From: Airbnb Data Analytics and Insight Team – New York Branch (via Sunny Chang)
To: Chief Data Officer, Grant Case
CC'd: Chief Digital Officer, Jessica Meyer
CC'd: Chief Operations and Analytics Officer, Soumya Kalra
Subject: Forecasting Rental Prices

Context:

It has come to our attention that Airbnb is attempting to forecast rental price because rental prices are significantly tied to our revenue and profit margins. Forecasting rental prices by using different models has allow us to see factors and consumer behavior trends of what customers are looking for when renting out an Airbnb property. Then, we are able to use these trends to create data analyses.

Our team has developed an optimal predictive model to be able to predict rental prices. We used an important metric called Root Mean Square Error (RMSE) to help us evaluate which model is optimal. We want a model with a low RMSE, that has more predictive power, and is the closest to reality. We were able to achieve a low RMSE due to a combination of 6 modeling approaches – Linear Regression, Decision Tree, Random Forest, Ranger, Bagging, Tuned Ranger. So far with our efforts, we conducted 6 modeling approaches, as of result, we were able to achieve an RMSE of 189.1483. For the reference, the R code will be provided under a separate report.

Data Preparation:

Airbnb data had many Nas (Null) values that needed to be addressed before advancing to creating models. First, we merged the analysis and scoring dataset together to make a base data. We divided the analysis dataset into test and train by splitting using the caret package in R.

As we're looking at the predictors in the analysis dataset, we came up with hypotheses before we get really technical and start modeling. Analysis data set is where we build our models and scoring data set helps us to check our models from the analysis data set. This is why we look for variables in the analysis data set to figure out which variables we should include when it comes to building our model.

Subset Selection:

After looking at 95 variables in the analysis dataset excluding price because price is our intercept, we used inductive reasoning of 18 out of 95 variables to see which are most likely to impact rental price, and then used feature selection to narrow it down to these 5 variables:

Variables:	Class()	Inductive Reasoning:
Bathrooms	numeric	People like to travel in groups and having many bathrooms is a plus especially if some people in your party takes too long in the bathroom. People would pay more to have a peace of mind.
Accommodates	numeric	People like to travel in groups and stick together, so they would want to be lodging at the same place. So, knowing how many people can be lodge at a rental property is important and don't mind having to pay extra because everyone would probably chip in.

Security Deposit	numeric	When people cancel because things come up, life happens, and they know that when they cancel, they'll lose that security deposit, but they'll factor in that and the price of the rental.
Cleaning Fee	numeric	The hosts get additional money by cleaning before guests arrive, or after guests leave. For example, when you go book a hotel room, you leave tips for the hotel staff who cleans up the room after you leave except in this scenario, the hosts tell you how much you pay upfront for them to clean after you leave. This fee is separate from how much you spend in their homestay per night. Each host have a different price point for the fee.
Room Type	categorical	People have different room type preferences when it comes to where they want to lodge at for vacation. They might like a room that feels like you're in a hotel, or apartment. Each room type has different price points.

Feature Selection:

We have extracted the 18 variables which makes up our subset. Now we are going to test our hypotheses and test our assumptions using feature selection. We want the best 5 variables that significantly impact rental price.

Top 5 Rankings based on significance using Feature Selection:

1. **Accommodates**
2. **Cleaning Fee**
3. Security Deposit
4. Room Type
5. **Bathrooms**

Data Wrangling:

We transformed baseData$bed_type, baseData$property_type, and basedata$instant_bookable into factors. Then, we cleaned the columns of the 3 significant variables we found: accommodates, bathrooms, cleaning fee.

Why we didn't use security deposit as a predictor? Security deposit is indeed significant but not to price, our intercept, when we run our Linear Regression Model later, the correlation between price and security deposit is not strong, but accommodate and bathrooms have a strong correlation with price. You can compare the correlations using cor(): cor(train$security_deposit, train$price) with the other 3 variables we'll use in our models: accommodates, bathrooms, cleaning fee, and see how strong each correlation is with each variable against price.

Why we didn't use room type as a predictor? Linear regression and other algorithms don't like categorical variables. It was difficult in eliminating the NAs of categorical variables such as Room Type. We tried to coerce it to convert it to a numeric, or a factor, and then remove the NAs. We tried doing it backwards too as in removing the NAs, then try to convert it to a different class. It didn't work.

On to dropping rows for the 3 significant predictors we chosen, we turned all of the predictors into numeric because it's easier to use numeric variables when it comes to modeling for our preference. We then figure out

which ones to clean using summary(), so we know which columns have NA's. We found baseData$bathrooms and basedata$cleaning_fee have NA's.

R console: summary(baseData$bathrooms)
Run:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.00	1.00	1.00	1.15	1.00	22.00	70

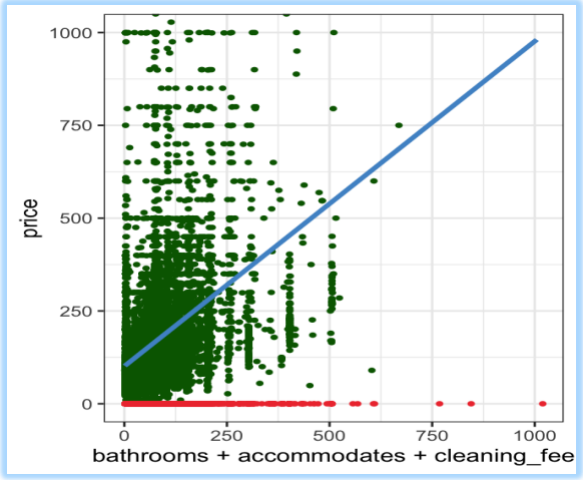
Then, we dropped rows that have NA's in bathrooms in our baseData because there's not a lot of NA's when you compare it to how many rows are in the baseData. There are 41,739 rows in the baseData and we're only dropping 70 rows which won't hurt our model too much. We do the same for cleaning fee.

