# Airbnb Price Prediction

Georgetown University Certificate in Data Science
Cohort 19

Aisha Awan
Mulyono Kertajaya
Daniela Collaguazo
Jessica Saenz
Darien Thayne

June 20, 2020

# Table of Contents

# Abstract

New York City is filled with renters, but also high vacancy rates. Despite this, rent in the city continues to rise. According to the website Inside Airbnb “Income levels for the average New Yorker haven't kept pace, and affordability is at record lows. Housing is scarce; homelessness levels are increasing; food insecurity is growing; and economic and racial inequality rates in New York City are near the highest in the United States” (Inside Airbnb). This makes the perfect environment to succeed in short term rentals. Not only is tourism accommodation taken care of, but rentals without a commitment as well. While there has been a huge move against Airbnb by the New York State Attorney General due to the lack of taxes and regulations, many hosts in NYC currently rent out multiple properties.

The team wanted to see which factors would affect the daily price per night of a listing. The original Airbnb data for New York City came with 105 features and about 50,000 instances in 2019. However, as we cleansed the data, we focused on only 45 features for our model.

For Lasso and Ridge Regression Models, we used Grid Search to find the optimal alpha to choose. For Lasso, Optimal alpha was .001 and applying this alpha improved the r^2 score. For Ridge, Optimal alpha was 10 and applying this alpha had no impact on R^2. After optimization, the models performed identically.

For Random Forest Regressor we were searching for the optimal number of estimators, which random state to use, and minimum sample splits. The base model showed high Train r^2 but a much lower Test r^2. This likely means the model is over-fit with the base model. This is the main issue we need to resolve if we want to use this model.

For the last model we tried, Gradient Boosting Regressor, the parameter we were looking to optimize was the max depth. Base model showed better r^2 scores compared to the Lasso and Ridge, and was similar to the Random Forest Regression scores on the test R^2. Using this, we saw an increase in both R^2 scores, but it may have been slightly over-fit.

Furthermore, we are suggesting topic modeling to feature engineer text that might be relevant to predict price. We analyze both the descriptions and titles of the listings and find their latent topics, we further analyzed the importance of the topic scores in the model.

## Background and Motivation

Airbnb is a U.S based online marketplace that allows people to rent their properties or spare rooms to guests around the world. The company's business model is to act as a broker that receives commission from each booking from both hosts and guests.

It has opened up new opportunities for people that want to experience tourism differently. Similarly, the application also offers new opportunities for real estate owners that are considering Airbnb as an innovative way to generate revenue. Motivated by this reality, our team wants to work on a way to smartly assist new Airbnb hosts to come up with a price to charge per night so they are more successful in pricing their accommodation and subsequently, getting as much revenue as possible.

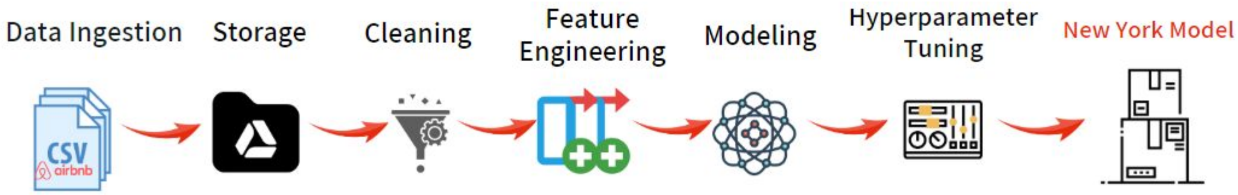
## Introduction

Airbnb began in 2008 when two designers who had space to share hosted three travelers looking for a place to stay. Now, millions of hosts and travelers choose to create a free Airbnb account so they can list their space and book unique accommodations anywhere in the world. Airbnb experience hosts share their passions and interests with both travelers and locals.

According to their website, "Airbnb is one of the world's largest marketplaces for unique, authentic places to stay and things to do, offering over 7 million accommodations and 50,000 handcrafted activities, all powered by local hosts. An economic empowerment engine, Airbnb has helped millions of hospitality entrepreneurs monetize their spaces and their passions while keeping the financial benefits of tourism in their own communities. With more than three quarters of a billion guest arrivals to date, and accessible in 62 languages across 220+ countries and regions, Airbnb promotes people-to-people connection, community and trust around the world" (Airbnb About Us).

While Airbnb started as local hosts renting out spare rooms in their houses, it transformed over time where hosts use the platform as a business opportunity. Data regarding Airbnb is available through the Inside Airbnb website. According to Airbnb's website, "Inside Airbnb is an independent, non-commercial set of tools and data that allows you to explore how Airbnb is really being used in cities around the world" (Airbnb About).

