest 1, which involves predicting the perfect_rating_score of Airbnb listings. The predictive model developed aims to provide valuable insights into the factors that contribute to perfect ratings, enabling stakeholders to make data-driven decisions and take actions to enhance their performance.

Business Cases

a) Airbnb Hosts: The predictive model developed for Contest 1 can provide valuable insights to Airbnb hosts regarding the key elements that contribute to receiving perfect ratings. By identifying these crucial factors, hosts can focus on improving their listings and enhancing the overall visitor experience. This, in turn, can lead to higher ratings, increased guest satisfaction, and ultimately, more booking opportunities for hosts.

b) Vacation Rental Management Businesses: The model's predictions can be leveraged by vacation rental management businesses that oversee multiple Airbnb listings. By utilizing the model's insights, these businesses can assess the performance of the residences under their management. They can identify areas for improvement and provide guidance to hosts on how to enhance their listings based on the factors that lead to perfect ratings. This can result in improved property management tactics, increased guest satisfaction, and enhanced competitiveness in the vacation rental market.

c) Potential Investors: Investors contemplating investments in the vacation rental sector can utilize the predictive model's outputs to assess the likelihood of success for Airbnb listings. By considering the predictions of perfect_rating_score, investors can make well-informed decisions about real estate purchases and investment opportunities. The model's insights can provide valuable guidance and help investors identify properties with a higher potential for positive ratings and successful returns on their investments.

d) Airbnb Competitors: Businesses operating in the vacation rental industry, including Airbnb's competitors, can leverage the classification model developed in this project. By gaining insights into the key elements that contribute to perfect ratings, competitors can benchmark their own platforms and services. They can identify areas for improvement and develop competitive strategies to enhance their offerings. This knowledge can help them differentiate their services, attract more hosts and guests, and improve overall user satisfaction.

Importance of the predictions

a) Listing Optimization: Leveraging the identified key qualities highlighted by our predictive model as crucial for achieving perfect ratings, hosts can strategically enhance their listings. By focusing on these essential factors, hosts can improve guest experiences, boost positive ratings, and ultimately achieve higher occupancy rates and increased income. This optimization approach enables hosts to make data-driven decisions that elevate the quality of their offerings and maximize their potential for success.

b) Risk Assessment: Our model's predictions provide valuable insights for prospective investors seeking to assess the likelihood of a property receiving perfect ratings or identify properties that may be less likely to achieve such ratings. Armed with this knowledge, investors can make informed decisions, effectively mitigating risks and optimizing their profitability. This risk assessment capability empowers investors to make smarter choices based on data-driven evaluations, enhancing their confidence in property investments and facilitating long-term financial gains.

Conclusion

Our classification model is a useful resource for different market participants, such as Airbnb hosts, vacation rental management businesses, possible investors, and rival companies. These stakeholders are better able to make data-driven decisions, streamline their operations, and perform better in the market by utilizing the forecasts and insights offered by our model. This model's application could boost client satisfaction, boost earnings, and promote corporate expansion.

Our methodology can also be used by hosts and property management companies to proactively spot listings that are more likely to get low ratings. Before they arise, hosts can take the required steps to address these possible problems, avoiding negative feedback. By improving the overall guest experience, hosts and property management companies can boost their chances of earning positive reviews, draw in more visitors, and eventually increase their revenue.

In conclusion, our model offers useful insights and forecasts that empower stakeholders to decide wisely, run their businesses more efficiently, and perform better in the vacation rental market. By using this strategy, hosts and property management companies can spot possible problems before they arise and take proactive measures to fix them, increasing client satisfaction and revenue.

Section 3: Data Understanding and Data Preparation

Model Features

ID	Feature Name	Brief Description	R Code Line Numbers
1	availability_30	Original feature from dataset	171
2	availability_365	Original feature from dataset	174
3	bathrooms	Original feature from dataset	167
4	beds	Original feature from dataset	166
5	cancellation_policy	Original feature from dataset, taken as factor	124
6	city_name	Original feature from dataset, taken as factor	257
7	extra_people	Original feature from dataset	149
8	first_review	Original feature from dataset	143
9	guests_included	Original feature from dataset	93
10	host_is_superhost	Original feature from dataset, taken as factor	150

