Springboard Data Science Capstone Project

Airbnb Yearly Revenue Prediction

Ashley Jiangyang

Mar, 2020

Table of Contents

1. Introduction

Since its founding, Airbnb has hosted over 60 million people in 34,000 cities across the world and is continuing to grow quickly. Airbnb provides an alternative source of income for people who have otherwise vacant properties and for guests looking for affordable and convenient housing options. With any service, trying to monitor and understand the underlying attributes dynamics of Airbnb and market trends is very important both for hosts and the entire housing market. As users continue to grow on both the supply and demand side, it becomes crucial to understand and predict how much can a listing earns on the platform.

We seek to analyze over 3,800 listings in the Seattle area in order to better understand how the use of listing attributes such as host activeness, accommodates, location, ratings, and more can be used to accurately predict the optimal earning for the host and those who want to join the platform to become hosts. With better-earning suggestion estimates, Airbnb hosts can improve their management of properties to gain better earning, also the information from the analysis would be insightful for the whole housing market. The objective of this project is to build a model that predicts the earning of a property taking into account listing features and attributes. The end goal is so hosts and the platforms can understand what features of an Airbnb listing are most important to managing and invest for greater yearly revenue.

Answering the question would be a benefit for both potential hosts and any home-sharing platform, that info will be very useful guidance to make the investment as well as regulate the listings market.

2. Data Collection and Wrangling Summary

The detailed data we use listings.csv (22 February, 2020) was acquired from the Airbnb official website, check the official website for more detailed information. This data set contains 7,544 Observations and 106 features regarding the listing attributes. Each row is an entry of a listing, and the columns describe the information such as host reaction(host response rate/time), listing properties(property type, room type, bathrooms, bedrooms), and review score. etc.

In this section, we will walk through the steps of data wrangling with the summary to demonstrate how to we achieve a cleaned dataset for further analysis.

2.1 Drop Features

51 meaningless features are dropped, which includes,

- 32 *Meaningless Features* - such as ID, name, constant value, text-related values,
- *16 Redundancy features* - we will keep one feature if a multiple features of the same information exist,
- *3 Over-missingness Features* - features with over 50% missing values.

2.2 Re-code the Features

42 features are recoded, which are,

- *Convert the Data Type*- 24 data objects are converted to Numeric/Categorical/DateTime/Boolean data type,
- *Re-categorize the Data – 17 features are re*grouped into less categories,
- *Extract Information from Useable Format* - extract 1 feature from list format and recoded as 30 dummy features.

2.3 Deal with Missingness

Dropping cases and imputation are applied,

- 961 (12.7%) records are dropped due to MNAR (Missing Not At Random),
- 2 features are imputed with 0 (13.63% NAs, 5.35% NAs),
- 2 features are imputed by median (9.90% NAs, 0.06% NAs).

2.4 Outliers Detection

Sanity check for all the data range and 0 outliers are dropped.

- Data Range and quantile are calculated for continuous data,
- The Percentage for each class are calculated for categorical data.

2.5 Generate New Features

3 new features are created by the raw features, which are,

- *time_to_first_review*: months a listing takes from join to get the first review,
- *last_review_month*: months a listing get the last review to the time that data was scraped,
- *days_on_airbnb*: days from the first review to the last review.

2.6 Define the Target Metric – Yearly Revenue

This section is a step-by-step demonstration how we define the metric **Yearly Revenue** as the target features.

The San Francisco Model refers to a method created by Alex Marqusee for the San Francisco Planning Department (Brousseau, 2015) and the Budget and Legislative Analyst's Office (Rodgers, 2015) to quantify the impact of Airbnb in this city. These method given develop ways of estimate business metrics to evaluate local Airbnb listing's booking, occupancy rate and revenue.

We use the San Francisco Model as a guide for our calculation, below we have listed the features and formulate that leads us to the final business metrics,

- **Days on Airbnb** = $last_{review}$ - $first_{review}$
- **Minimum Booking in Year** = *reviews_per_month*12*
- **Estimated Booking in Year** = *Minimum Booking in Year/ 50%*
- **Nights Per Year CAP** = *Estimated Booking in Year * minimum_nights_avg_nt*
- **Occupancy Rate** = *Nights Per Year CAP /365*
- **Yearly Revenue** = *price * Occupancy Rate * 365*

2.7 Data Codebook

The final dataset contains 6,583 observations and 80 features in total. A simplified data codebook is given below,

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6583 entries, 0 to 7513
Data columns (total 80 columns):
 #   Column                                          Non-Null Count  Dtype
---  ------                                          --------------  -----
 0   host_id                                         6583 non-null   int64
 1   host_since                                      6583 non-null   datetime64[ns]
 2   host_response_time                              6583 non-null   object
 3   host_response_rate                              6583 non-null   category
 4   host_acceptance_rate                            6583 non-null   float64
 5   host_is_superhost                               6583 non-null   float64
 6   host_identity_verified                          6583 non-null   float64
 7   neighbourhood_group_cleansed                    6583 non-null   category
 8   zipcode                                         6554 non-null   object
 9   latitude                                        6583 non-null   float64
 10  longitude                                       6583 non-null   float64
 11  is_location_exact                               6583 non-null   float64
 12  property_type                                   6583 non-null   category
 13  room_type                                       6583 non-null   category
 14  accommodates                                    6583 non-null   int64
 15  bathrooms                                       6583 non-null   float64
 16  bedrooms                                        6583 non-null   float64
 17  beds                                            6583 non-null   float64
 18  price                                           6583 non-null   float64
 19  security_deposit                                6583 non-null   float64
 20  cleaning_fee                                    6583 non-null   float64
 21  guests_included                                 6583 non-null   int64
 22  extra_people                                    6583 non-null   float64
 23  minimum_nights_avg_ntm                          6583 non-null   float64
 24  calendar_updated                                6583 non-null   category
 25  availability_90                                 6583 non-null   int64
 26  number_of_reviews                               6583 non-null   int64
 27  number_of_reviews_ltm                           6583 non-null   int64
 28  first_review                                    6583 non-null   datetime64[ns]
 29  last_review                                     6583 non-null   datetime64[ns]
 30  review_scores_rating                            6583 non-null   category
 31  review_scores_accuracy                          6583 non-null   category
 32  review_scores_cleanliness                       6583 non-null   category
 33  review_scores_checkin                           6583 non-null   category
 34  review_scores_communication                     6583 non-null   category
 35  review_scores_location                          6583 non-null   category
 36  review_scores_value                             6583 non-null   category
 37  instant_bookable                                6583 non-null   float64
 38  cancellation_policy                             6583 non-null   object
 39  require_guest_profile_picture                   6583 non-null   float64
 40  require_guest_phone_verification                6583 non-null   float64
 41  calculated_host_listings_count_entire_homes     6583 non-null   int64
 42  calculated_host_listings_count_private_rooms    6583 non-null   int64
 43  calculated_host_listings_count_shared_rooms     6583 non-null   int64
 44  reviews_per_month                               6583 non-null   float64
 45  time_to_first_review                            6583 non-null   category
 46  last_review_month                               6583 non-null   category
 47  Wifi                                            6583 non-null   float64
 48  Essentials                                      6583 non-null   float64
 49  Heating                                         6583 non-null   float64
 50  Smoke detector                                  6583 non-null   float64
 51  Shampoo                                         6583 non-null   float64
 52  Hangers                                         6583 non-null   float64
 53  Hair dryer                                      6583 non-null   float64
 54  Carbon monoxide detector                        6583 non-null   float64
```

