Data Science Final Project

Author: Lee Waxman

Introduction

Airbnb provides a platform for house owners to accommodate travelers who search for alternative hospitality. Since it was established in 2008, Airbnb has seen enormous growth, as more and more people decided to use its website and rent a house for their vacations. Berlin is a popular tourist city, which attracts people from all over the world, and has more than 22,552 listings as of November 2018, on the Airbnb website.

This project will discuss the most important thing for a host who publishes his property on the website. How many days his property will be rented during the month. The number of days that the property would be rented has a tremendous impact. More days the property is rented in the month means more money for the host. Furthermore, a low number in the listing means low cost on maintenance. Hence, the goal of this project is to predict the number of days in a month a property will be rented three months from a given date, meaning what the data about a property today can tell me about the renting numbers in three months from now. To do that I would have to search for the variables that can affect my outcome, the number of days that the property will be rented in a month.

As a traveler, I know what will affect my decision when choosing accommodation for my vacation. I will prefer a property with specific facilities, for example, air-conditioning in the summer, and hitting in the winter. The location is also very important because I want to be as close to the main attraction as I can. Moreover, I want the price to be fair, so I will search for different options that are available on the same date, and compare them.
When I search for accommodation the description of the property is very important because it helps me to decide whether or not it is a good fit for me. Also, I like to see reviews from previous guests.

It is common knowledge that during the holidays and summertime more people are going on vacation. However, the data set that I am using in this project doesn't include information about holidays. Therefore, I would have to check if there is a change in the number of days a property is rented during the year, and if some months show an increase in rented days than others.

According to my personal experience, that was described above, I will try to answer the following questions :

- Does the neighborhood have an impact?
- Do reviews from the previous months have an effect on how many people want to rent the property ?
- Does the description of the property affect the price?
- Does the rent date have an impact?

By answering those questions I will build a new data set that will help me to predict the number of days a property will be rented.

Methodology (Project design)

Data

The dataset contains: data about the property, reviews guests write about their experience, and data of Airbnb listings in Berlin, starting from November 07th, 2018 and until November 08th, 2019.

The data set “calendar summary” contains: an id of the property, a price of renting the property for one night, a date, and if the property is available at this date. The availability of the property is marked with “t” if it is available and “f” if it is not. Since the goal of this project is to find out what is the sum of days in a month a property will be rented, meaning it won’t be available, I used this data set to sum all the days in a month that a given property is rented. This calculation created for me the outcome for this project - “number of days was rented” - the number of days in a month a property isn't available.

The “listing summary” data set contained information about the property. Most of this information is text. Therefore, I had to do text analysis and created a bag of words. I didn’t use “summary”, “space” and “description” since they didn’t give me any added value. Those three variables give information about the property, like the number of rooms, if it is on the first floor, and in which neighborhood the property is located. All this information was available to me in other variables, and in a more organized way.
After performing the text analysis on all the variables, I found out that several of them created a big set of words, even when using only the English language, which is the most common one. Therefore, I decided to also drop “space”, “neighborhood-overview”, “notes”, “transit”, “access”, “interaction”, “house rules”, “host-about”.
The variables that created a reasonable bag of words that I could use are “bed_type”, “host_verfications” and “amenities”.
The “listing summary” data set also contains the location of the property, which includes the neighborhood name, the streets names, and the latitude and longitude. Since the latitude and longitude are representing a geographic point in the map, I decided to calculate that point and put it in the data instead of them.
Other information that can be found on the “listing summary” data set is about the host, his name, his id at the system, and where he lives. This is important information because some of the hosts don’t live in Berlin or not even in Germany, which can affect the decision of a traveler.
I also learned from the data set if the host has a profile picture, does he have a website, does he have a license, which kind of license. I found information about the number of rooms, bathrooms, beds, and accommodation. All are included in the flat file.

The “reviews summary” data set contained guests' comments about the property. The comments were very relevant since travelers usually look for other people's opinions before they make a decision. Because of the experience I had with the text analysis at the “listing summary” data set, which created a huge bag of words that was impossible to work with, I decided to do a sentiment

analysis. The idea is that the more good reviews a property has the bigger the chance it will be rented. Sentiment analysis got me exactly that, the number of good reviews.

I had other three data sets: “listings”, “neighborhood”, and “reviews”. All of them had the same information as the three above, and therefore I didn’t use any of them.