11	host_response_rate	Original feature from dataset	189
12	host_response_time	Original feature from dataset, taken as factor	160
13	host_listings_count	Original feature from dataset	153
14	host_identity_verified	Original feature from dataset, taken as factor	152
15	host_since	Original feature from dataset	169
16	instant_bookable	Original feature from dataset, taken as factor	268
17	is_location_exact	Original feature from dataset, taken as factor	259
18	is_business_travel_ready	Original feature from dataset, taken as factor	277
19	price	Original feature from dataset, taken as log	162, 234
20	room_type	Original feature from dataset, taken as factor	253
21	maximum_nights	Original feature from dataset, taken as log	231
22	minimum_nights	Original feature from dataset, taken as log	232
23	require_guest_phone_verification	Original feature from dataset, taken as factor	269

24	monthly_price	Original feature from dataset, taken as log	164, 240
25	no_of_amenities	Numerical variable created from “amenities” feature	127
26	has_notes	Factor variable created from “notes” feature	158
27	has_security_deposit	Factor variable created from “security_deposit” feature	175
28	has_square_feet	Factor variable created from “square_feet” feature	185
29	is_extra_people	Factor variable created from “extra_people” feature	196
30	is_availability_30	Factor variable created from “availability_30” feature	200
31	is_availability_365	Factor variable created from “availability_365” feature	205
32	pratio	Ratio between “price” and “accommodates”	217
33	wratio	Ratio between “weekly_price” and “accommodates”	218
34	mratio	Ratio between “monthly_price” and “accommodates”	219

35	sratio	Ratio between “security_deposit” and “accommodates”	220
36	aratio	Ratio between “square_feet” and “accommodates”	221
37	apratio	Ratio between “price” and “square_feet”	222
38	bedroomratio	Ratio between “bedrooms” and “accommodates”	226
39	property_category	Factor variable created from “property_type” feature	244

Feature Insights:

1. To gain a deeper understanding of the impact of a specific feature on the target variable, we employed both the table function and the summary function.

   The table function allowed us to examine the distribution of the target variable across different levels or categories of the feature. By tabulating the data, we could observe the frequency or count of each category and assess any imbalances or patterns in the distribution. This analysis provided valuable insights into how the feature relates to the target variable and whether certain categories had a higher or lower likelihood of being associated with the target.

   Example 1

```
> table(train_y$perfect_rating_score, train_x$accommodates)

         1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    18
  NO  5520 28643  7457 12978  3798  6134  1085  2339   328  1145   113   479    41   142    53   294     0
  YES 3133 11972  2647  5125  1415  2575   425  1019   122   478    38   231    17    61    23   147     2

        19    26
  NO     1     1
  YES    0     0
```

   Here, we would intuitively expect the number of listings having a perfect rating score to go up with the number of people a property can accommodate, as more often than not, bigger groups struggle to find one common place for the whole group, and providing this can be something that leads to a perfect rating score. However, as we can see in this table, the ratio of listings without a perfect score to listings with one, does not deviate much across the whole spread for the number of people that the listing accommodates.

   Example 2

```
> table(train_y$perfect_rating_score, train_x$cancellation_policy)

      flexible moderate no_refunds strict super_strict_30 super_strict_60
  NO     12804    21870          4  35641             168              64
  YES     9033     8540          0  11722              93              42
```

   The strictness level of any listing’s cancellation policy can have a considerable impact on what kind of rating a guest might leave. From this table, we can see that listings with a flexible cancellation policy have the highest percentage of properties having a perfect rating score. This makes it a good feature for our modeling purposes.

NOTE: super_strict_60 also shares a similar trend, but the data points are too low in number to draw meaningful conclusions from.

Using this method, we shortlisted many features for our model.

2. We analyzed the correlation between features by generating a correlation matrix. This allowed us to identify and avoid features that exhibited high similarity or redundancy. In doing so, we could make informed decisions about which features to retain and which ones to exclude from our models.