```
 55  Laptop friendly workspace                    6583 non-null   float64
 56  Iron                                         6583 non-null   float64
 57  TV                                           6583 non-null   float64
 58  Washer                                       6583 non-null   float64
 59  Dryer                                        6583 non-null   float64
 60  Hot water                                    6583 non-null   float64
 61  Fire extinguisher                            6583 non-null   float64
 62  Dishes and silverware                        6583 non-null   float64
 63  Refrigerator                                 6583 non-null   float64
 64  Microwave                                    6583 non-null   float64
 65  Coffee maker                                 6583 non-null   float64
 66  Self check-in                                6583 non-null   float64
 67  Free street parking                          6583 non-null   float64
 68  Cooking basics                               6583 non-null   float64
 69  Stove                                        6583 non-null   float64
 70  Bed linens                                   6583 non-null   float64
 71  First aid kit                                6583 non-null   float64
 72  Oven                                         6583 non-null   float64
 73  Private entrance                             6583 non-null   float64
 74  Extra pillows and blankets                   6583 non-null   float64
 75  Free parking on premises                     6583 non-null   float64
 76  Dishwasher                                   6583 non-null   float64
 77  days_on_airbnb                               6583 non-null   float64
 78  occupancy_rate                               6583 non-null   float64
 79  yearly_revenue                               6583 non-null   float64
dtypes: category(14), datetime64[ns](3), float64(51), int64(9), object(3)
memory usage: 3.5+ MB
```

Check details and source codes for Data Wrangling are available at

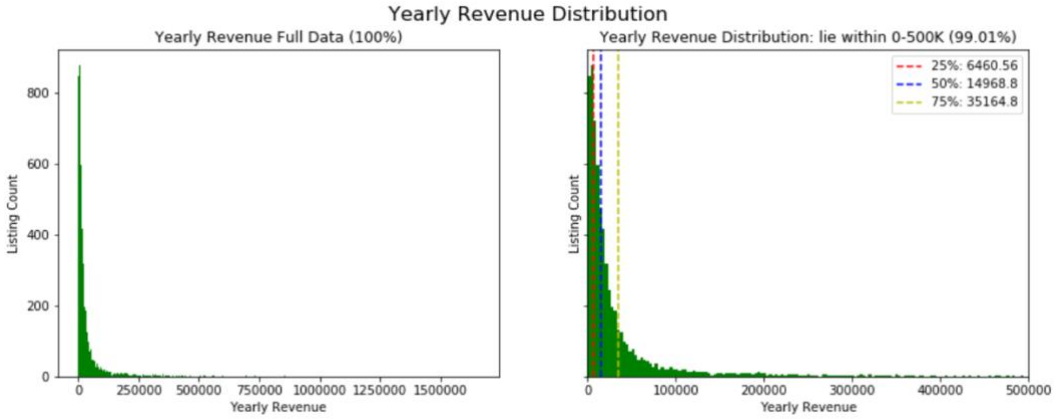
Capstone_Project_Data_Wrangling.

3. Exploratory Data Analysis Summary

In this section, we perform the Exploratory Data Analysis on the cleaned data generated from the Data Wrangling step. We will go through most of the features in the data to explore their relationship with the Yearly Revenue. And the EDA is organized based on different topics on the related information, features regarding the same topic information will be analyzed together.

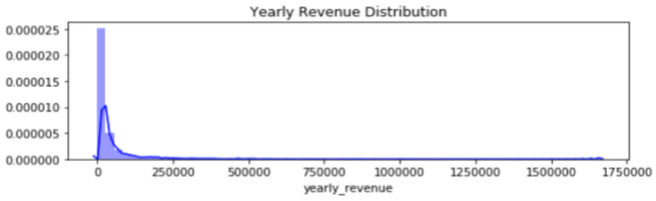
3.1 Yearly Revenue

The Yearly Revenue is the target business metric we want to evaluate for all the listing population from our data, the definition was given in 2.7. Let's get the distribution and statistics summary for it.



Yearly Revenue Statistics

Mean	$42,788.59
Std	$92,133.01
Min	$96.00
25%	$6460.56
50%	$14,968.80
75%	$35,164.80
Max	$1,659,240.00



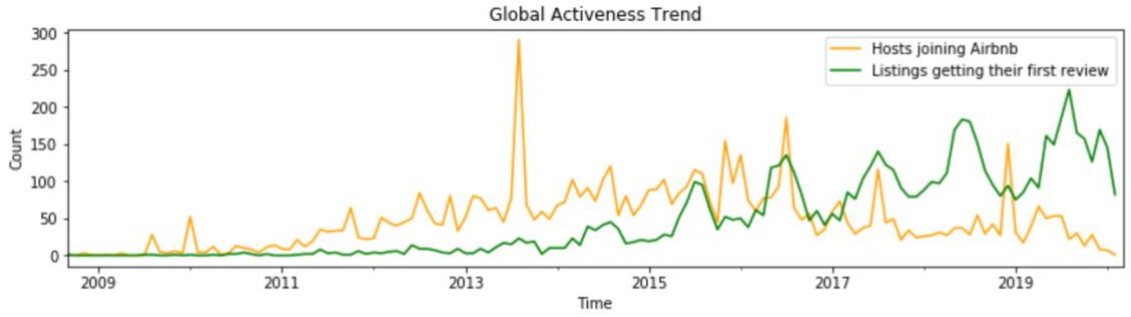
From the summary statistics above, we can see that a listing can have a very small revenue at all among the year whereas a listing can gain a very high revenue up to $1,659,240. However, the distribution of the yearly revenue is highly right-skewed, we can see that most of the data are centered within $100,000, and 99% of our data lies within $500,000. The median is about