As part of the Data Science Pipeline, we took the data available for New York City in the Inside Airbnb website and saved it to Google Drive as part of Data Preparation and Infrastructure. The data was prepared and cleansed where the important features were selected. In the training and testing phase, the data was modeled and we completed hyperparameter tuning. For visuals, we primarily used the Yellowbrick library and Flask to demo. The architecture for our project can be seen below:



# Hypothesis

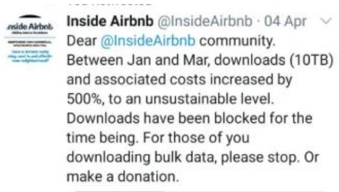
The team believes that there are many factors that will affect the price per night of a specific Airbnb listing. In real estate, if a property is priced properly, there is a greater chance it will be rented. We believe that by understanding the type of property, the location of the property, as well as having additional information regarding the property, we will be able to predict the price per night that will generate the most revenue.

Our team thinks that using input features such as location, days of booking availability per year, together with an analysis of the accommodations, a suggested price that will boost revenue for Airbnb hosts can be predicted. We believe that predictive machine learning algorithms such as regression will help suggest an optimal price per night that a new host can charge to maximize revenue.

## Data Sources, Storage, and Wrangling

On the Inside Airbnb website, there are 3 types of data available in .csv format for the team to use for New York City data. They have data about each specific listing, each review that was left, and a glimpse of how the hosts' calendar looks for the following year.

Since the data was in .csv format, the team was able to directly pull the data from the Inside Airbnb website. However, in the middle of our project, we noticed that the data was not available and our notebooks were not working. We came across the tweet below and realized that we needed a safer way to keep our data. And yes, we did make a donation!



Inside Airbnb @InsideAirbnb · 04 Apr

Dear @InsideAirbnb community.
Between Jan and Mar, downloads (10TB) and associated costs increased by 500%, to an unsustainable level. Downloads have been blocked for the time being. For those of you downloading bulk data, please stop. Or make a donation.

We decided to download the data once and add it to a Google Drive. We have since started pulling the data via code from Google Drive. To set the stage of our Git repository, we created and updated some of the core repo files including the License, Readme and a .gitignore files.

The simplified model we have to demo is stored on a Flask hosting site.

## Exploratory Data Analysis

### Data Examination

To start with the initial data examination, the team explored Airbnb datasets for three major cities in the United States: 1) Austin 2) Washington DC and 3) New York City. The idea behind this is to make an unbiased decision on what is the best and more complete dataset that will allow us to serve our purpose of predicting the price of a listing. After doing exploratory data analysis, the team decided to select New York City as the city we wanted to start using to train our model.

As mentioned above, we have Ingested and Wrangled with three different datasets. The **Listing data** included relevant information about each specific listing in New York City in 2019. We cleansed the Listing data to make it more usable. There were a handful of listings of boutique hotels and resorts that were an anomaly and removed.

The **Reviews data** included reviews that were left for a specific listing. We decided this would be a good dataset to complete Natural Language Processing and text analysis.

The **Calendar data** gave a glimpse into the future. This data reflects which dates in the following year the host has available for booking. We initially used this data in our modeling, however we realized this data is not real so cannot be used for modeling. For example, the Calendar data that was available was for 2020, however due to Covid-19, most Airbnb listings were cancelled for a few months. In fact, many hosts particularly in New York City are no longer renting their space.

We have also completed some exploratory data analysis on the **Subway system data** in New York City but were unable to tie it to the Airbnb data. We do believe this would be great information to include for future research.

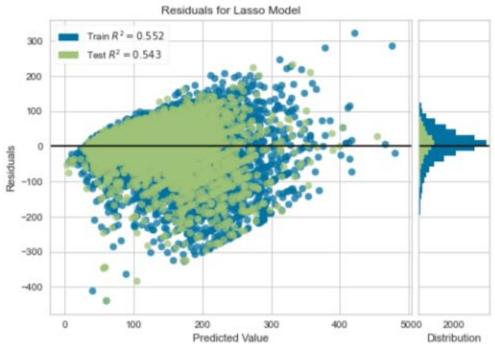
## Target Identification and Feature Selection

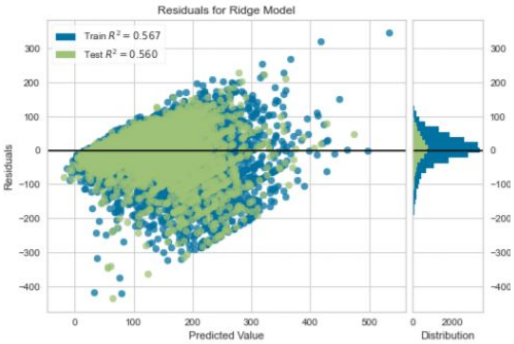
Due to these reasons, our focus was on the Listings data. Our work and research has been guided by the following criteria:

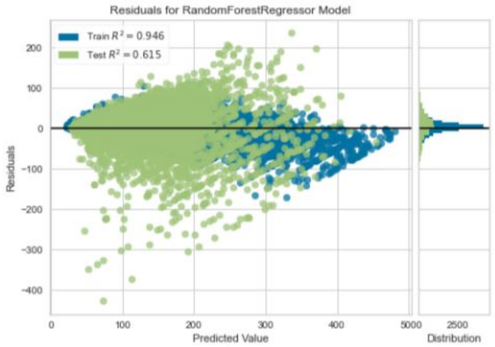
1) Determine which variables we would like to consider to do our prediction. The team considered each **instance** to be a specific Airbnb listing. The **features** we are considering using for our prediction include (but are not limited to): location (latitude/longitude, neighborhood), booking availability, price per night, amenities, cleaning service fee, review per month, and property type. Our target is the **price per night**.
2) To determine which features were to be used in the models, the team followed the following criteria. This can be seen inside the **variable_exploration** folder of the repo.
    1. Remove columns that have 50% or more null values.
    2. Remove columns with 0 variance.
    3. Remove columns that have only two distinct values, and one value counts for 95%+ of the total.
    4. For certain columns that have missing values, replace null values with the mean.
    5. Create columns out of a single column (i.e. Amenities).
    6. Delete columns that have a thumbnail URL or pictures.
    7. Performing Bayesian Average to normalize rating values so it takes into account the number of reviews to confirm the value of the rating per listing.
    8. Feature engineer over 10 columns to include relevant information that was not initially included in our dataset. This is primarily derived from existing columns.

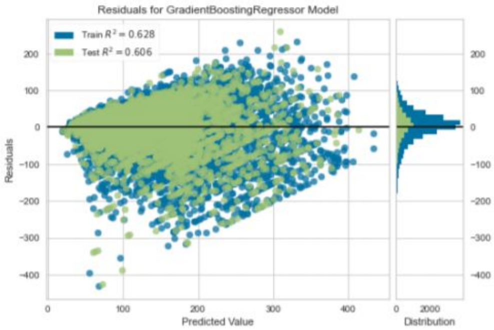
## Data Modeling and Analysis

Our data led us to regression modeling. We applied a few regression algorithms such as: Lasso Regression, Ridge Regression, Random Forest Regression, and Gradient Boosting Regression. The dataset was split into a test and training dataset, which was then used to run these regressions. Below you can see our base modeling:









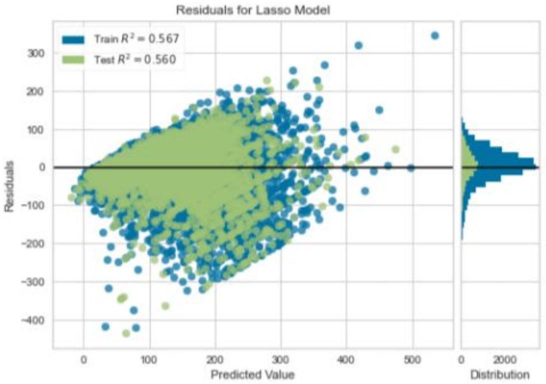
For **Lasso and Ridge Regression Models**, we used Grid Search to find the optimal alpha to choose.

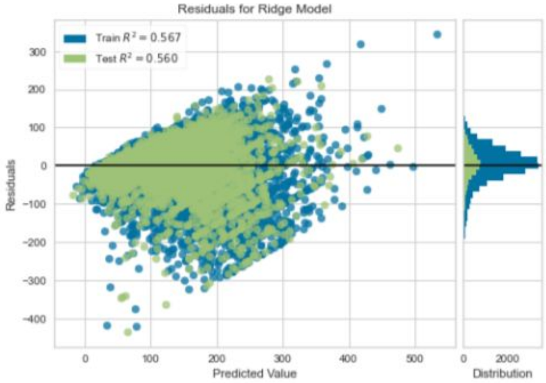
For Lasso, Optimal alpha was .001 and applying this alpha improved the r^2 score.

On Train Data, r^2 score went from .552 -> .567

On Test Data, r^2 score went .543 -> .560

For Ridge, Optimal alpha was 10 and applying this alpha had no impact on R^2. As you can see below, after optimization, the models performed identically.