After I finished collecting all the relevant information and created a flat file, I started to search for a correlation between the variables, not including the outcome variable which I left for a later observation. Correlation between variables can create bias in the data which can affect the models. There was an interesting correlation between some of the words from the bag of words I created in the formar stages. Those correlations discover word pairs that were probably separated at the text analysis. Moreover, the correlation calculation also discovered words that are usually put up together, for example “oven”, and “refrigerator”, that describe a full equipped kitchen, also showed a significant correlation between them.
The variables “availability_30”, “availability_60” and “availability_90” showed a connection between them as well.
After the correlation was checked, I tested the differences between the correlated variables. I used two tests, wilcoxon for the continuous variables which didn’t have normal distribution, and the kruskal test for two categorical variables.
I found that “dishes” had a significant difference from “stove”, “cooking” and “basics” , all words were created from the variable amentis. Although, “stove”, “cooking” and “basics” didn’t have any difference between one another. Therefore, I removed only “cooking” and “basics”. In addition to that I also found that “private” and “room”, which were created from the variable “bed_type”, had a connection between them, and showed a significant difference. I left both of them in the data set.

My exploration data strategy was to separate the data for numeric and object variables. For the numeric variables I searched for outliers. I used boxplot to see in which variables I can find outliers, and in which variables there are no outliers. This approach helped me to make the list of variables a little shorter, and therefore the calculation took less time. In order to find the outliers I calculated the IQR, using the “describe” method in python to get the 25% and 75% for every variable. After I had the IQR, I used the following equation to get the bottom border of the variables and the top border.

top_board = data[75%] + IQR * 1.5
bottom_board = data[25%] - IQR * 1.5

When I had the outliers I drew a distribution graph for each variable to see if it had changed when the outliers were removed. I did the same to check to see if the correlation with the outcome had changed after removing the outliers. You can see the results at the result chapter.

The search for the missing data started with creating a heatmap that marked with white all the places where data was missing. This gave me the big picture about how much data is missing. The next step was to find the variables which have more than 80% of their data missing. Those variables were deleted from the file. The other variables had to be treated. For the numeric variables I planned to use static methods and KNN to fill in the missing values. For the text variables I planned to

transform them into integers and give the missing values a matching number. You can see the result in the result chapter.

When I finished with the outliers and missing data I ran a calculation of every variable with the outcome. I used different tests for continuous variables and ordinal variables. For continuous variables I used a pearson test, and for ordinary variables I used a spearman test. I used the results to get the relevant variables, so I could run my models in the most efficient way.

Models

My final data set contained 6378 rows, I thought this is too small for three parts - train, validation and test. Therefore, I had decided to split it into “train”, which contained 80% of the data, and test, which contained 20% of the data.

My outcome is a continuous numeric variable, hance, the models that was the best feet are: Logistic regression, decision tree, random forest, ada boost regression, gradient boosting machine (GBM) and support vector machine (SVM). I run every model separately and calculate for each of them the following regression metric: “RMSE”, “MSE”, “MAE”, “MSLE” and “RMSLE”. “RMSE” and “MSE” were more important to me than the others since, as you will see at the result section, I removed all my outliers, and those metrics are better feet in this case. I used those metrics to find the best model to predict my outcome.

Another method I used to make my model a little more accurate is to combine between models which had good results in their matrices. I also used different algorithms such as “SVM”, “Lasso” and “Random Forest” to select the best variables for my models.

I check the accuracy of the models with the test part, and calculate the percentage of the true positive answers the prediction gave. You can see the results in the result chapter of this project.