$ 14,968 and the average revenue goes up to $42,788. The density plot is given alongside. Putting those numbers into perspective, the 2019 Airbnb official sites reported that On average globally, Experience hosts earn $10,000 a year, our data indicating a healthy premium for actively managed listings for Seattle Airbnb Market.

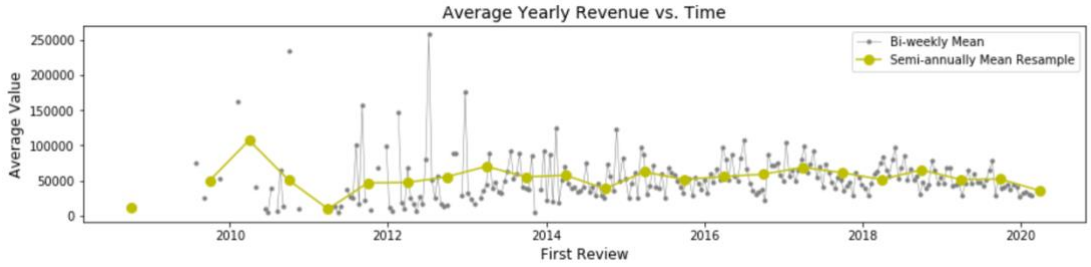
3.2 Host Activeness on the Platform

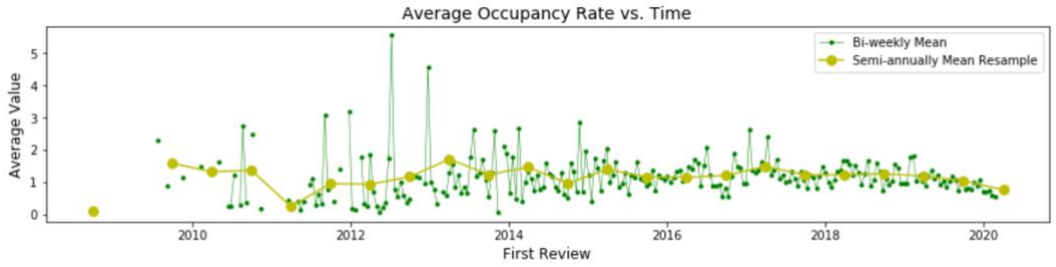
In this section, we will exam the host activeness from different perspectives, which includes the **global activeness on the platform**, **the ability to attract the guest and stay on the market**, and it's **longevity on the platform**. And also explore their relationship with the yearly revenue.

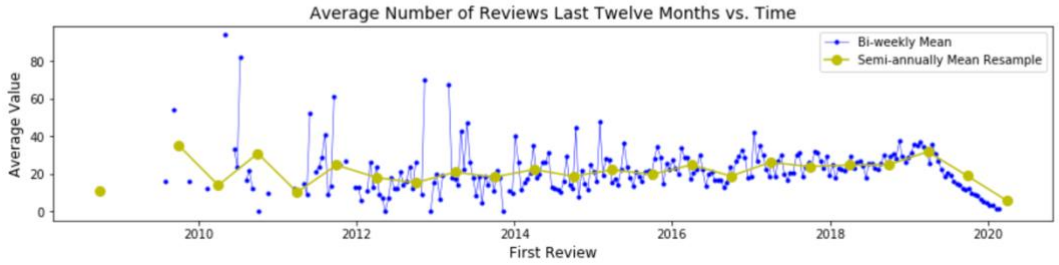
Global Business Trend

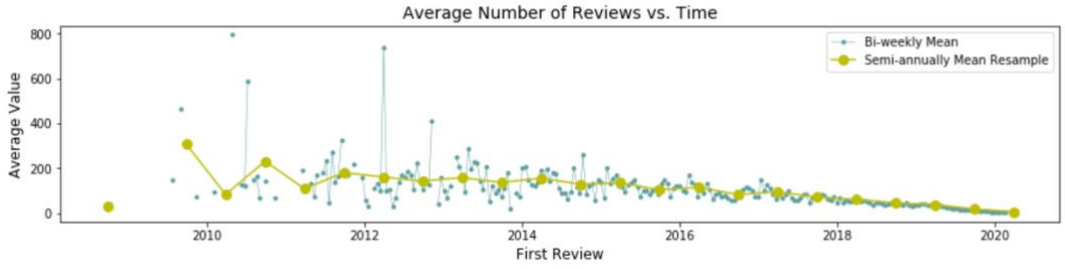


The global activeness trends are using the time point that the listing joins Airbnb and the time a listing gets their first review as the X-axis, and the y-axis indicates the count of the listing counts. The plots indicate that the business keeps boosting all these years – more host join Airbnb and the number of a host gets its first review is increasing over the decades.





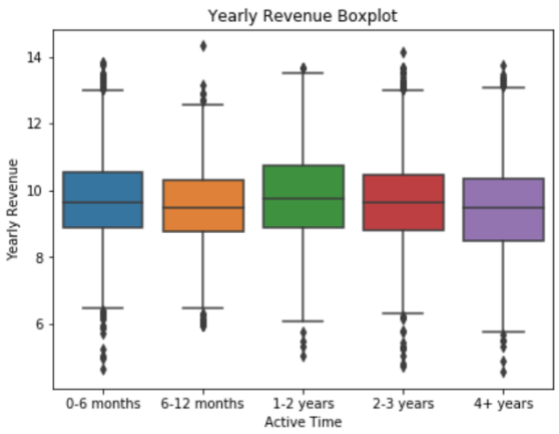


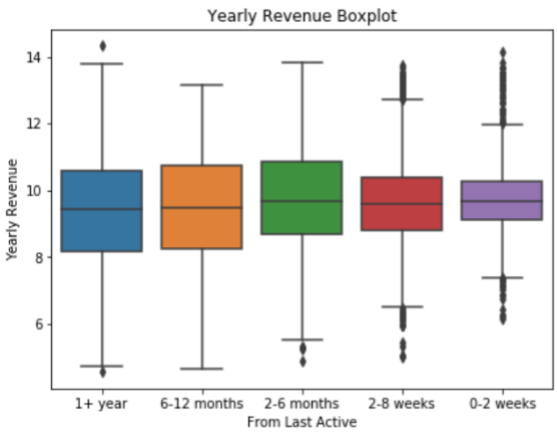


In general, during the decades of 2010 to 2020, there is not a tremendous increase among all the business metrics. The yearly revenue, price, and occupancy rate look flat with a slightly increasing tendency. Noticed that both the average review number and that feature in the last

twelve months have a slight decrease trend could cause by a delay of getting the review data coming in since it takes time for a guest to leave their review.

Power of Attraction & Staying



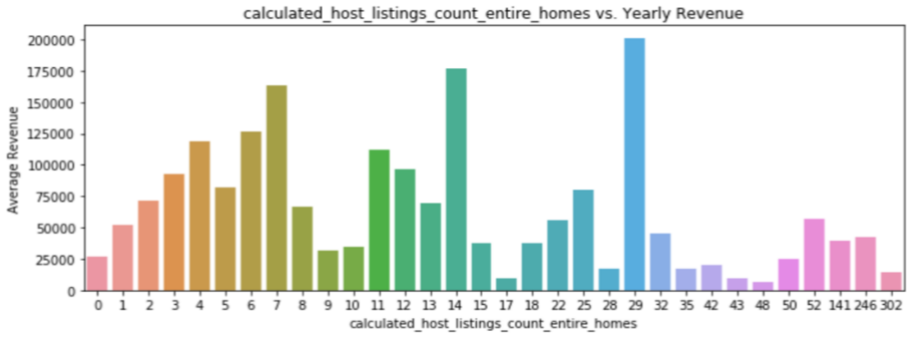


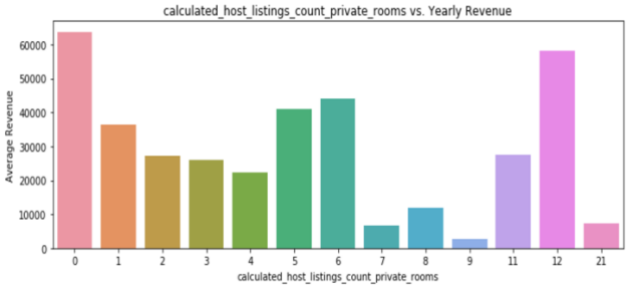
Statistical summary table

active_time				from_last_active			
	Count	Mean	Median		Count	Mean	Median
0-6 months	1,603	$47,808	$15,585	**0-2 weeks**	1,655	$33,775	$16,236
6-12 months	630	$34,332	$12,994	**2-8 weeks**	1,987	$43,877	$14,515
1-2 years	981	$51,735	$17,280	**2-6 months**	1,463	$54,541	$16,245
2-3 years	1,743	$43,588	$15,552	**6-12 months**	480	$41,594	$13,212
4+ years	1,626	$34,860	$13,058	**1+ years**	998	$38,912	$12,391

The created feature active_time is the months from joining to get the first review, which indicated the power of attraction - If a listing can get a review faster once it joined the platform, the more powerful of attraction it is. It looks like the among the groups of different months length to get the first review, the yearly revenue doesn't change much. The statistics summary indicates that the 1-2 years have the highest mean ($51,735) and median ($17,280). To embody the staying power, we created feature from_last_active, which is the length of the last review to the data was scrapped time, which we believe is an indication of the survival ability - The listing would have better survival power if it can staying active as late as it can. The plots indicate that the groups within 6 months(2-6 months, 2-8 weeks and 0-2 weeks) have a relatively high yearly revenue.