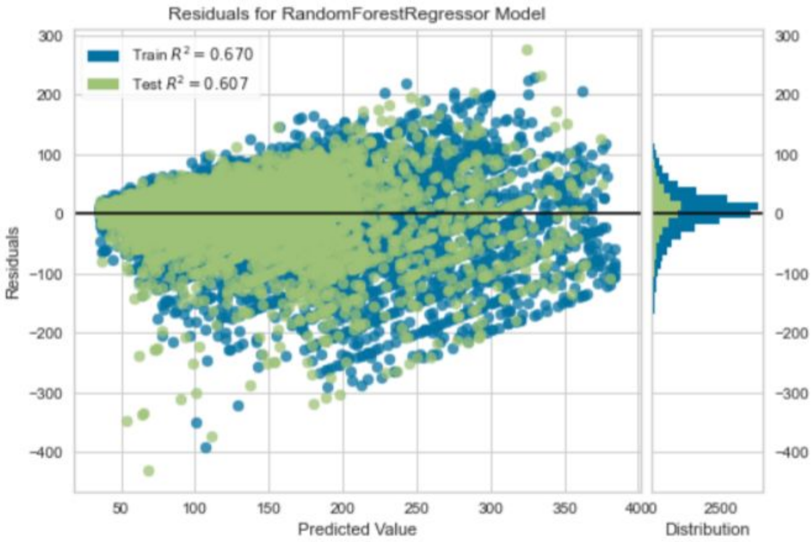


For **Random Forest Regressor** we were searching for the optimal number of estimators, which random state to use, and minimum sample splits.

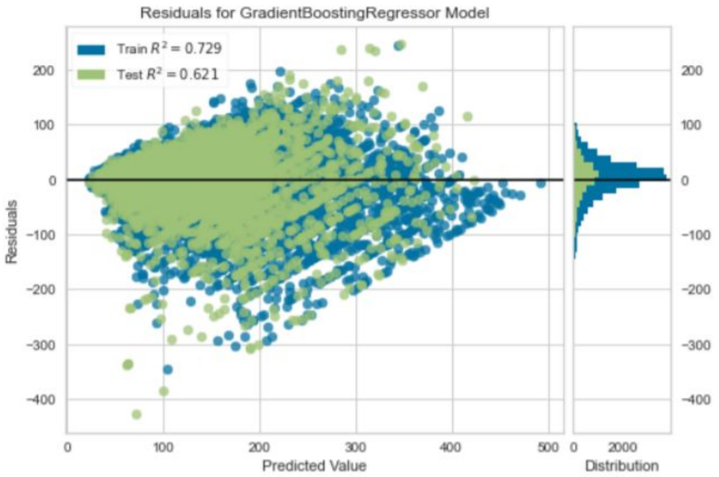
The base model showed high **Train** r^2 but a much lower **Test** r^2. This likely means the model is over-fit with the base model. This is the main issue we need to resolve if we want to use this model. Optimal parameters based on grid search we found were:

- n_estimators = 100
- random_state = none
- min_samples_split_100 = 100

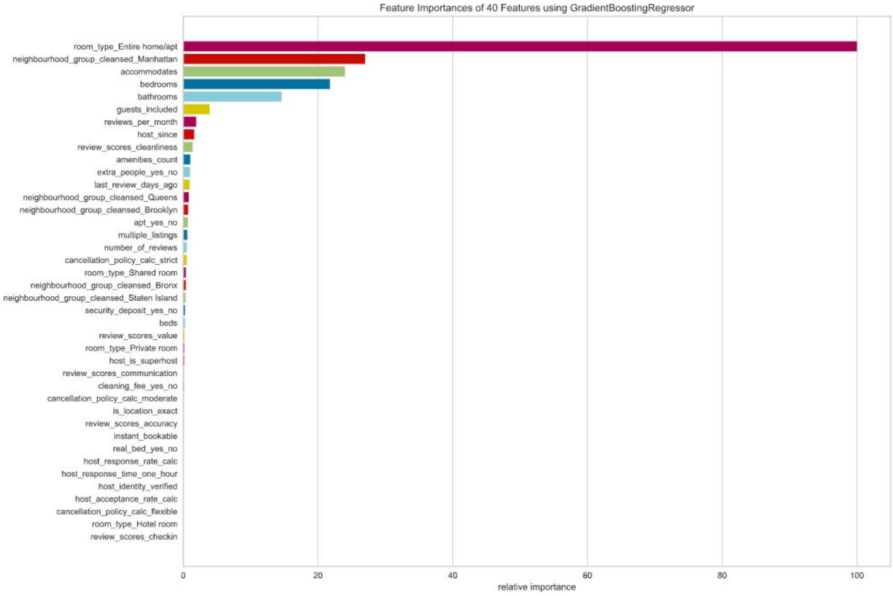
This still may be slightly over-fitted, however overall it is much less than before while maintaining a relatively good test r^2. See below:



For the last model we tried, **Gradient Boosting Regressor**, the parameter we were looking to optimize was the max depth. Base model showed better r^2 scores compared to the Lasso and Ridge, and was similar to the Random Forest Regression scores on the test R^2. Optimal max depth from the grid search gave 6. Using this, we saw an increase in both R^2 scores, but it may have been slightly over-fit. See below:



After hyperparameter tuning, we ranked the features in order of importance and that can be seen in the chart below. Our chart below shows that there are only a few features that had the biggest impact on the model. We have decided to use these features in our Flask application.