3. By examining the p-value associated with each feature, we could determine its significance and contribution to the model's performance. In this process, we fit a simple logistic regression model, considering one feature at a time. The resulting p-value for each feature served as an indicator of its statistical significance. By evaluating the p-values, we were able to prioritize and focus on features that demonstrated a significant association with the target variable.

```
                                                 Pr(>|z|)
(Intercept)                          < 0.0000000000000002 ***
availability_30                        0.0000000000000119 ***
availability_365                       0.0000000012503565 ***
bathrooms                              0.0000000027543298 ***
beds                                   0.0000000000022607 ***
cancellation_policymoderate          < 0.0000000000000002 ***
cancellation_policyno_refunds                    0.890132
cancellation_policystrict            < 0.0000000000000002 ***
city_nameAustin                                  0.311687
city_nameBoston                      < 0.0000000000000002 ***
city_nameChicago                       0.00000951026782168 ***
city_nameDenver                        0.0000000023128972 ***
city_nameLos Angeles                   0.00000017780936958 ***
city_nameNashville                               0.000105 ***
city_nameNew Orleans                   0.0000000000293958 ***
city_nameNew York                    < 0.0000000000000002 ***
city_nameOakland                                 0.218348
city_namePortland                      0.0000000000151707 ***
city_nameSan Diego                               0.701754
city_nameSan Francisco               < 0.0000000000000002 ***
city_nameSanta Cruz                              0.964232
city_nameSeattle                                 0.608495
city_nameWashington DC                 0.0000002515228415 ***
extra_people                                     0.026038 *
first_review                         < 0.0000000000000002 ***
guests_included                        0.00003995088604109 ***
host_is_superhostTRUE                < 0.0000000000000002 ***
host_response_rate                     0.00000080379921483 ***
host_response_timeOTHER                          0.053866 .
host_response_timewithin a day         0.00000905579779811 ***
host_response_timewithin a few hours   0.00000020767762180 ***
host_response_timewithin an hour       0.0000000156688271 ***
host_listings_count                  < 0.0000000000000002 ***
host_identity_verifiedTRUE                       0.009283 **
```

4. By analyzing the gain of each feature, we could quantify its contribution to the model's predictive power. Higher gain values indicated that a feature had a greater impact on improving the model's accuracy and reducing the prediction error. In addition, we examined the cover of each feature, which provided insights into the number of observations that a feature influenced. Features with a higher cover indicated that they were more prevalent and had a greater influence on the model's decision-making process. This analysis helped us prioritize features that provided the most valuable information for making accurate predictions.

	Feature	Gain	Cover
1:	first_review	0.38783994323	0.1359881989
2:	pratio	0.05521608790	0.0501451320
3:	host_response_time.OTHER	0.04534132651	0.0061226201
4:	city_name.New.York	0.04253921010	0.0249101061
5:	availability_30	0.03503646619	0.0252291168
6:	availability_365	0.03206338680	0.0216094992
7:	instant_bookable.TRUE	0.03147796197	0.0085622800
8:	host_response_time.within.an.hour	0.03133780332	0.0044882285
9:	price	0.02743444441	0.0518400281
10:	minimum_nights	0.02718115191	0.0461217788
11:	host_is_superhost.TRUE	0.02497778105	0.0085313502
12:	host_listings_count	0.02231877306	0.0291977590
13:	city_name.Los.Angeles	0.01633762487	0.0062897369
14:	host_since	0.01610704351	0.0645390052
15:	city_name.Austin	0.01283475379	0.0108803526
16:	bathrooms	0.01200429559	0.0206669590
17:	extra_people	0.01162086478	0.0100538756
18:	sratio	0.01139622622	0.0266285153
19:	host_response_rate	0.01083637699	0.0280134861
20:	maximum_nights	0.00917979058	0.0262999783
21:	cancellation_policy.strict	0.00911261136	0.0077367455
22:	city_name.San.Francisco	0.00900953174	0.0222385997
23:	bedroomratio	0.00877512935	0.0099197217
24:	city_name.San.Diego	0.00864994396	0.0125918644
25:	property_category.house	0.00860647805	0.0076886453
26:	city_name.Seattle	0.00836317317	0.0108802864
27:	no_of_amenities	0.00699892986	0.0259026500
28:	has_notes.TRUE	0.00618679846	0.0058030204
29:	city_name.Boston	0.00598227188	0.0127637079
30:	guests_included	0.00536938985	0.0114513970
31:	apratio	0.00526686302	0.0151535321

Section 4: Evaluation and Modeling

Winning model

Our winning model was based on the XGBoost algorithm using 39 features, which achieved an impressive accuracy of approximately 77% on the validation dataset. Additionally, when evaluating the model's performance at a false positive rate (FPR) of 10%, we observed a true positive rate (TPR) of around 46% on the validation dataset. The list of final features that were included in the model were:

1. availability_30
2. availability_365
3. bathrooms
4. beds
5. cancellation_policy
6. city_name
7. extra_people
8. first_review
9. guests_included
10. host_is_superhost
11. host_response_rate
12. host_response_time
13. host_listings_count
14. host_identity_verified
15. host_since
16. instant_bookable
17. is_location_exact
18. is_business_travel_ready
19. price
20. room_type
21. maximum_nights
22. minimum_nights
23. require_guest_phone_verification
24. monthly_price
25. no_of_amenities
26. has_notes
27. has_security_deposit
28. has_square_feet
29. is_extra_people
30. is_availability_30

31. is_availability_365
32. pratio
33. wratio
34. mratio
35. sratio
36. aratio
37. apratio
38. bedroomratio
39. property_category

Note: The description of these variables is provided in the above section.

The decision that the XGBoost model was the winning model was based on a thorough evaluation and comparison of various models. We considered several factors to assess the performance and suitability of each model. These factors included accuracy, true positive rate (TPR), computational efficiency, and stability.

After training and testing multiple models, the XGBoost algorithm consistently demonstrated superior performance across these metrics. It achieved an impressive accuracy of approximately 77% on the training dataset, indicating its ability to correctly classify instances.

The code for generating the model and calculating its performance is in between lines to 390 to 452.

All models

Model 1:

a) Type: XGBoost

b) R Function used: xgboost

```
402
403 bst <- xgboost(data = as.matrix(data_train_x_num),
404                label = as.matrix(data_train_y_num),
405                max.depth = 2, eta = 0.2, nrounds = 700,
406                objective = "binary:logistic")
407
```

c) Performance on validation dataset:
   - Accuracy = 77% (approx.)
   - True positive rate at 10% FPR = 46% (approx.)

tpr_at_cutoff	0.453508174386921
tpr_at_cutoff2	0.465159755268525

acc_bst	0.771004200840168
acc_bst2	0.774516177426614

d) To estimate the generalization performance of our model, we employed a train-validation split approach. We divided the original training dataset into three parts: a training set consisting of 60% of the data (59988 observations) and two validation sets, each comprising 20% of the data (19996 observations each). To assess the stability of our model and gain insights into its performance on different training datasets, we performed the splits using various seeds. By systematically varying the seeds during the training process, we generated multiple training datasets with different random samples.

e) The best performing set of features of this model are as follows:

	Feature	Gain	Cover
1:	first_review	0.38783994323	0.1359881989
2:	pratio	0.05521608790	0.0501451320
3:	host_response_time.OTHER	0.04534132651	0.0061226201
4:	city_name.New.York	0.04253921010	0.0249101061
5:	availability_30	0.03503646619	0.0252291168
6:	availability_365	0.03206338680	0.0216094992
7:	instant_bookable.TRUE	0.03147796197	0.0085622800
8:	host_response_time.within.an.hour	0.03133780332	0.0044882285
9:	price	0.02743444441	0.0518400281
10:	minimum_nights	0.02718115191	0.0461217788

These features consistently improved the TPR and Accuracy of the model for the multiple training datasets.

f) The code for generating the model and calculating its performance is in between lines to 390 to 452.

g) For XGBoost model, we set the hyperparameters as follows:

   - max.depth = 2
   - eta = 0.2
   - nrounds = 700
   - objective = "binary:logistic"

Model 2:

a) Type: Random Forest

b) R Function used: ranger

```
369 # Random forest model
370 rf.mod <- ranger(x = data_train_x, y = data_train_y,
371                  mtry=39, num.trees=700,
372                  importance="impurity",
373                  probability = TRUE)
374
```

c) Performance on validation dataset:
   - AUC = 80 (approx.)