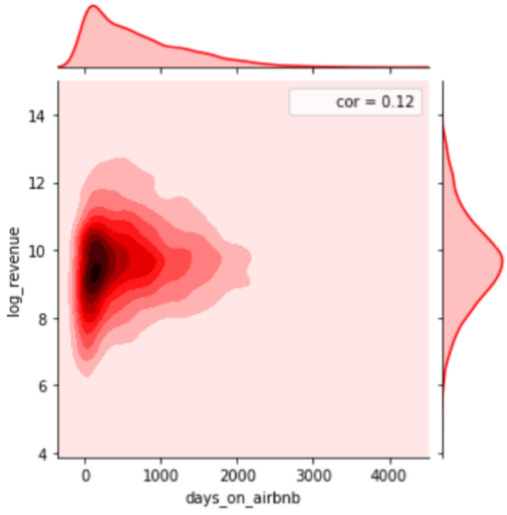
Host Listing Counts





The Host Listing Count features are specified for entire homes/private rooms/shared rooms, the average yearly revenue varies and shows no specific patterns among different counts.

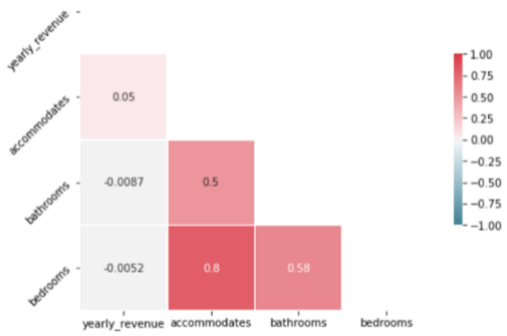
Days on Airbnb



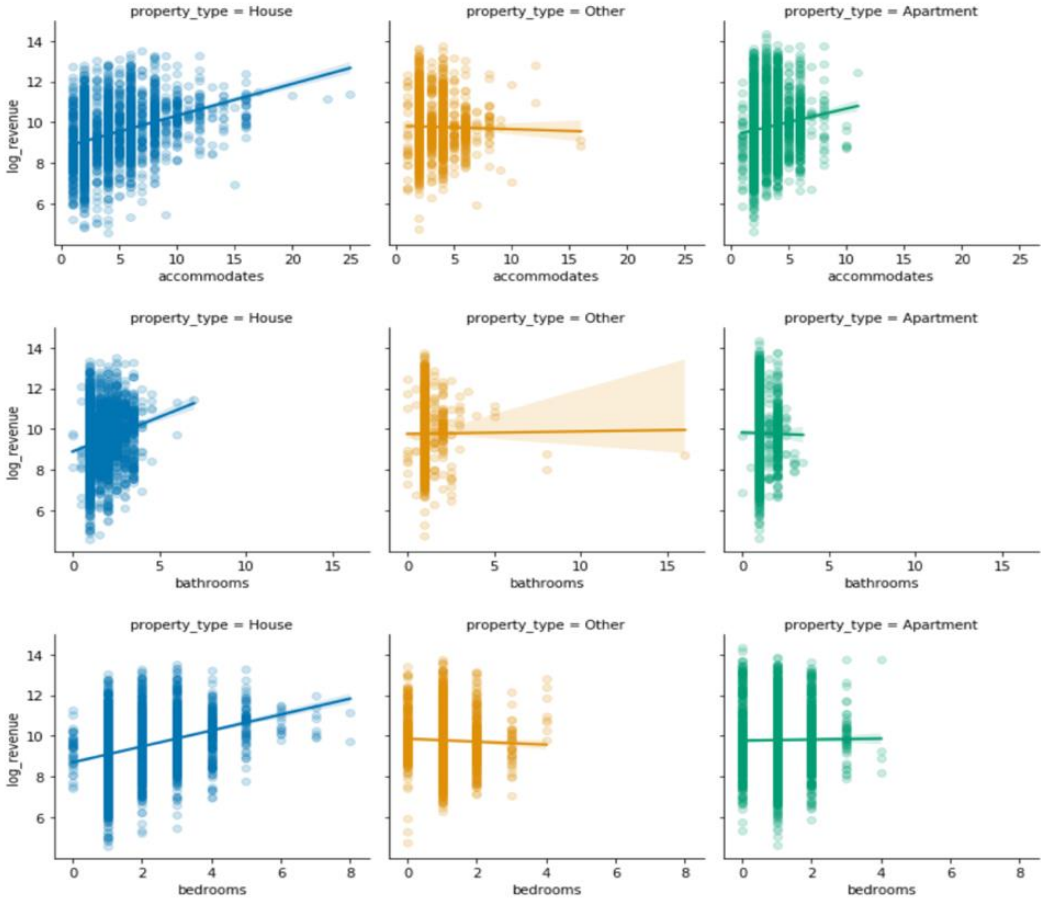
The features days_on_airbnb is calculated length from the first review to the last review, which potentially indicates the activeness length on the platform. Original Pearson correlation is weak 0.035 between the yearly revenue and days on Airbnb. The KDE plot above shows the log-transformed revenue and correlation. We can see that most intense data are within 0-1000 days, which is about 3 years, there's a weak indication that longer days are associated with higher revenue from the plot.

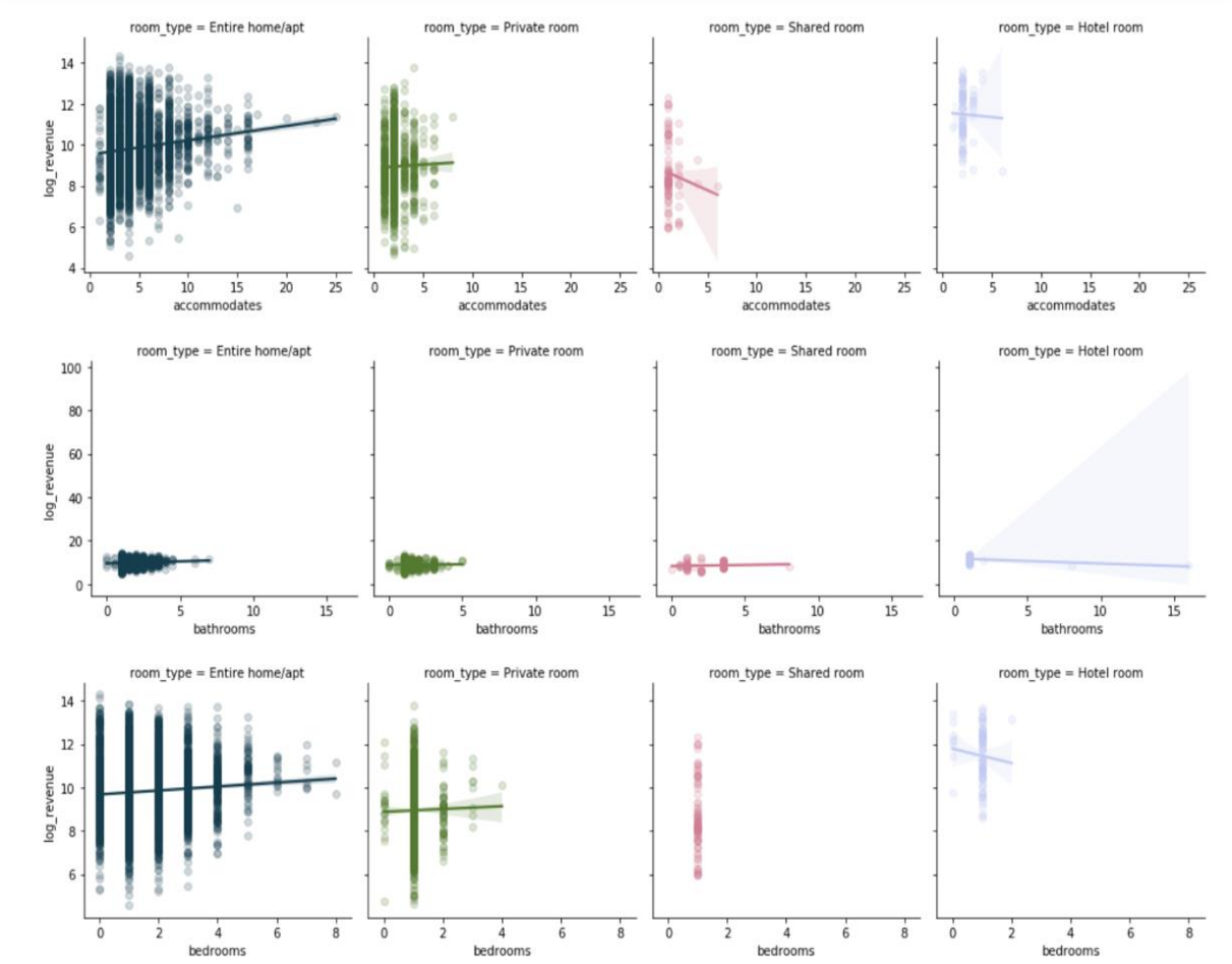
3.3 Listings Accommodates & Facilities

Bedrooms & Bathrooms & Accommodates



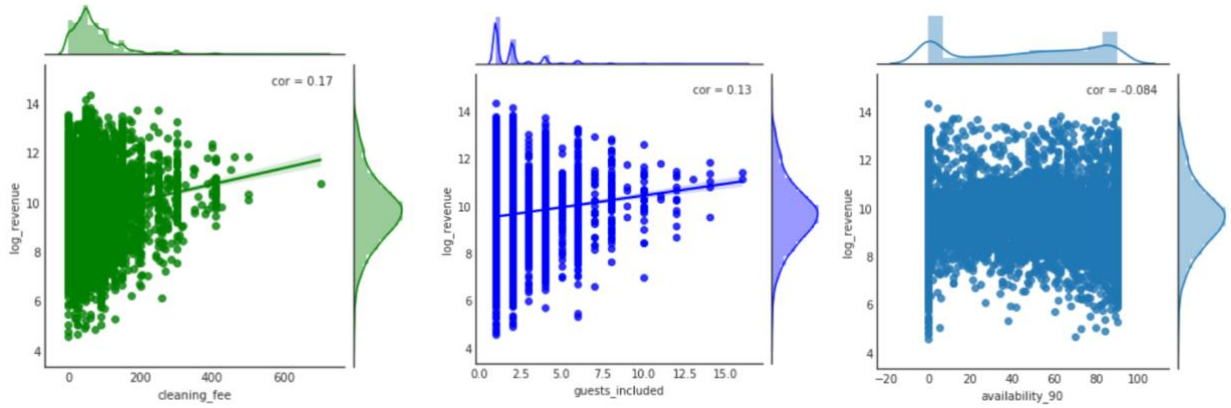
The correlation matrix above shows a negligible correlation between the revenue and bedrooms/bathrooms, let's take a further examination this accommodates/bedrooms/bathrooms features within two features - property type and room type.





The above-detailed plots the relationship between yearly revenue with each class in property type and room type. Only under the property type is house and apartment, there's positive relationship indicate that as the number of accommodates increases, the yearly revenue increases. And the same for revenue and accommodates within property type is an apartment. However, the relationship varied among all these break-down features. Besides, for hotel room and shared room, it's hard to determine the correlation trend - since one or two extreme points bias the regression line.

Clean Fee & Guest Included & Availability



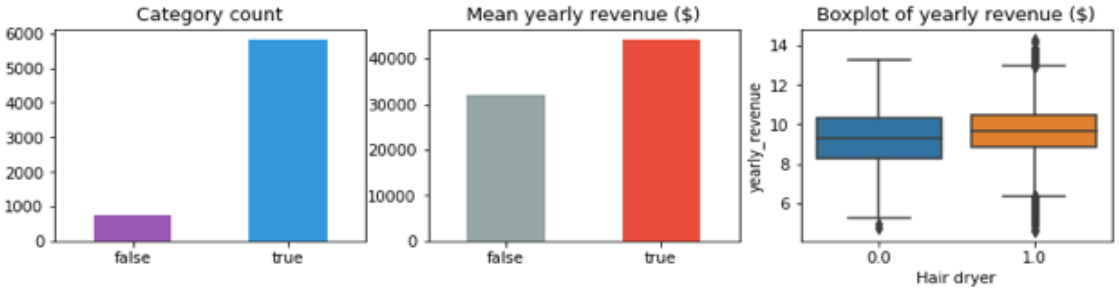
The above features are examined with log-transformed yearly revenue. There is positive relationship between the yearly revenue and features of cleaning fee and guests included.

Amenities

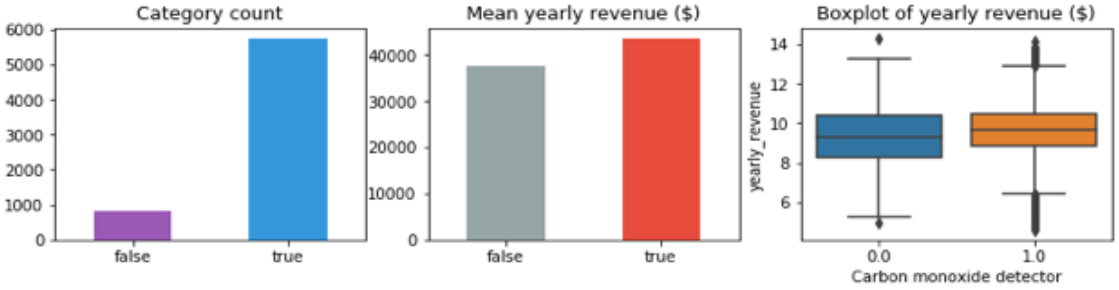
We have 30 single amenity features to check - they are essential facilities like Wifi, TV, Hairdryer, etc. We combine the plots of the counts with and without the single amenities, the mean revenue for each class along with the boxplot, we display the 5 out of 10 single amenities here, check the Jupyter notebook for detailed plot analysis.

Hangers

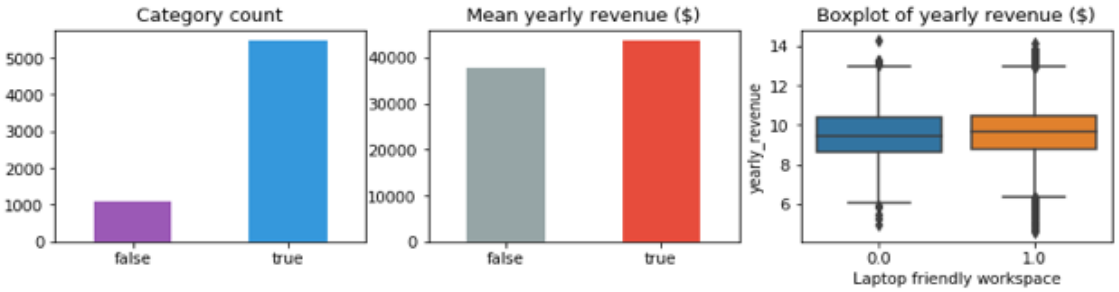
Hair dryer



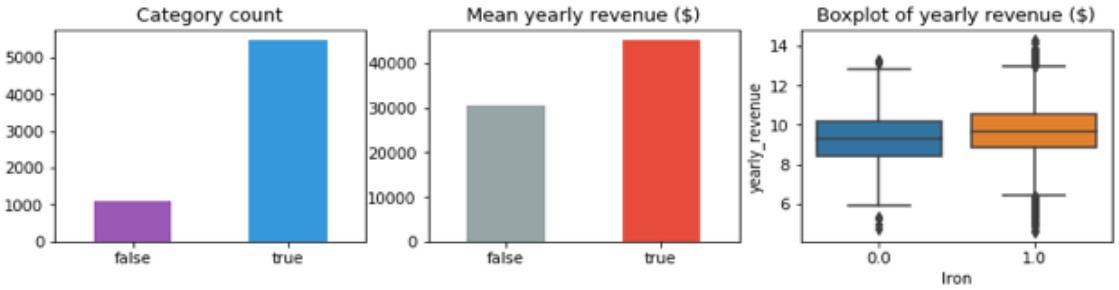
Carbon monoxide detector



Laptop friendly workspace



Iron



For better information, we implement bootstrap hypothesis testing across the 30 amenities. The univariate test reveals statistical significance for five amenities - **Fire extinguisher, Free street parking, First aid kit, private entrance and Free parking on premises,** which indicates with those amenities provided, the yearly revenue is statistically different from the listings don't. Further examination brings in more surprising results - the mean yearly revenue is lower with those amenities.

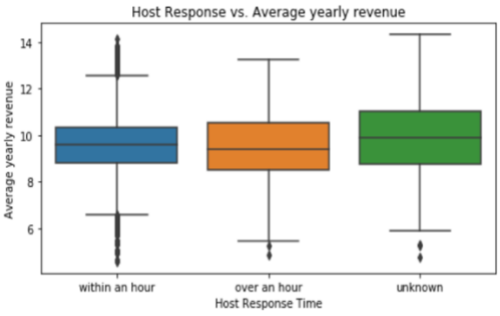
```
Bootstrap hypothesis test for difference of means yearly revenue:

Hypothesis Test of Wifi                          | p-value = 0.4641
Hypothesis Test of Essentials                    | p-value = 0.9673
Hypothesis Test of Heating                       | p-value = 0.4547
Hypothesis Test of Smoke detector                | p-value = 0.9892
Hypothesis Test of Shampoo                       | p-value = 0.9614
Hypothesis Test of Hangers                       | p-value = 1.0
Hypothesis Test of Hair dryer                    | p-value = 1.0
Hypothesis Test of Carbon monoxide detector      | p-value = 0.9708
Hypothesis Test of Laptop friendly workspace     | p-value = 0.9898
Hypothesis Test of Iron                          | p-value = 1.0
Hypothesis Test of TV                            | p-value = 1.0
Hypothesis Test of Washer                        | p-value = 1.0
Hypothesis Test of Dryer                         | p-value = 1.0
Hypothesis Test of Hot water                     | p-value = 0.7468
Hypothesis Test of Fire extinguisher             | p-value = 0.0
Hypothesis Test of Dishes and silverware         | p-value = 0.2256
Hypothesis Test of Refrigerator                  | p-value = 0.0344
Hypothesis Test of Microwave                     | p-value = 0.0519
Hypothesis Test of Coffee maker                  | p-value = 0.4128
Hypothesis Test of Self check-in                 | p-value = 1.0
Hypothesis Test of Free street parking           | p-value = 0.0
Hypothesis Test of Cooking basics                | p-value = 0.9986
Hypothesis Test of Stove                         | p-value = 0.9977
Hypothesis Test of Bed linens                    | p-value = 0.9245
Hypothesis Test of First aid kit                 | p-value = 0.0
Hypothesis Test of Oven                          | p-value = 0.9998
Hypothesis Test of Private entrance              | p-value = 0.0
Hypothesis Test of Extra pillows and blankets    | p-value = 0.798
Hypothesis Test of Free parking on premises      | p-value = 0.0
Hypothesis Test of Dishwasher                    | p-value = 0.9967
```

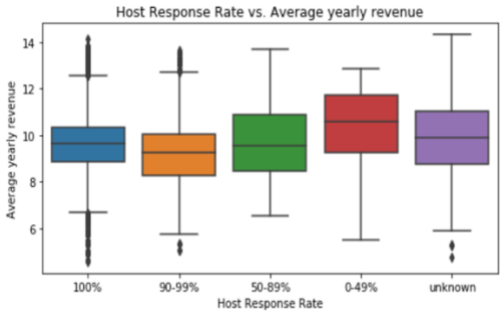
3.4 Interactive Relationship

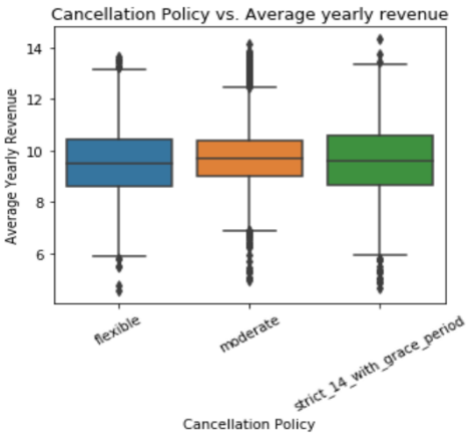
The section we group the features into two perspectives and the exploration analysis will be performed from both host and guest sides whether the active interaction between from either side will relate to different earning.

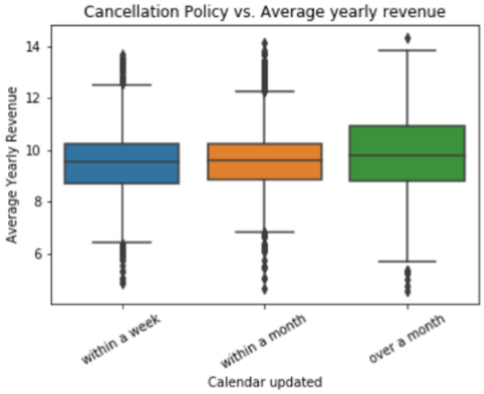
From Host's Side

The more active response rate and time do not appear to have a better revenue than we expected before. Neither are the features as cancellation policy and calendar updated time.







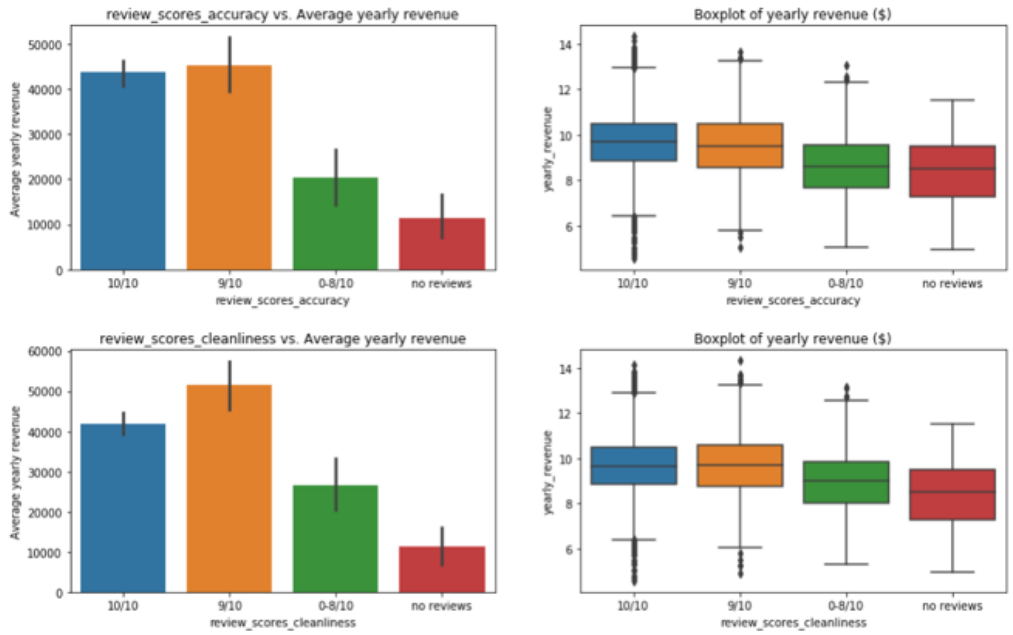


We explore the five features – *host_is_superhost, is_location_excat, instant-bookable, require_guest_profile_picture* and *require_guest_phone_verification in this below section*. The visualization and bootstrap hypothesis testing are applied to these features. It looks like only *require_guest_profile_picture* and *require_guest_phone_verification* are associated with lower yearly revenue – maybe the guests prefer no verification and skip these listing when making a choice, which also validated by the bootstrap hypothesis testing below. Check the Jupyter notebook for detailed visualization.