# Text Analysis - Natural Language Processing

We used the text available in columns such as “name” and “description.” We believe that there might be a relationship between the listing description and the price is because part of what makes a listing valuable is also the way the place is presented to its potential guests. This, not only includes the pictures of the listing, but also the words that the listing owners pick to describe their listing to seemingly seek to attract customers.

The purpose is to see if we can identify certain patterns in the text to determine if those patterns can influence the prediction of the listings price. If we see that words in the description and title of the listing are determinant of the price, we would recommend users some words to use or topics to write about when describing their listing to potentially increase their probability of getting a higher price.

We used an unsupervised probabilistic method for Topic Modeling to try clustering the information presented in both the description and the title in their assumed latent topics. The algorithm we are using is Latent Dirichlet Allocation which is an unsupervised model that infers topics from a collection of text documents. To get a better idea of this part please refer to the notebook “Airbnb-Price-Prediction2_Model_Building/5_text_analysis_name_description.”

## Data Preparation and Feature Engineering

To prepare the text for topic modeling we first concatenated both the name and description into a single column. Furthermore, we removed characters including numbers, we ran an algorithm to extract the text that was written in English only. We removed stopwords and finally performed lemmatization. One of the big challenges of working with unsupervised topic modeling is to come up with the right number of topics. We used two metrics together with our intuition and expertise on the matter, to determine the right amount of *n*.

The two metrics used were perplexity and coherence. According to (Jansen, 2018) *"Perplexity when applied to LDA, measures how well the topic-word probability distribution recovered by the model predicts a sample, for example, unseen text documents. It is based on the entropy H(p) of this distribution p and computed with respect to the set of tokens w. Measures closer to zero imply the distribution is better at predicting the sample".* In terms of Coherence, we will be using Gensim implementation based on the paper written by (Roder, et al., 2015). Larger scores of coherence mean better topic representation of the corpus.

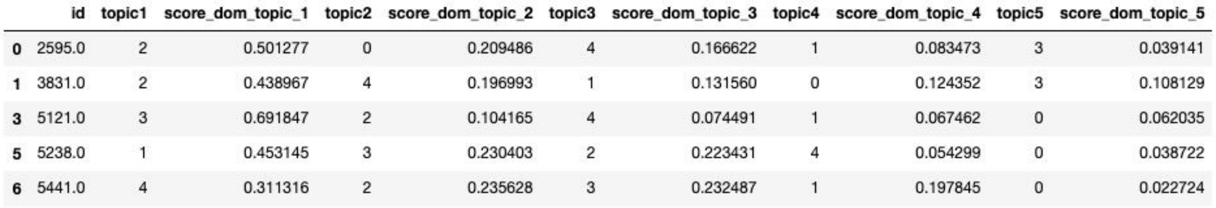
Based on lower amounts in perplexity combined with larger coherence we were expecting that the right number of topics could be between 5 and 9.  To come up with a final decision we used the LDA visualization tool pyLDAvis to draw the topics from three different models that have been fit in our corpus (link to visualization with 5 topics, link to visualization with 7 topics, link to visualization with 9 topics).  This visualization allows the user to see if the topics have a clear separation between each other. Overlapping circles might mean that we might want to join them instead of presenting them separated.

In the visualization for 5 topics, we can observe that there are five circles representing our ideal topic model for our corpus. Hovering over each bubble shows the most frequent and relevant words for each topic. According to (Carson and Shirley, 2014), relevancy is defined as the ranking method of the terms within topics. Whereas frequency is a measurement of presence of words throughout the corpus and thus will be found as top words in several topics.

Consequently, these “common” words will be penalized and score less on relevance metrics. However, those words scoring high should be more descriptive of the topics itself.

