|  |  |
| --- | --- |
| A description... | **BUDT 758T Class Project**  **Airbnb.com Prediction Contest** |

# Introduction

Airbnb.com is a homesharing 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.

For this data mining project, your team’s job is to try to do the best job of predicting one of two target variables on a test set for which you do not know the labels. Each team can enter one of two contests, each of which involves predicting one target variable. There will be two total winning teams across the two sections of BUDT758T - one winning team for each target variable.

* **Contest 1 – predict perfect\_rating\_score**: Predict whether an Airbnb.com listing will have a perfect rating score (binary YES/NO predictions whether the listing had a 100% perfect rating score). The winning team will generate binary predictions for the test instances with the **highest True Positive Rate (TPR), given a False Positive Rate (FPR) of no more than 10%.** *This means that if your predictions in the test set have an FPR of more than 10%, your team will not be eligible for the top prize.*
* **Contest 2 – predict high\_booking\_rate**: Predict whether an Airbnb.com listing will have a high booking rate (numerical predictions for the probability that a listing will have a high booking rate). The winning team will achieve the highest **AUC** for the test instances. *There is no disqualifying threshold as in Contest 1, but AUC is a more complicated metric to optimize for.*

# Data Sets

The data comprise nearly 100,000 training instances, and just over 12,000 test instances. I have provided you with about 70 features for each project, some of which you should already be familiar with from the first two homework assignments. Descriptions of these variables are given in the data dictionary.

There are four data sets posted on Canvas:

1. airbnb\_train\_X\_2023.csv: features for the training instances (99,981 listings).
2. airbnb\_train\_y\_2023.csv: labels for the two target variables for the training instances (perfect\_rating\_score, high\_booking\_rate).
3. airbnb\_test\_X\_2023.csv: features for the test instances (12,205 instances). **Your goal is to make predictions for the instances in the test set – you will not see the actual labels in the test set.**
4. data\_dictionary\_2023.xlsx: descriptions of the 70 features for each listing.

# Contest Rules

1. There will be two winning teams (one for each target variable). The winning performance will be based ***only*** on your final submission.
2. Prior to the final submission, there will be at least six official opportunities to have your predictions evaluated against the test set labels.
   1. If you choose to take advantage of these opportunities, your team’s score will go on a leaderboard that everyone can see.
   2. Each team can have their predictions evaluated for at most one contest per evaluation round. If your team submits more than one set of predictions, I will choose the most recent submission. This also means that you can submit to different contests across different evaluations.
   3. You can get “extra” evaluations (other than the six official ones) but each one will cost your team one point from the final project grade.
3. You will be required to turn in the code you’ve written to generate your final model/feature set, and I expect to see different code for each team.
   1. **If multiple teams have the same winning performance, there will be no winner.**
   2. You are allowed to share code and insights with other teams, but remember that you’re in competition with each other!
4. You are not allowed to use any information regarding the target variables to make your predictions. **If I see any evidence of this in your code you will be disqualified from the contest.**
   1. Please keep to the spirit of the contest and only use machine learning and feature engineering techniques to improve your predictions.
5. All feature engineering and machine learning must be done in R.
6. Submission format: Follow the exact format of the sample output files. I have included a sample R script (sample\_prediction\_code.R) that generated these outputs. **If your predictions don’t match this format exactly, you might not get credit for your performance!**

# Deliverables

I have posted the grading rubric. Only the winning team will be eligible to get a perfect score on the project, but it is possible for any team to get a good grade on the project. The winners will also get a special prize. Of course, the ultimate prize is always impressing your professor ;)

Your deliverables are:

1. The final set of predictions.
2. The R code you used to generate your final predictions (including feature cleaning/engineering, model selection, and evaluation).
3. A short report (template forthcoming).
4. Peer evaluations.

# Important Dates and Deadlines

|  |  |  |
| --- | --- | --- |
| Date | Time | Deliverable |
| April 10 | Midnight | First evaluation |
| April 11, 13 |  | No class – project consultations |
| April 17 | Midnight | Second evaluation |
| April 24 | Midnight | Third evaluation |
| May 1 | Midnight | Fourth evaluation |
| May 4 |  | Final Exam |
| May 7 | Midnight | Fifth Evaluation |
| May 9 | Noon | Sixth Evaluation |
| May 10 | Midnight | Final predictions due |
| May 11 |  | Last day of class – contest winners announced |
| May 12 | Midnight | Final report, R code, peer evaluations due |

# Hints and Tips

* Your first goal should be to get the data loaded into R and explore it a bit. For the first submission I recommend trying to train models using a subset of the training instances – the data from the second homework assignment would work well.
* Your second goal should be to try to get usable predictions in the correct format (start simple and work up to more complicated models).
  + Make sure your submissions are always in the correct format, including the file name.
  + Also make sure you have the correct number of rows and no NA’s in your submissions!
* Any transformations you do on the training data will also have to be applied to the test data.
  + In fact, you should stack the training and test sets for any data preparation and/or engineering, then re-split them for model training.
* This data is messy! Some attributes have missing values (both systematically missing and randomly missing), there may be errors or inconsistencies, some attributes are redundant, etc. You’ll have to check the data and decide what to do about the various issues.
* You may find it helpful to document what you’ve tried as you go, to facilitate writing your report later.