```
Bootstrap hypothesis test for difference of means yearly revenue:

Hypothesis Test of host_is_superhost                | p-value = 1.0
Hypothesis Test of is_location_exact                | p-value = 0.4659
Hypothesis Test of instant_bookable                 | p-value = 1.0
Hypothesis Test of require_guest_profile_picture    | p-value = 0.0
Hypothesis Test of require_guest_phone_verification | p-value = 0.0
```

From Guest's Side

The features we can get info from the guest's side are their rating scores. There are 7 review scores regarding accuracy, cleanliness, checking, communication, location and value. As we expected – the higher score groups earn more revenue in general, and the yearly revenue trend almost monotonic decreasing along with the review score go down. Noted that 2 features are given here, check the Jupyter notebook for more details.

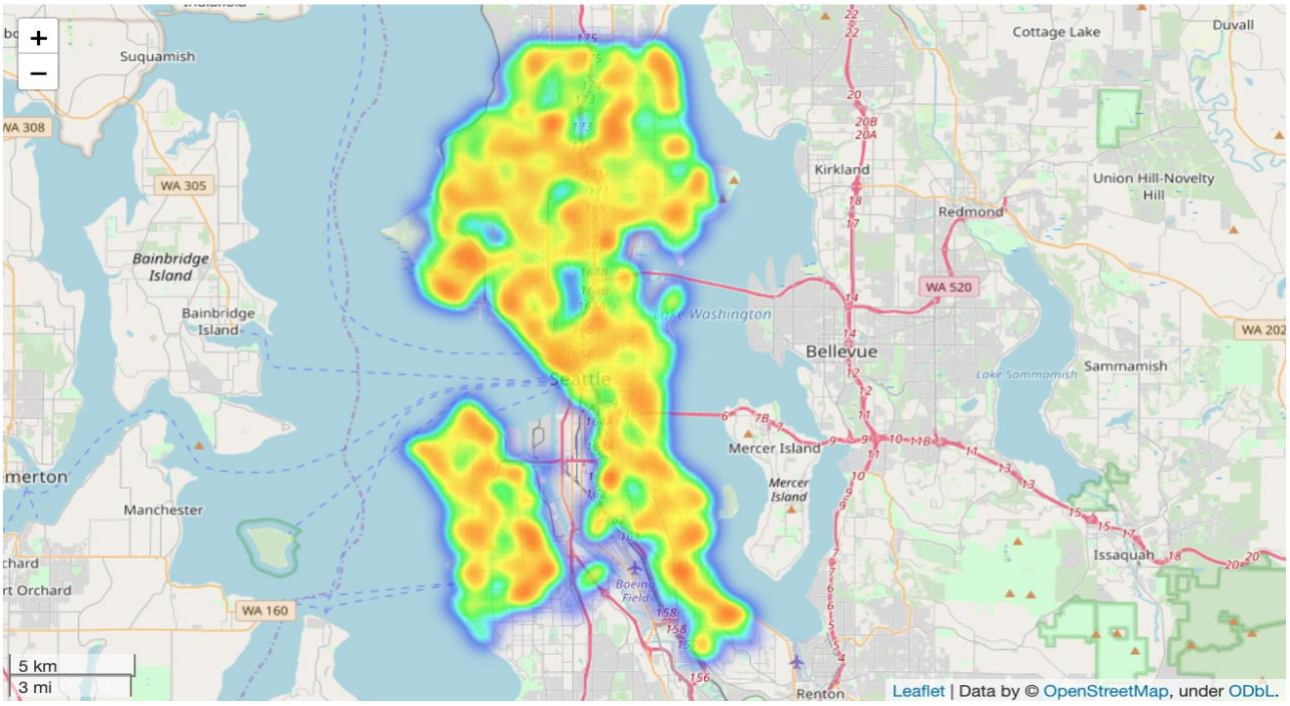


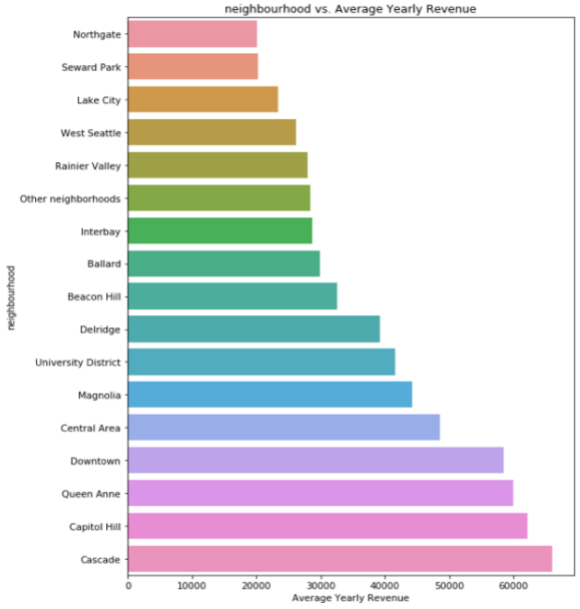
3.5 Geographical Information

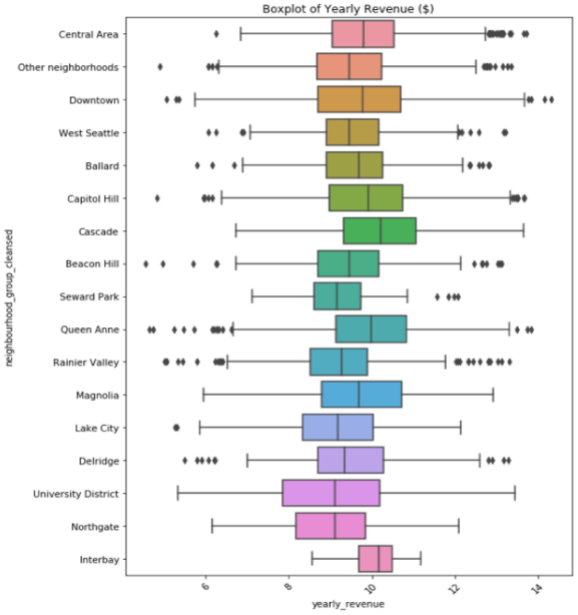
We mapped the median revenue that group by the geographical information (latitude, longitude) and use the heatmap to visualize the revenue distribution on the map.

In general, we spot that obvious that the central part has better revenue than the marginal region in the whole Seattle region, When zooming in the plot, it's more clear some region such

as Queen Anne, and Capitol Hill.







We group the data by the neighborhood classes and use bar plot to display the average revenue for each area, the trends align with we get from the geographical data, the top regions are Cascade, Capitol Hill, Queen Anne, Downtown, etc.

4. Results and In-depth Analysis Using Machine Learning

This section consisted of two parts, we will start from the **Data Preprocessing**, in which we will access the collinearity to drop the highly correlated features, and also apply the normalizing and standardizing on the dataset. After that, we will apply the machine learning algorithms in the **Modeling** part with the processed data.

4.1 Preprocess Data

4.1.1 Access Collinearity

We use the heatmap as a tool to access the correlation between all the features we have.

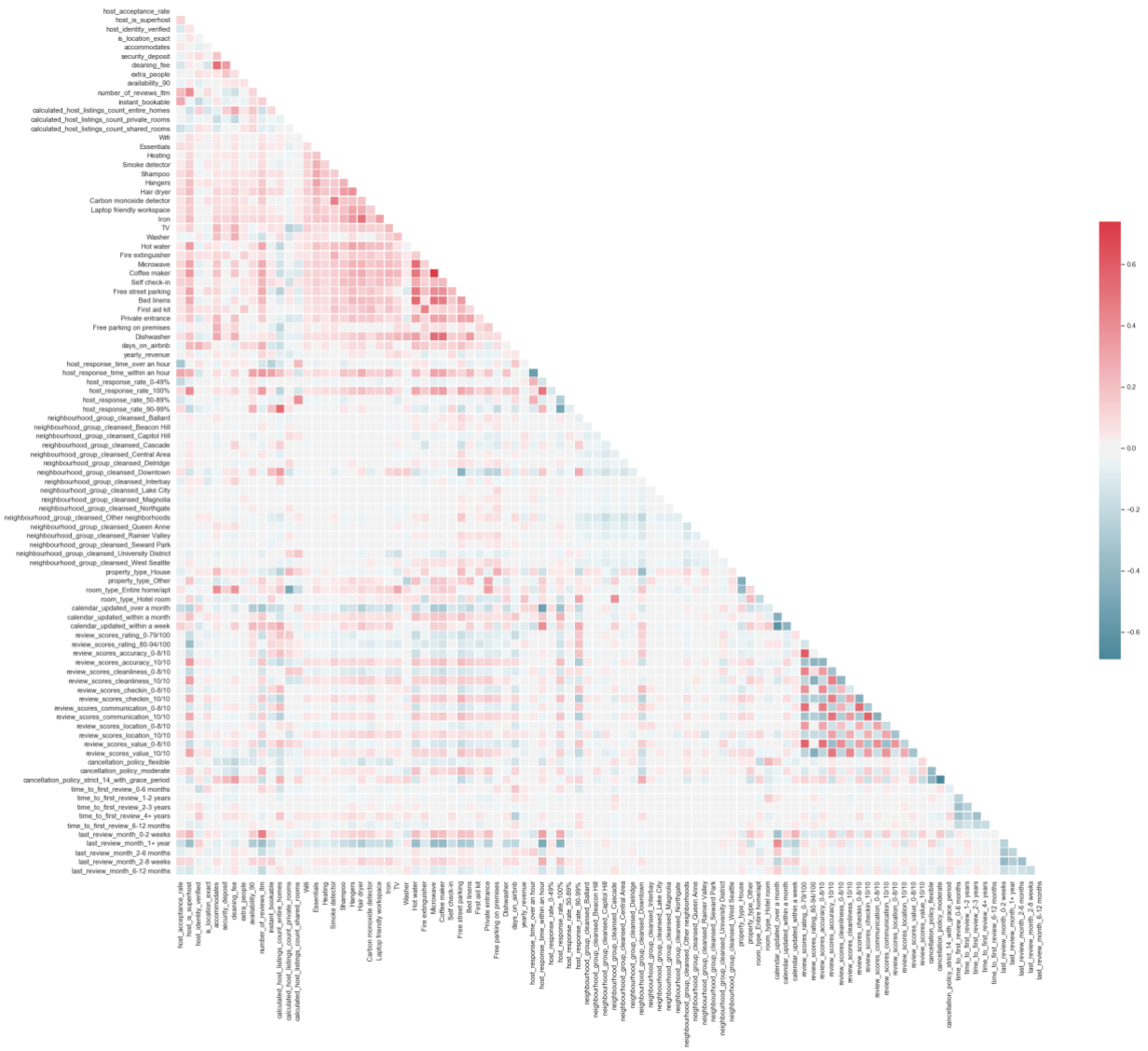
Heatmap on features before dropping

The heatmap show's before processing, the correlation range is (-0.75, 0.75). Some features are highly correlated,

- *Accommodates* and *bathrooms, bedrooms, bed, guest_included,*
- *require_guest_phone_verification* and *require_guest_profile_profile_picture,*
- *Dryer* and *Washer, Dishes and silverware* and *Refrigerator, etc.,*
- All the *'_No Reviews' features,* all the *'review_score_9/10'*

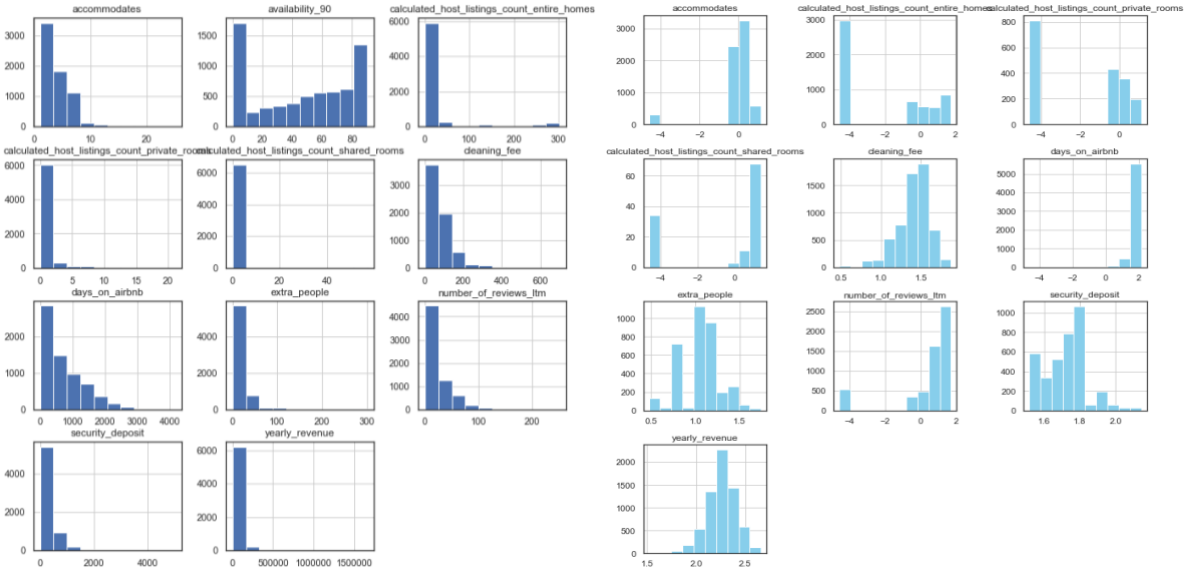
For all of the correlated features, we will only keep one from them. The heatmap for the reduced data is given below, the correlation range goes down to (-0.6, 0.6).

Heatmap on features after dropping



4.1.2 Normalizing & Standardizing

Before vs. after normalization



Then for all the continuous features, we applied the log transform. We still see that not every single feature approached the normal distribution perfectly, but all of them improved a lot. Finally, we split the data into the train (80%) and the test (20%) dataset, and the standard scaler is applied to the train data, the train scaler the applied to test data to prevent the leaking problem.