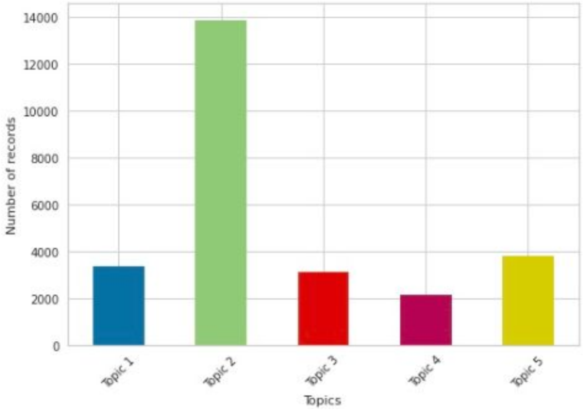
## Modeling with Topic Modeling scores

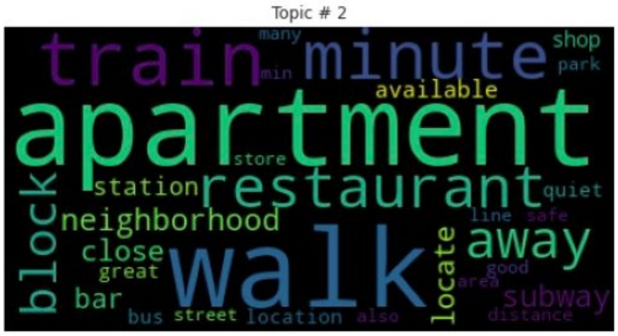
We created a dataset with all the descriptions with their corresponding topic scores. Here an example of the dataset that was formed with a five topic model.

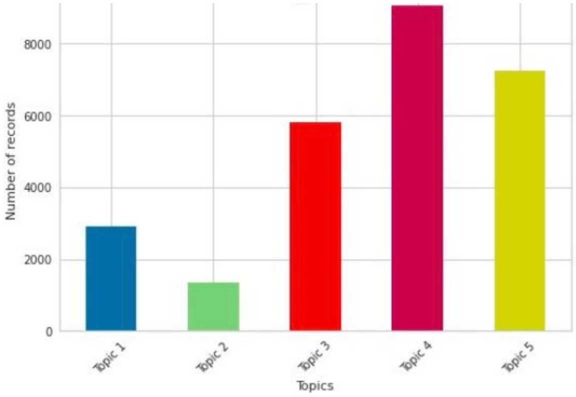


| | id | topic1 | score_dom_topic_1 | topic2 | score_dom_topic_2 | topic3 | score_dom_topic_3 | topic4 | score_dom_topic_4 | topic5 | score_dom_topic_5 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2595.0 | 2 | 0.501277 | 0 | 0.209486 | 4 | 0.166622 | 1 | 0.083473 | 3 | 0.039141 |
| **1** | 3831.0 | 2 | 0.438967 | 4 | 0.196993 | 1 | 0.131560 | 0 | 0.124352 | 3 | 0.108129 |
| **3** | 5121.0 | 3 | 0.691847 | 2 | 0.104165 | 4 | 0.074491 | 1 | 0.067462 | 0 | 0.062035 |
| **5** | 5238.0 | 1 | 0.453145 | 3 | 0.230403 | 2 | 0.223431 | 4 | 0.054299 | 0 | 0.038722 |
| **6** | 5441.0 | 4 | 0.311316 | 2 | 0.235628 | 3 | 0.232487 | 1 | 0.197845 | 0 | 0.022724 |

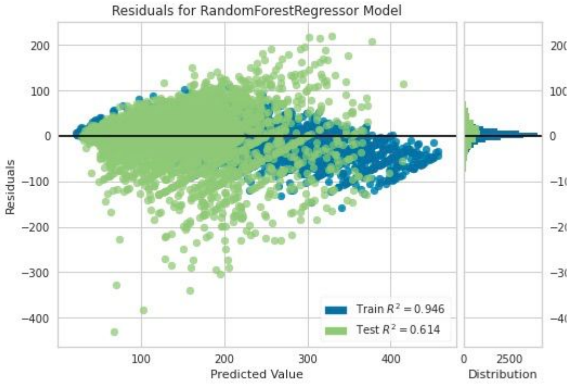
The column “score_dom_topic_1” represents all the topics with higher scores for that particular description. To know what was the topic that was more prevalent in the dataset, we would have to do a count of the topics in the “topic1” column. As we can notice in the graphic below, we see that the most dominant topic is Topic 2.  As the word cloud indicates, relevant words from topic 2 such as: apartment, minute, walk, restaurant, block, away, block, subway, are  a clear reference to location. Therefore assume that topic 2 together with other features that reference location should appear related or close to each other when calculating feature importance.