We checked summary(), and table() to make sure these NA's are removed, so that we can proceed with conducting our models, or else our RMSE will get NA's as the answer and we don't want that. Our models will be affected because whenever we run RMSE, it will keep giving us NA's as the answer for every RMSE we run in every model we attempt.

Modeling and Utilizing Algorithms:

Simple Model: Linear Regression

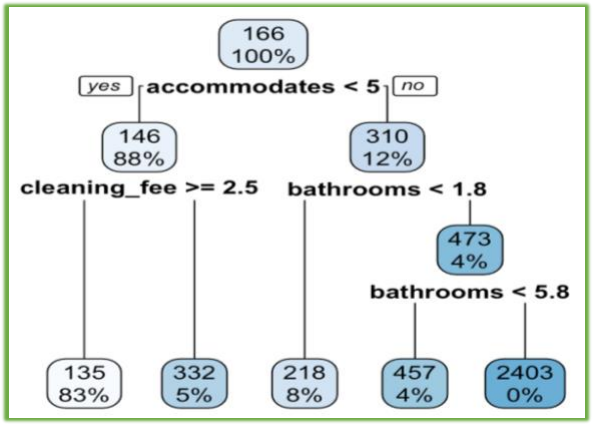
Based on our theory approach, we were able to start with an RMSE of 183.0847 using our linear regression model. Our predictors were based on the results of our feature selection: accommodate, bathrooms, cleaning fee. Based on Appendix 1, we see a high positive correlation between price and our predictors.



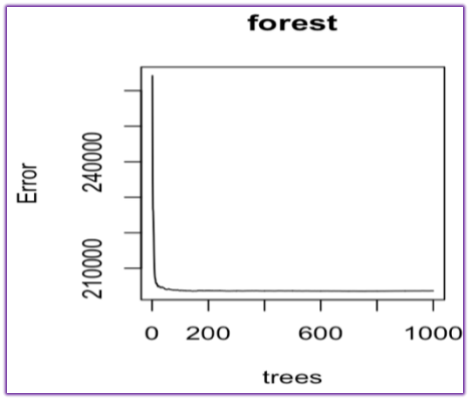
Appendix 1

Simple Model: Decision Trees

Appendix 2 is a simple decision tree with an RMSE of 185.7574. We are greedy so the tree starts with accommodates because that's our most important variable. We made complexity to 0.005 because we didn't want to make our tree to be too complex.



Appendix 2



Appendix 3

Advance Model: Random Forest

Our random forest has a RMSE of 190.856. I wanted random forest to build us 1,000 trees. In Appendix 3, we see that once you get to 200 trees, it averages it out at that point after the drop.

Advance Model: Ranger, Bagging, Tuned Ranger

Our ranger RMSE is 190.9052, tuning it brought it down to 189.1483, so our model is doing better with tuned ranger. We tell each model to build us 1,000 trees to be consistent in what we're using in all of our models.

Modeling Approach:

Model	RMSE
Ranger	190.9052
Bagging	213.008
Tuned Ranger	189.1483

Our thoughts:
We would use decision trees to be able to explain to the audience because it's easier to explain and easily interpreted by a non-expert. Then, for the most optimal model, we chose to use the RMSE of tuned ranger because tuned forest ranger model can run across cross validation and find the best tuned for us. Our tuned ranger does have a slightly higher RMSE compare to our linear regression because tuned ranger does tend to over fit. However, tuned ranger RMSE wasn't far off from linear regression RMSE and based on our understanding of the prediction vs. inference chart, hence why we believe Tuned Ranger is the most optimal model for predicting rental prices because it has high accuracy, but low explain ability.

My Reflection:

Rights	Wrongs
I was able to use 6 different models to see which one can give me the best model with the lowest RMSE and be able to be confident to explain why I chose this model. I can give reasons why I chose these predictors using inductive reasoning and feature selection. I went to tutoring for assignments 7 & 8 because I knew these assignments can help me understand the concepts and be able to use it them in the Kaggle Competition.	I tried to impute NA's using mean but it did not work out, so I removed the rows that had NA's in it in the predicator's columns, so I can properly run my models and RMSE. Also, I attempted to impute NA's for categorical variables, but it was difficult.

What would I do if I was given more money for phase 2 of the project? I would want to try to impute NA's instead of dropping rows by using the mean or mode again. I would attempt one more advance model: XGBoost. I want to also try to incorporate categorical variables such as: room type in all of my models but linear regression and other algorithms don't like categorical variables, so it's going to be difficult.

Closing Statement: My goal in the beginning for this Kaggle Project was to understand what I was coding using algorithms and gain real life experience using R. I am actually very satisfied with my work because I have achieved my goal. It was definitely a challenge, but it was worth it to apply the concepts that I learned in class like Prediction vs. Inference, Data Tidying, Data Exploration to the Kaggle Competition, and what we'll eventually have to do in real life. It was a nice first try, but I know with reviewing my notes, going to tutoring, doing the assignments, taking advance courses, and getting more experience in R that I can do even better in the future.