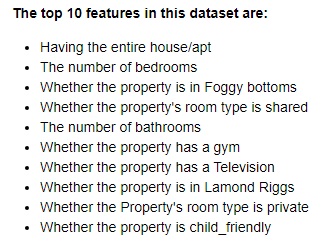
Daniel Park

DATA 606

Delivery 3

After looking into my data even more, I was able do a full data exploratory analysis on my data. I did this by graphing all my numerical columns to see what kind of trends were present. It was no surprise to me that almost all of them showed a skewed pattern, mainly because most of the properties share the same number of a certain feature (like bathroom, bedroom, beds, accommodations, etc. By looking at the graph, it would be hard to apply models to such data because of their skewed patterns. In order to make it more distributed, I would have to apply log transformation to each graph, which is exactly what I did. Once I did that, I applied One Hot Encoding, which basically converts categorical variables into a format that makes it easier for ML algorithms to read, which will be explained in my presentation. However, there were so many different types of variables in property\_type and amenities that I had to format them even more in order to make it easier to read and have more accurate results. Once all of the data was cleaned out, I created a Multi Collinearity Heatmap to see what kind of relationships other features have towards each other. This is important to understand because when two features show a strong relationship with each other then they are considered either positively or negatively correlated. This will evidently cause a skew in the overall output. By looking at the heatmap, I see that there are multiple strong relationships between features like bedrooms and price, price and guests\_included, etc.

After exploring the data and seeing its trends, I was finally able to start with my first model, which is XGBoost. XGBoost stands for eXtreme Gradient Boosting. It is a ML algorithm that uses decision trees to make predictions. In order to start, I had to apply One Hot Encoding to my dataset using the pd.get\_dummies() function. Once that was set, I set my X variable to my dataframe that did not involve the ‘price’ column, and my y variable as the price column itself. Once that was set, I split my data into testing and training, where the test size was 0.2. After running XGBoost I got a training MSE (Mean squared error) of .0014 and a test MSE of .0057. These are good numbers because it shows that the estimation and the actual values are very close to each other. For r2 (measurement of how close the data is to the fitted regression line), I got .9313 for the Training and .6984 for the test, which are promising numbers for the first model. Once the results came in, I went ahead and acquired the feature importances of the dataset. The purpose of feature importance is to show how valuable each feature was in order to construct the model that created the decision. I was able to create a chart that showed the feature importance of every single feature in my dataset. Through that I was able to find the top 10 features that gave the most importance. However, by changing the random\_state variable, the answers may change.

There’s no surprise that number of bathrooms, bedrooms, and having the entire house/apt to yourself is on the list because that is what people usually look at first when they look for property, especially if there are multiple people joining the Airbnb. So far I have only don’t XGBoost but for my final report I plan on applying 2-3 more neural network related algorithms instead of Hedonic Regression that I mentioned earlier. I feel like doing neural network type algorithms for a project like this would be more suitable than regression models.