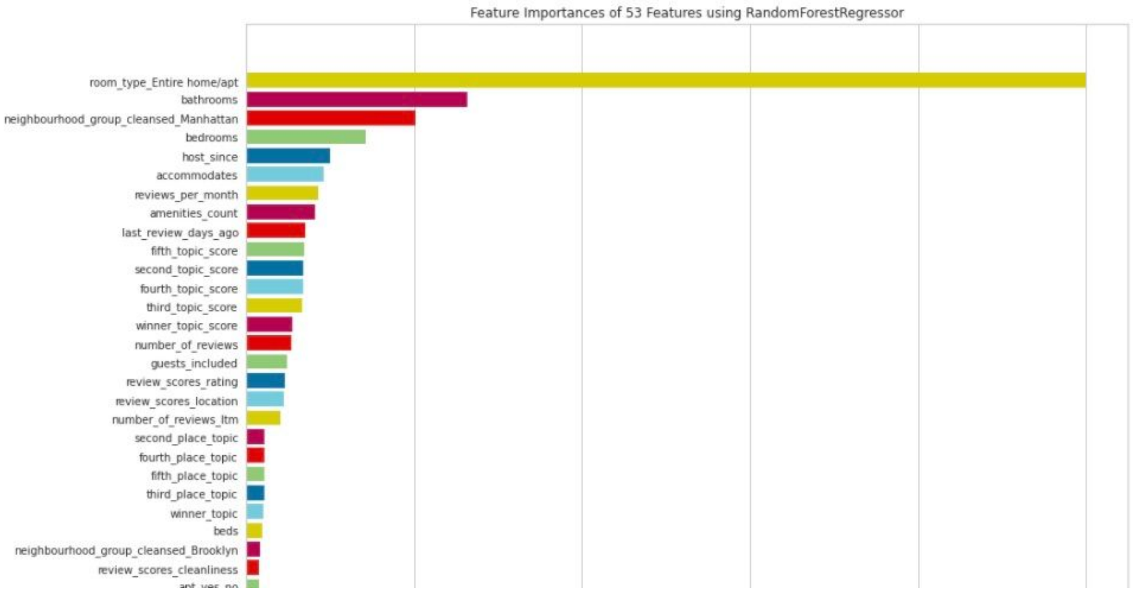


Conversely, to see the less prevalent topics in the corpus, we would have to count the occurrences of each topic, appearing in the "topic5" column. This barchart lets us know that the less prevalent topics in the corpus are topics 3, 4, and 5. When taking a more detailed look for words that are in topic five, we could say that this is the bucket where the leftover words go. Topic 4 seems to have many words that relate to sentiment such as love, feel, be, travel.

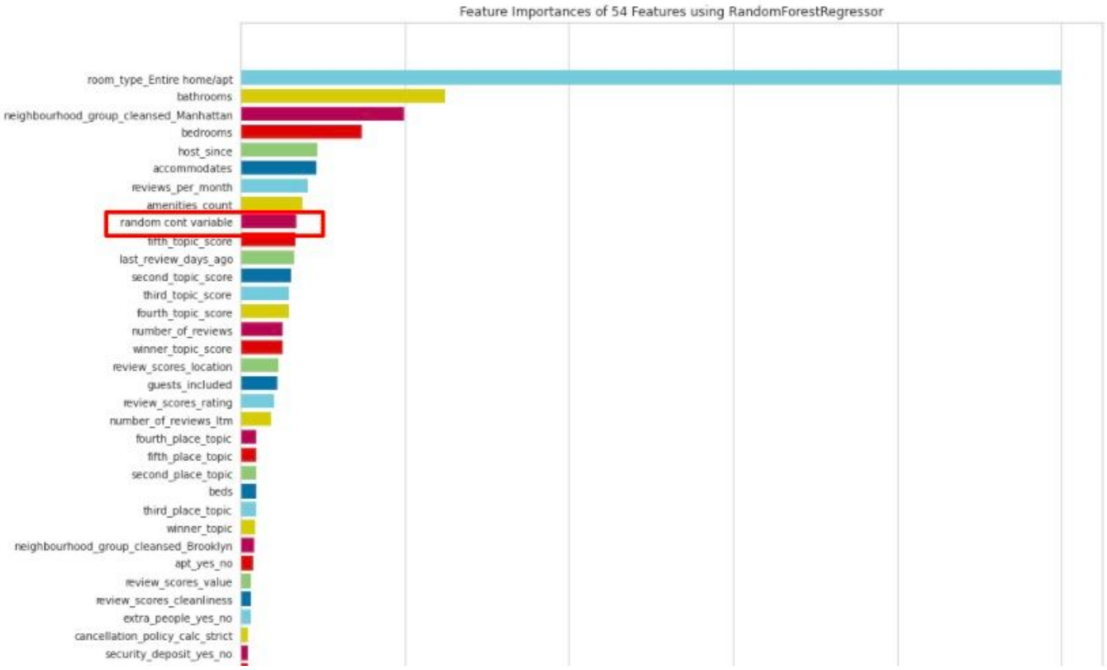


We then merged the dataset with the rest of the features with our topics and topic scores for our model with five topics. We ran multiple regressions such as Support Vector Machine, KNeighbors, Bayesian Ridge, Random Forest Regressor, etc. Our winner base model was Random Forest with a train r^2 score of 0.946 and a test score of 0.614. A highly overfitted regression. However, before doing any hyperparameter tuning we wanted to take a look at the most important features.