auc_rf	0.80155045338022

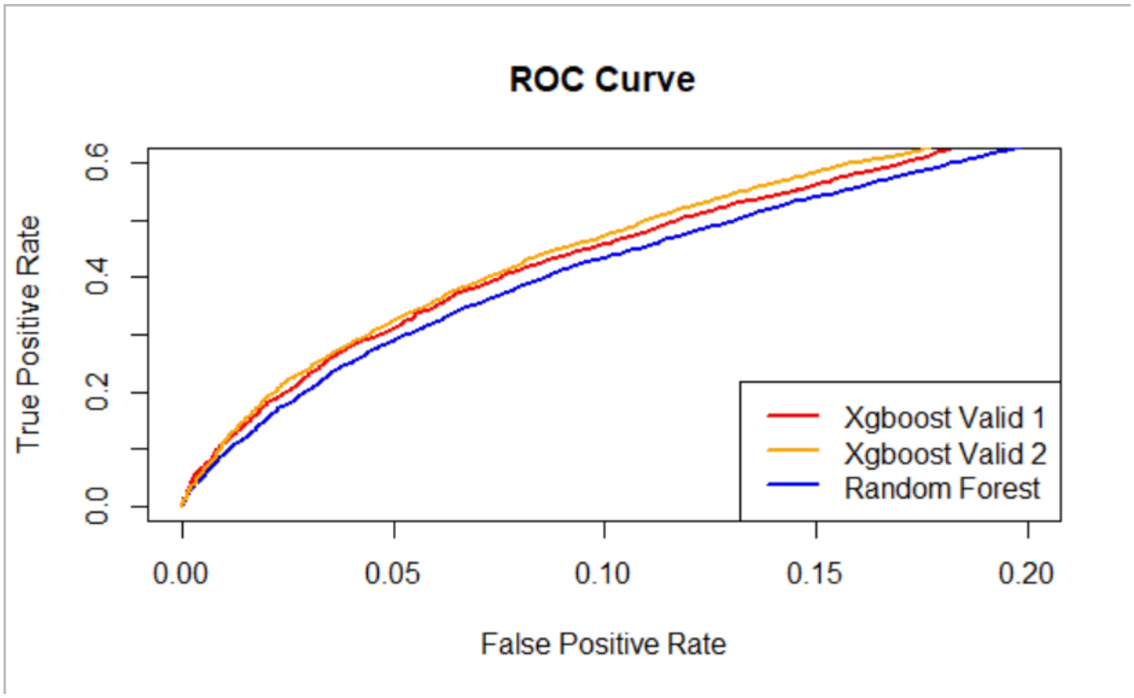
d) To estimate the generalization performance of our model, we employed a train-validation split approach. We divided the original training dataset into three parts: a training set consisting of 60% of the data (59988 observations) and two validation sets, each comprising 20% of the data (19996 observations each). To assess the stability of our model and gain insights into its performance on different training datasets, we performed the splits using various seeds. By systematically varying the seeds during the training process, we generated multiple training datasets with different random samples.

e) The best performing set of features of this model were first_review, pratio and instant_bookable. These features consistently improved the TPR and Accuracy of the model for the multiple training datasets.

f) The code for generating the model and calculating its performance is in between lines to 369 to 387.

g) For Random Forest model, we set the hyperparameters as follows:
   - mtry = 39
   - num.trees = 700
   - importance="impurity"
   - probability = TRUE

ROC Curve for the two models:



Section 5: Reflection/takeaways

What did your group do well?

Over the course of this project, our group displayed various skills and competencies across different tasks, the most notable ones being:

1. Understanding and Processing of data: Our group was able to go through and understand the nature of the data right away which helped us later with the thorough data cleaning and preprocessing that was required. This knowledge of the data helped us use our intuition and judgment to further streamline our approach when it came to feature selection. This in particular enabled us to get started on modeling sooner than expected, and also made sure that we did not need to turn back and alter things retrospectively when it came to data processing.

2. Business Context Understanding: Our group was able to quickly identify the use-cases for a project such as this, and this shared understanding of the same helped us to think along common lines when it came to selection and addition of features, which eventually led to a better overall model.

What were the main challenges?

1. Data Cleaning: The dataset provided was quite exhaustive and had a lot of features that needed to be carefully studied before making a decision upon. This, along with the irregularities in data points for a lot of features meant that a lot of effort was needed to be put in to make the data usable for modeling.

2. Feature Engineering: After selecting the optimal set of features, there was still a lot of feature engineering required to find the underlying patterns of the features and how some features can interact with each other to provide a better representation of the full dataset. This was time consuming as well as difficult to implement, but essential for the building of an effective model.

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

A lot of our initial iterations were built using sub-optimal sets of features, which made us hit a wall in terms of generalization performance. To overcome this, we relied on data modeling fundamentals and decided on optimal feature sets based on individual p-values for the features respectively. Having done this earlier would have given us more time to improve our model.

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

Improved Model Interpretability: We would explore techniques and methodologies to improve the interpretability of our model. This would involve employing methods such as analyzing feature importance or using alternative algorithms that provide more interpretable outputs.

Integration of External Data Sources: We would incorporate relevant external data sources to improve our model's predictions and insights.