Deployment of your model

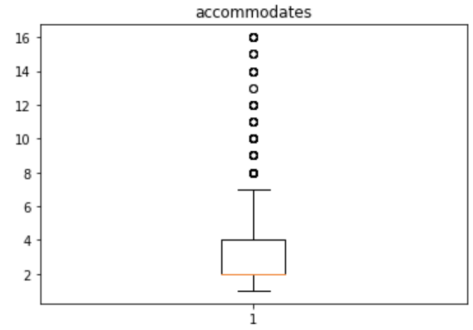
The model that was created in this project is based on data that was collected during one year only. This is not enough data to create a strong model. Therefore it is highly suggested to test this model in another year, to see if the accuracy doesn't change.
For testing this model I would recommend to use only the variables I found that have the best correlation with the outcome. The variables are : 'id', 'host_id', 'accommodates', 'guests_included', 'minimum_nights', 'maximum_nights', 'availability_90', 'availability_365', 'number_of_reviews', 'calculated_host_listings_count', 'tv', 'cable', 'heating', 'friendly', 'smoke detector', 'essentials', 'shampoo', 'lock bedroom door', 'hangers', 'iron', 'laptop workspace', 'living', 'bathtub', 'microwave', 'coffee maker', 'dishes', 'stove', 'allowed', 'long term stays', 'washer', 'internet', 'buzzerwireless intercom', 'carbon monoxide', 'fire extinguisher', '24hour', 'checkin', 'pets', 'free parcking', 'translation missing', 'host greets', 'premises', 'high chair', 'babysitter recommendations', 'crib pack n playtravel', 'roomdarkening shades', 'oven', 'entrance', 'ethernet connection', 'smoking', 'patio', 'balcony', 'wheelchair accessible', 'suitable events', 'lockbox', 'keypad', 'doorman', 'breakfast', 'pool', 'kettle', 'reviews', 'facebook', 'selfie', 'google', 'private', 'room', 'price', 'extra_people', 'sentiment', 'reviews_per_month', 'number_of_days_rented', 'reviews_per_month', 'number_of_days_rented'.
After creating the data with those variables it is possible to remove the rows which contain outliers.
Missing data should be replaced only for the text categorical variables, by switching the text to numeric values and giving the missing values a dedicated number.
When the data is prepared, and there are no outliers and the missing values were treated, the model Decision Tree should be run on this data. Any accuracy that is lower than 80% will suggest that something is wrong with the model, and the models should be all tested again with the new data that was created.

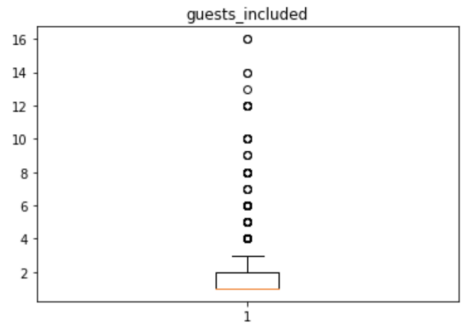
This model can serve new and old hosts that want to know how many days their property will be rented in three months from a given date. The variables they need to take into account are mentioned above. For the model to work in its best way, it's better they will not miss any variable, and try to fill them all.
The hosts will get a csv file that will contain a column of month, year, and number of days the property will be rented in the mentioned month. The prediction will suggest the number of days in a month the property will be rented, and the host should act as he pleases according to those suggestions.

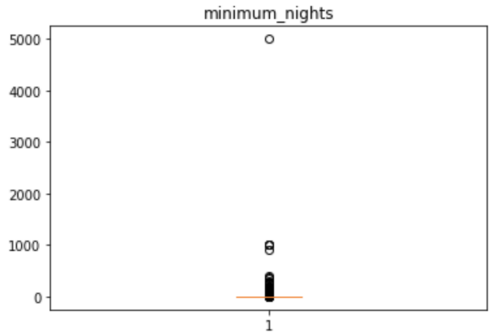
Since the tourist world is stable for the most part, I would recommend updating the model every year, and running it on a one year data. I am not taking into account events like the coronavirus, which affected tourism around the world. In those cases the model is not effective.

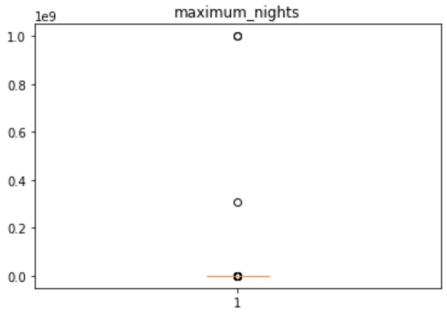
Results

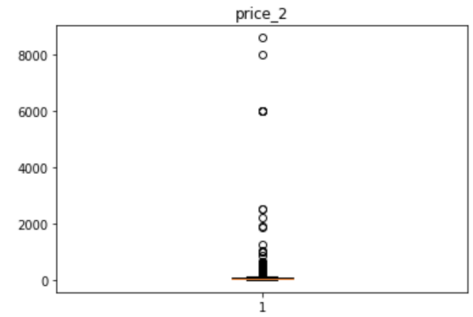
The final file, after analysis, was 10334 rows and 215 columns.
I used this data to find outliers in the numeric variables. I found that “accomodates”, “guests included”, “minimum nights”, “maximum nights”, “number of reviews”, “calendar hosting count”, “price”, “extra people” and “sentiment” had outliers.