As we can see in the figure below, it is apparent that the topic scores all appear to come over the top. Even above already determined highly important features in other models such as number of guests ("guests_included").



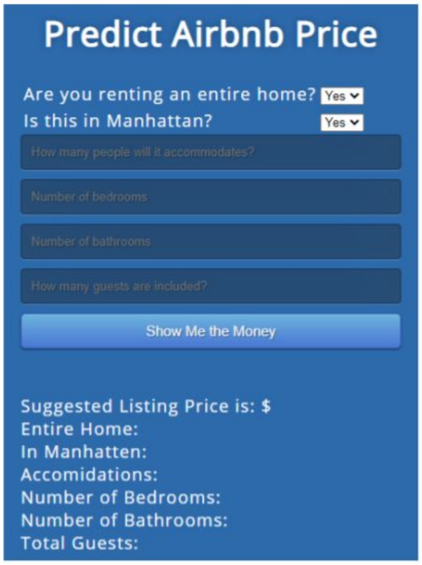
After noticing this we thought that a potential reason why the scores were so relevant, would be because of the nature of the scores, meaning: they are continuous values ranging from 0 to 1. We also know that many of the other features are mainly discrete values. To make sure that the scores were included as part of the most important features was not because of the nature of the scores (numbers), we decided to create a random variable from zero to one and perform the same regression analysis and feature importance to determine how this variable will get placed. Surprisingly we learned that this random variable got also placed at the top of important features.



This leads us to question our validity of the hypothesis that topic modeling is the way to feature engineer our text. We believe that topic modeling might be too broad of an approach and we might want to go a little bit more in detail in terms of word scoring, not group of words scoring. We are therefore suggesting not to include the topic scores as part of our application before exploring with other algorithms such as Term Frequency or maybe even TF-IDF which are approaches that would provide with a word by word scoring.

## Data Product (Flask)

Once we completed each stage of the data pipeline with the New York City data, we were then able to stitch together a pipeline procedure so we could expand our testing to additional cities. The Lasso and Ridge Regressions performed identically after optimization so we decided to use the Lasso Regression with the optimized alpha alongside both the Random Forest and Gradient Boost Models.



Predict Airbnb Price

Are you renting an entire home? Yes

Is this in Manhattan? Yes

How many people will it accommodates?

Number of bedrooms

Number of bathrooms

How many guests are included?

Suggested Listing Price is: $
Entire Home:
In Manhatten:
Accomidations:
Number of Bedrooms:
Number of Bathrooms:
Total Guests:

To display our model in an end-user friendly way, the team has implemented a demo in Flask. We selected the top 6 features based on our Feature Analysis and when the user inputs data about their potential listing, the application will tell us how much to price an accommodation per night.

Though this app is interesting, and works as expected, without further testing in the field, we won't be able to verify its accuracy in the real world.

Link: http://seebeyond.pythonanywhere.com/

## Lessons Learned and Future Research

Lessons Learned:

- Scope out the work and stick to the scope! While we did scope out our work, we had a very wide scope and time was just not permitting. Everyone had more-than-usual busy days at work and home, and we initially tried to do everything at once. This made for us not being able to go in depth on specific things the way we wanted to.
- While data may be readily available in .csv format, it does not mean that our work as a Data Scientist is clear cut. There is still a ton of work to be done.

Questions for Future Research:

- Does being close to a train station or bus stop increase the price per night by a considerable amount?
- Is there a way to see if bookings are more or less likely to occur based on the weather on a certain date? Does weather impact the price per night?
- How can we use our NLP to predict the text a host should put as their Listing Title and Listing Description?

## Bibliography

Inside Airbnb. New York City. Retrieved from http://insideairbnb.com/new-york-city/

Airbnb. About Us. Retrieved from https://news.airbnb.com/about-us/

Airbnb. About. Retrieved from http://insideairbnb.com/about.html

Jansen, S. (2018). Perplexity. In Hands-on machine learning for algorithmic trading: Design and implement investment strategies based on smart algorithms that learn from data using Python. Birmingham: Packt Publishing.

Roder, M., Both, A., and Hinneburg, A. (2015). Exploring the Space of Topic Coherence Measures. In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM ’15, pages 399–408, New York, NY, USA. ACM.

Sievert, Carson and Shirley, Kenneth E. (2014). LDAvis: A method for visualizing and interpreting topics. In Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces, pages 63–70, Baltimore, Maryland, USA.