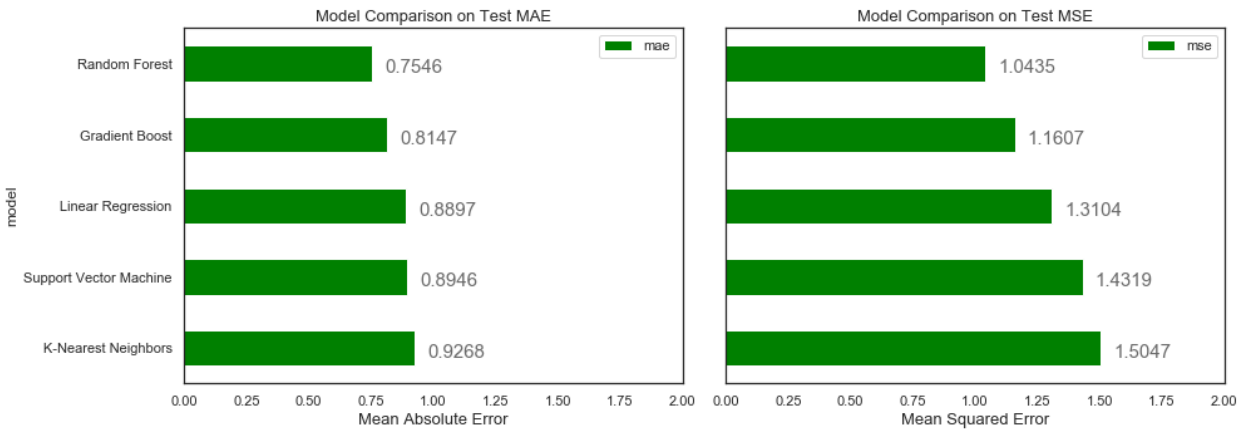
4.2 Evaluate & Compare Machine Learning Models

4.2.1 Models to Evaluate

We will compare five different machine learning models using the Scikit-Learn library:

- Linear Regression
- Support Vector Machine Regression
- Random Forest Regression
- Gradient Boosting Regression
- K-Nearest Neighbors Regression

To compare the models, we are going to be mostly using the Scikit-Learn defaults for the model hyperparameters. And in this section, we will focus on these models' baseline performance on our dataset generally instead of optimizing the model with a determined model to use. Then we can select the best performing model for further optimization using hyperparameter tuning.



We use the testing **Mean Absolute Error (MAE)** and **Mean Standard Error (MSE)**. The Base model comparison shows Random Forest, Gradient Boost and Linear Regression have the best performance.

4.2.2 Model Optimization

4.2.2.1 Hyperparameter Tuning

In machine learning prediction, optimizing a model will find us the best set of hyperparameters for to achieve the best performance. we will start with the best base model random forest first. In our case of a random forest, hyperparameters include,

- **n_estimators** - number of trees in the forest
- **max_features** - max number of features considered for splitting a node
- **max_depth** - max number of levels in each decision tree
- **min_samples_split** - min number of data points placed in a node before the node is split
- **min_samples_leaf** - min number of data points allowed in a leaf node
- **bootstrap** - method for sampling data points (with or without replacement)

The best hyperparameters are usually impossible to determine before training, which means the best practice to determine the optimal settings is to try many different combinations to evaluate the performance of each model. However, evaluating each model only on the training set can cause overfitting - which means our model will score very well on the training whereas it will not be able to generalize to new data as good as we expected.

Hyperparameter Tuning with Random Search within Cross Validation will generate more reliable results for our reference.

In our case, we will use Random Search with cross validation to explore where the model can have relatively good performance, once we narrow down the options and then use the grid search with a more limited range of options for the best model.

The K-Fold Cross Validation technique dividing the training data into K folds, and then going through an iterative process on each K-1 fold data set and evaluate performance on the kth fold. Average error on each of the K iterations as the final performance will be computed as the given results.

The next section will demonstrate how we use **random search** to narrow down the scope and use **grid search** to find our best model, with the combination of cross validation.

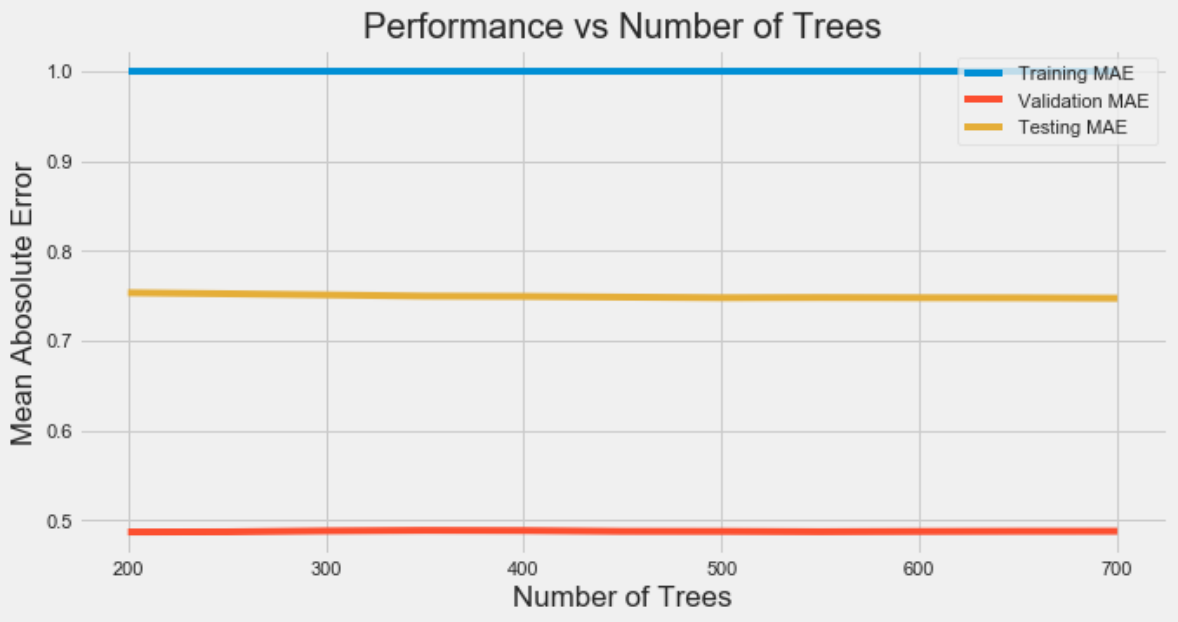
Random Search Hyperparameter Range

- **n_estimators**: [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]
- **max_features**: [auto, sqrt]
- **max_depth**: [10, 20, 30, 40, 50, 60, 70 , 80, 90, 100, 110, None]
