**Price prediction model for a specific Airbnb in Vienna (Austria)**

**Introduction**

A company who operates small and mid-sized apartments (usually hosting 2-6 guests) asked us to help them setting their prices for new apartments in Vienna, Austria. As there are currently many hosts in the city, the data from Inside Airbnb[[1]](#footnote-1) may contain false information, thus a thorough cleaning may be necessary to get a usable data frame to work with. Next, multiple models are used to determine which one may predict the actual worth of the company’s apartments the best, based on multiple prediction fit factors (such as RMSE).

**Data & Cleaning**

The dataset itself could be cleaner, many modifications had to be done. The most important one, is that many listings **have missing values for the price** variable. These rows have been deleted from the dataset, as filling them with any number, the models will use false information for its dependent variable and filling them any other values than an integer/float, will break the code. The dataset further filtered to only include **2 to 6 accommodates**, based on the company’s request and **Hotel rooms were excluded**. Many types of properties remained, thus more filtering was needed and now it includes **just apartment-like properties** (Rental Units / Condos). Flags are introduced for two review columns and the host columns are used mainly to get dummy variables for each listing. The reason behind the latter one, is that how the host “advertises” him/herself through Airbnb, how does the person care about his/her appeal in it, should have an effect on the price.

**Usable Models**

Using only one model has its danger, that the one we used may not be the best one among the possible models known to us, for this reason, three models are used to predict the price as best as it is possible. These models are the following:

1. Linear OLS Regression
2. Random Forest
3. LASSO

The reasons of the usage of the said models are the following:

**A.: Linear OLS Regression**

The OLS Regression let’s us to choose the variables from the dataset ourselves. While we have to make some important decisions in this model, by choosing which variables will be used, the code chunk *LinearRegression()* takes care the rest of the calculations and will give us back a simple RMSE value.

**B.: Random Forest**

Random Forest needs relatively less tuning than for the other prediction models, while more coding is needed in python, due its “Black Box Model” attribute. This model, unlike the OLS regression model, while giving all the variables that have chosen previously, will only use a randomly selected amount of it. It will combine the results of multiple trees and should give back a really good RMSE value. Due its Black Box attribute, feature importances will be also coded to get a better understanding of the model.

**C.: LASSO**

Because the dataset contains many columns/features (75 columns), the Least Absolute Shrinkage and Selection Operator model will select the variables in a way, to avoid overfitting, it tries to find the best measure.

**Models’ Results**

**A.: Linear OLS Regression**

**B.: Random Forest**

**C.: LASSO**

1. <http://insideairbnb.com/get-the-data/#:~:text=Vienna%2C%20Vienna%2C%20Austria> [↑](#footnote-ref-1)