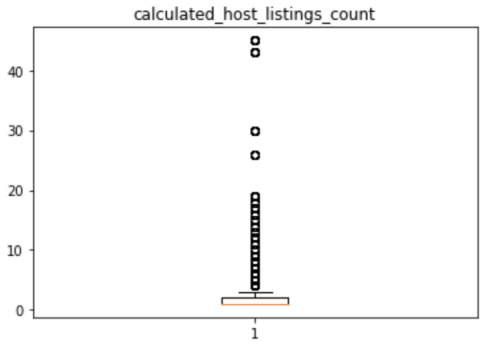


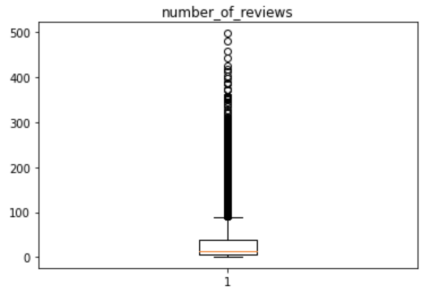


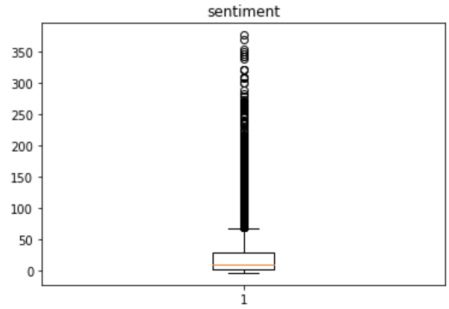




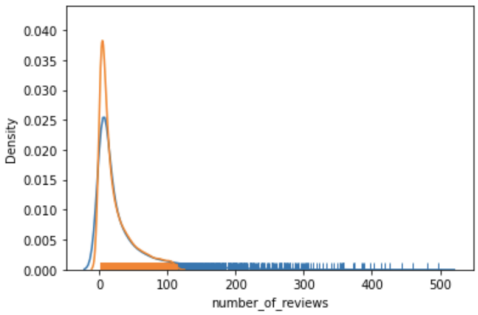


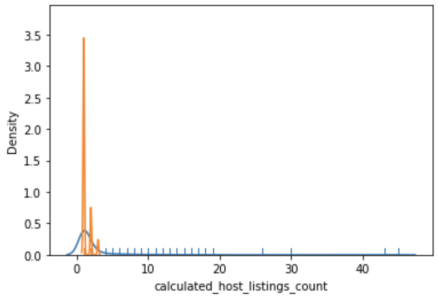






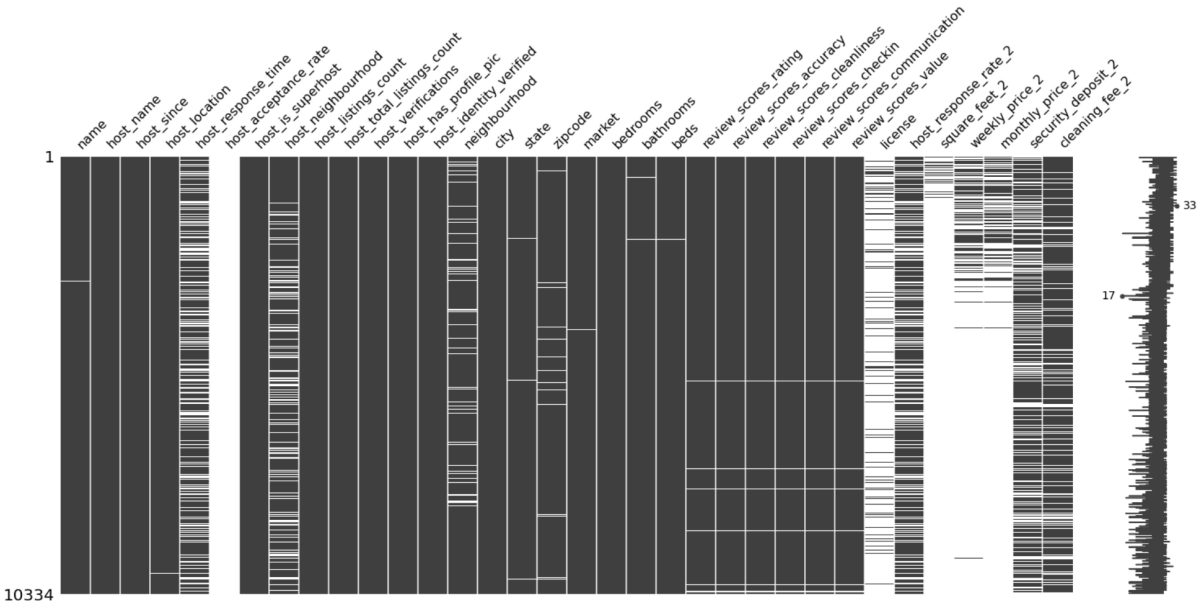
For each variable that had outliers I checked the differences in the distribution and the correlation with the outcome when the outliers are part of the variable and when they are removed. It was very clear that the distribution was changed in every one of the variables (example below).





