**Kaggle Project Report**
**Changfei(Judy) Guan**

## 1. Exploring the data

I started the project by running summary() and str() commands of the local dataset in order to understand what variables are presented in the dataset as well as their levels if they are categorical variables. While browsing all the information, I thought about which variables could be related to the price and started to note them.

I ended up choosing three categories of variables that I think are related to the Airbnb price, including the property itself, the host as well as reviews.

Property: neighbourhood_group_cleansed + bathrooms + bedrooms + room_type + bed_type + beds
calculated_host_listings_count_entire_homes + calculated_host_listings_count +
calculated_host_listings_count_private_rooms + calculated_host_listings_count_shared_rooms

Host/Management : minimum_nights + host_identity_verified + instant_bookable + cancellation_policy +
guests_included +availability_90+availability_30+availability_60+extra_people +
minimum_nights_avg_ntm

Past Clients Review: review_scores_rating + number_of_reviews + review_scores_accuracy +
review_scores_cleanliness + review_scores_checkin+review_scores_communication
+ review_scores_location+review_scores_value

***Reflection:***

Although hand picking the variables manually based on common sense can work, I would run feature selection if I were to do this project again. Basically what I have done here is to throw in all the variables that are clean, ready to use and might have some effect on price based on my speculation. I am not fully clear which variables are truly statistically significant or not. Running feature selection is also more scientific and convincing if I need to present this project to other people.

## 2. Efforts to prepare the data

While my report moves to the next section of preparing data, the actual process of doing the project involves a lot of back and forth of tidying the data. After including all the variables mentioned above, my RMSE score is still very high. Then I began creating new variables based on the existing ones.

***Creating New Variables:***

I noticed that amenities factor can influence the price of an Airbnb property a lot but this variable is not categorical nor numerical and I couldn't include it in my model as it is. Then I began separating all the amenities into independent variables like TV, Kitchen, Wifi and etc as below:

has_TV: *amenities, pattern= "TV"*
has_kitchen: *amenities, pattern = "Kitchen"*
has_wifi: *amenities, pattern = "Wifi"*
has_ac: *amenities, pattern = "Air conditioning"*
has_washer: *amenities, pattern = "Washer"*
has_gym : *amenities, pattern = "Gym"*
has_doorman : *amenities, patter = "Doorman"*

Also, I found that how long had the host been on the site can also influence the price. From there, I created a numerical variable called "host_days" by subtracting "host_since_new" from today's date as below.

host_days: *currentdate-host_since_new.* how long has the host been on Airbnb

***Missing Values:***

Apart from creating new variables, I also replaced missing values in order to tidy up the data. Specifically, I replaced the missing values with their means for "beds" and "host_days". Although this process should be done before running the model, I didn't develop the habit of checking missing values at the start. So I didn't realized this issue until I started running my model and error message appeared.

***Other Oversights:***

This was not the only mistake that I made for the data preparation. When I started creating new variables in my local dataset, I didn't realize that I should do the same for the scoringData dataset as well and my model couldn't be applied at all. Sometimes even when you create a same variable in both local dataset and the scoringData, the model might fail to apply because the same variable doesn't have the same levels across the datasets. And that's why I had to drop variables including "property_type" and "cancellation policy". This oversight made me realize that I not only need to examine and explore the local dataset, but also need to compare it to the dataset that I am predicting.

***Reflection:***

The process of cleaning and preparing data is the very foundation of future model constructions, and this is where I did very poorly. Putting the issues of data preparation I have encountered in writing as above doesn't seem complicated, but they actually took me a lot of time and efforts to figure out along the way. And thanks to this project, I realized that being

organized is the key for this step. Being organized means not only checking all the missing values and comparing local dataset to the scoring dataset, but also keeping track of all the work. If I were to do this project again, I would keep a list of all the variables that I checked, changed or created on the side so that I could always be fully aware what I have done, what I am doing and what I still need to work on. This would help me avoid repetitive work and careless mistakes.

## 3. Analysis techniques explored:

To this point, my measurements are as following and ready to be included in my models (variables in blue are the ones that I created and variables in black are the one that already existed in the original dataset):

Property: neighbourhood_group_cleansed + bathrooms + bedrooms + room_type + bed_type + beds
calculated_host_listings_count_entire_homes + calculated_host_listings_count +
calculated_host_listings_count_private_rooms + calculated_host_listings_count_shared_rooms
has_TV + has_kitchen + has_wifi + has_ac + has_gym + has_doorman + has_washer

Host/Management : minimum_nights + host_identity_verified + instant_bookable + cancellation_policy +
guests_included +availability_90+availability_30+availability_60+extra_people +
minimum_nights_avg_ntm + host_days

Past Clients Review: review_scores_rating + number_of_reviews + review_scores_accuracy +
review_scores_cleanliness + review_scores_checkin+review_scores_communication
+ review_scores_location+review_scores_value

Before applying any model, I first split the local dataset into training data and test data so that I could test my model even before submitting it to Kaggle. Following the examples in homework and in-class notes, I placed 70% of the local data in the training set and the rest in test.

I started with the linear regression model presented in the sample submission and got a really high RMSE at 105.56. Then I switched to decision trees as I thought this was the most complicated model we had learned in the class, and thus could work the best. I started with random forest, and then boosting. Boosting didn't improve my RMSE score at all, so I went back to random forest, adding cross validation. My final model is as following:

```
cvforest = train(price~
    minimum_nights+ review_scores_rating+bathrooms+bedrooms+
    number_of_reviews+host_identity_verified+instant_bookable+
    review_scores_accuracy+review_scores_cleanliness+
    review_scores_checkin+review_scores_communication+
    review_scores_location+review_scores_value+cancellation_policy+
    room_type+bed_type+minimum_nights_avg_ntm+
    neighbourhood_group_cleansed+ #listing_time+
    has_TV+has_kitchen+has_wifi+has_ac+guests_included+availability_90+
    availability_30+availability_60+extra_people+
```

```
  calculated_host_listings_count_entire_homes+
  calculated_host_listings_count+calculated_host_listings_count_private_rooms+
  calculated_host_listings_count_shared_rooms+has_gym+has_doorman+has_washer+beds+host_days,
data=train,
 ntree = 100,
 trControl = trControl,
 tuneGrid = tuneGrid)
```

And this model delivered the best RMSE score of 63.50826.

***Reflection:***

Running the model took much longer than I expected. I started with 1000 trees and it took hours for the code to run. I later switched to 100 trees and the process took much less time. Next time before I run the code, I should calculate how long the code would run in order to better plan the process.

Also, this time I didn't explore much of linear regression as I automatically assumed more advanced model might be more accurate. After reading the paper shared by the professor, I think linear regression can also be effective. In fact, any model might work really well if I prepare the data right. If I can do this project over again, I would try more models instead of sticking with the random forest model although it theoretically would yield a very good result as a bagging ensemble learning methods.