However, the correlation of all the variables with the outcome didn't change in any of the cases. Hnce, I decided to remove all the rows with the outliers.

After treating the outliers I searched for missing values. I created the graph below.



As it can be seen from this graph “host acceptance_rate” has no values, and it seems that there are a lot of missing values in the variables “licences”, “square feet”,” weekly price” and “monthly price”. For every variable in the file I checked if it had more than 80% of missing values. This had verified that the variables “host_acceptance_rate”, “license”, “square_feet”, “weekly_price” and “monthly_price” really have a large amount of missing values, and I decided to delete them from the file. Others had some missing values but in a much lower percentage. I split those variables into two groups: one is the continuous numeric group and the other is categorial text group.
For the first group I used a statistical method and KNN algorithm to try and fill the missing values. Both of the methods didn’t work for my data. Both of them created a list of zeros for each variable. Because of that, and since the percentage of those variables in most cases is less than 5%, I had decided to live them as they are.

	0	1
0	name	4.0
1	host_name	1.0
2	host_since	1.0
3	host_location	26.0
4	host_response_time	3150.0
5	host_acceptance_rate	10334.0
6	host_is_superhost	1.0
7	host_neighbourhood	2083.0
8	host_listings_count	1.0
9	host_total_listings_count	1.0
10	host_verifications	1.0
11	host_has_profile_pic	1.0

12	host_identity_verified	1.0
13	neighbourhood	815.0
14	city	4.0
15	state	55.0
16	zipcode	268.0
17	market	22.0
18	bedrooms	7.0
19	bathrooms	16.0
20	beds	12.0
21	review_scores_rating	166.0
22	review_scores_accuracy	168.0
23	review_scores_cleanliness	168.0
24	review_scores_checkin	171.0
25	review_scores_communication	170.0
26	review_scores_value	172.0
27	license	8966.0
28	host_response_rate_2	3151.0
29	square_feet_2	10009.0
30	weekly_price_2	8604.0
31	monthly_price_2	8769.0
32	security_deposit_2	3084.0
33	cleaning_fee_2	1814.0

For the second group I had another approach. Since data in this group was text, I decided to transform it to ordinal data and give the missing values a numeric value.

At this point of the project the size of my data was 6378 rows and 210 columns. I ran a correlation test on this data between every variable and the outcome. The goal is to get only the variables that had a significant correlation with the outcome, meaning, the p-value of the test was equal or lower than 0.05. After that the number of the columns in the data was reduced to 121.

For the models part of the project I splitted my data to train and test groups. The train group had 5103 rows and the test group had 1276 rows.
At first, I ran the models on all the 121 variables which had a significant correlation with the outcome. The result was as followed:

	MAE	MSE	MSLE	RMSE	RMSLE	model
1	0.000000	0.000000	0.000000	0.000000	0.000000	Decision Tree
2	0.734428	3.180663	0.154636	1.783441	0.393238	RandomForest
4	3.474309	38.979444	0.894438	6.243352	0.945747	GBM
3	4.710172	53.709016	1.228200	7.328643	1.108242	ADABoost
0	5.411394	63.265022	1.368501	7.953931	1.169829	Linear Regression
5	11.193891	253.493566	2.710861	15.921481	1.646469	SVM

As you can see from the table above, random forest and decision tree gave the most satisfying results. However, the decision tree gave a little too perfect result, which made me suspicious that it may be over- freating. Therefore, I decided to create another model that combines the random forest and the decision tree. The result is represented at the table below:

	MAE	MSE	MSLE	RMSE	RMSLE	model
0	5.411394	63.265022	1.368501	7.953931	1.169829	Linear Regression
1	0.000000	0.000000	0.000000	0.000000	0.000000	Decision Tree
2	0.734428	3.180663	0.154636	1.783441	0.393238	RandomForest
3	4.710172	53.709016	1.228200	7.328643	1.108242	ADABoost
4	3.474309	38.979444	0.894438	6.243352	0.945747	GBM
5	11.193891	253.493566	2.710861	15.921481	1.646469	SVM
6	0.367214	0.795166	0.068967	0.891721	0.262615	Decision Tree + Random Forest

As it seems from the table, the combination of the two models is better than the random forest alone. I tested the accuracy of both the combined model and the model of the random forest on its own. The combined model was accurate by **68%** while the random forest was accurate only by **61%**. However, despite my speculation it seems that the decision tree gave the best result. It was **82%** accurate.

I tried to improve my model by using feature selection methods. Using these methods redused the column number to 29. The feature selection algorithm is supposed to get the variables that will predict the result in the best way.

Here are the results of all the models after running them on the improved data.

	MAE	MSE	MSLE	RMSE	RMSLE	model
1	0.000000	0.000000	0.000000	0.000000	0.000000	Decision Tree
2	2.034329	10.429522	0.406915	3.229477	0.637899	RandomForest
4	5.096778	64.038375	1.053036	8.002398	1.026175	GBM
0	5.838353	72.200370	1.398125	8.497080	1.182423	Linear Regression
3	6.253927	73.074865	1.357088	8.548384	1.164941	ADABoost
5	9.376546	169.008915	1.593938	13.000343	1.262513	SVM

Comparing those results with the ones we got when all the variables (that passed the correlation test) were present, we can see that random forest and decision models are still the best model for this data. However, it seems that the feature selection didn't do any improvement. Furthermore, when I tested the accuracy of those two models and the combined one on the new data I discovered that it was less accurate than the last one. In this case the random forest was accurate in only 36% percent of the time. The combined model was 56% percent accurate, and the decision tree decreased to 47 %.

Therefore, I think that the best model to use is a **decision tree** that gets all the variables from the correlation test, and ignores the result from the feature selection because in my case it didn't contribute to the project.

Conclusion

The project is the final task in a Machine learning course. I chose the Berlin Airbnb data set over other data sets that were represented to us since I wanted to discover the factors that affect the chances of a property to be rented.
As a traveler myself, I have my own reasons why I would choose one property over the other. I used my subjective decisions of choosing a property to lead me in this project, but I was careful not to follow them completely. I am one person who travels the world among lots of people who enjoy doing so, and my decisions are not the only ones. This was the first challenge in the project, not to be deceived by my own preferences. The second challenge in this project was to understand the data. What would be useful to me, what kind of variables will make my data too messy, and which variables need further analysiation.
At first I read every data set separately, and tried to get an idea of what it contained. Secondly, I tried to find the connection between the different data sets. I discovered the three of them are merely a fraction of the others, and didn't contain any added value. This made my life a little bit easier, because then I could focus my attention on the other three.
Another problem I had was that a big part of the data was long text. Using text analysis didn't help since it created a big bag of words. Since this kind of data can make my file really heavy and messy, I had to find the information represented in those texts in other places and to think how to compensate for data that is lost.

After analysing the data and creating the flat file I had to deal with outliers and missing values. With the outliers I experienced no difficulties. With the missing values I had some difficulties since I couldn't find a good method that will help me to fill in the missing values. For the continuous variables I had to leave the things as they are, for the categorical variables, which in most cases were texts, I transformed them to integers and gave the missing values a number of their own.

The final step was to run the models, and find the best model for my data. As can be seen from the result chapter of this project, I succeeded in finding one - a decision tree.

It is important to put in mind that the data in this project didn't take into account holidays. Although you can see differences between the months, this kind of information would have changed the prediction, and would have made it a little more accurate. Another thing that it is important to remember, is that this data was taken before the coronavirus had struck the world. Unfortunately, I don't think this model should be used in today's situation. The coronavirus has changed the tourist view around the world. Less people are leaving their own country, and if they do leave, they are looking for different properties than before the coronavirus.