- **min_samples_split**: [2, 5, 10]
- **min_samples_leaf**: [1, 2, 4]
- **bootstrap**: [True, False]

The **Random Search** with 3-fold cross-validation returns the best parameters, which are highlighted in red. To further elaborate the search range, we fix the other parameters from the best search above and set a narrower range for the number of trees above as [200, 250, 300,

350, 400, 450, 500, 550, 600, 650, 700]. As a result of searching, the **grid search** 3-fold cross-validation give us the best parameter as the number of trees as 350.

To gain more information on the grid search process, we plot the MAE for training, validation, and testing respectively, along with the iteration of the increased number of trees.



With the range of [200, 700] in general, the validation MAE is lower than the validation MAE, which means the model generalizes very good when applied to the validation dataset, the MAE is below 0.5. However, when we use the testing data to access the predict into the new data, the MAE bounce back a little around 0.75.

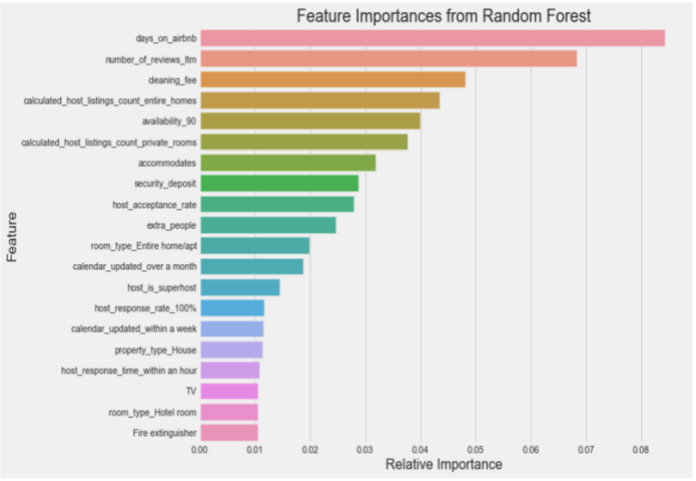
Though the prediction error has a slightly higher, as the model is getting more complicated (as we keep increasing the number of trees), the error looks stable, hence we think our model is not involved with overfitting issues.

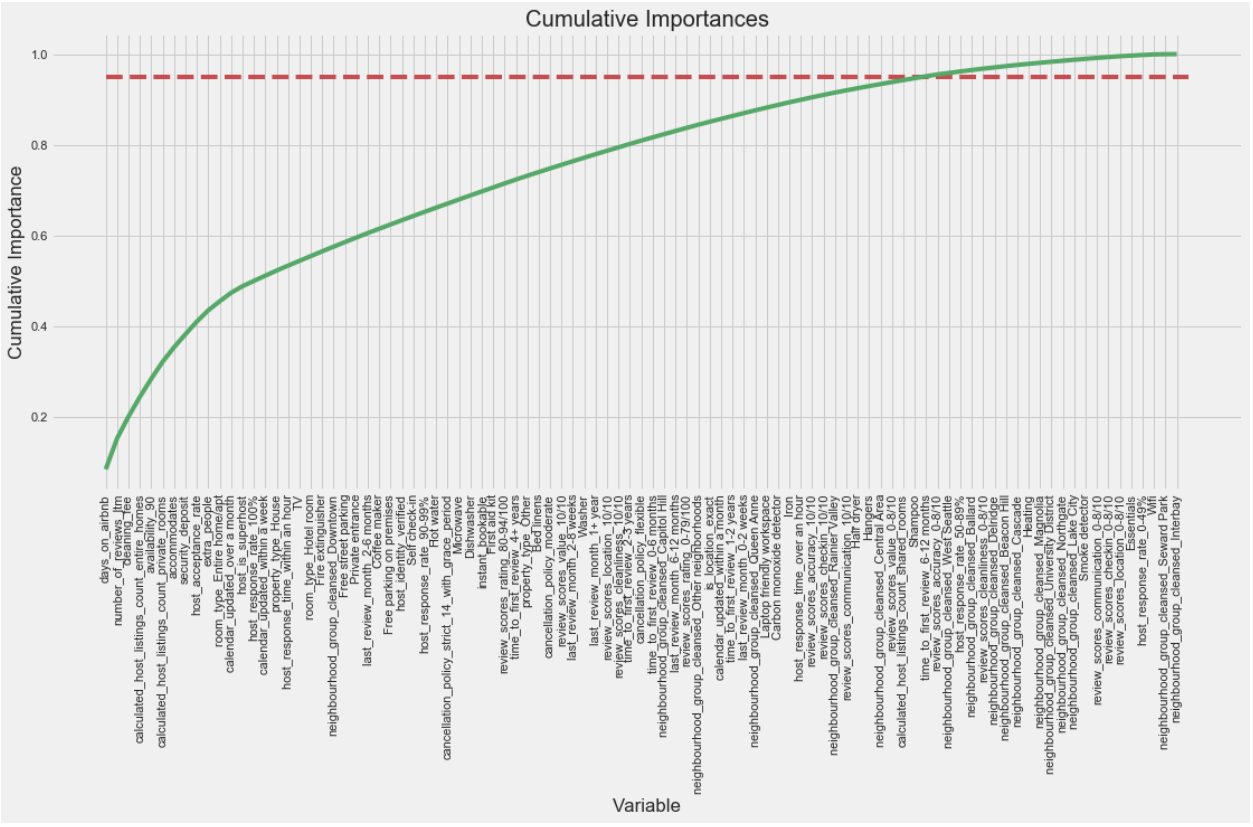
4.2.2.2 Feature Reduction

To better improve the model – we think of the curse of dimensionality – our model has 93 features! It's time to consider the feature reduction for the less complex model.

We plot the top 20 features in our model. The y-label are sorted values, the x- label indicates the relative importance of those features with respect to the predictability of the target variable. The *days_on_airbnb, numbers_of_reviews_lst, cleaning_fee* are the top 3 features for prediction.

	feature	importance
0	days_on_airbnb	0.084265
1	number_of_reviews_ltm	0.068338
2	cleaning_fee	0.048117
3	calculated_host_listings_count_entire_homes	0.043466
4	availability_90	0.039873
5	calculated_host_listings_count_private_rooms	0.037628
6	accommodates	0.031854
7	security_deposit	0.028797
8	host_acceptance_rate	0.027890
9	extra_people	0.024623
10	room_type_Entire home/apt	0.019932
11	calendar_updated_over a month	0.018718
12	host_is_superhost	0.014539
13	host_response_rate_100%	0.011703
14	calendar_updated_within a week	0.011527
15	property_type_House	0.011379
16	host_response_time_within an hour	0.010812
17	TV	0.010551
18	room_type_Hotel room	0.010536
19	Fire extinguisher	0.010521

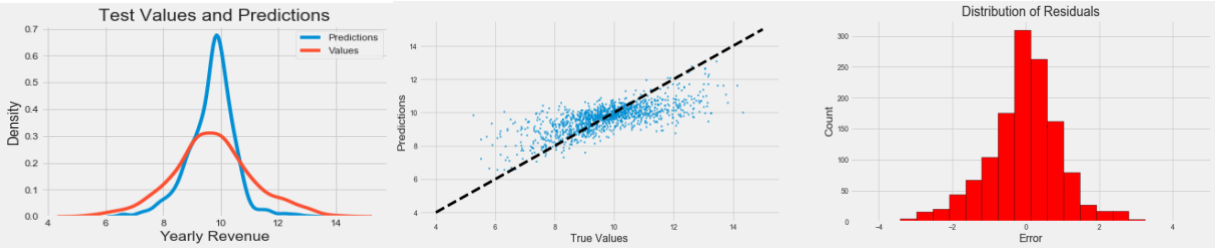




Further, we plot the accumulative importance by descending sorted features, and we added a 95% cutoff line, all the features within the 95% percentage contribution will be used in the new model with the best parameter as the re-fitting model.

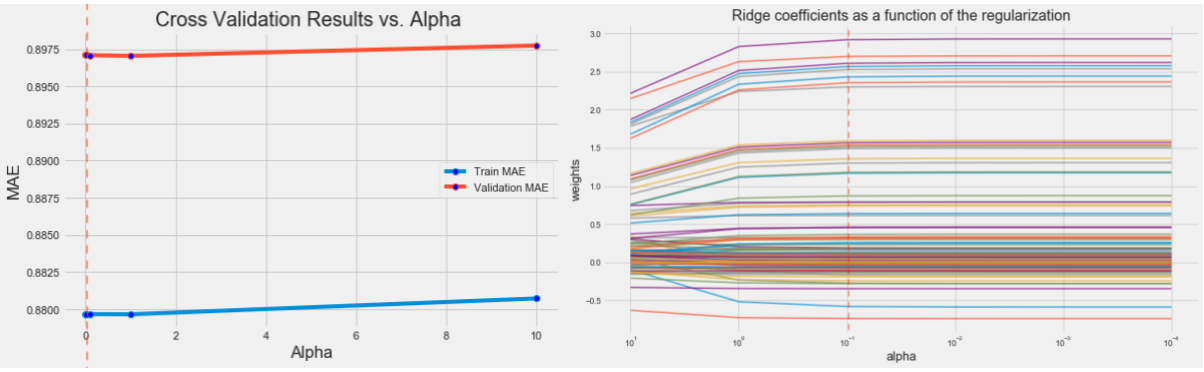
After refitting, the model contains 72 features, and the performance is as good as the model with 96 features.

We access the prediction of the best random forest model with reduced features, our model underpredict the yearly revenue towards the middle part of the data whereas overpredict the values on the tails for both sides. And compared to a normally distributed residual, our residual plot has a wider shape.



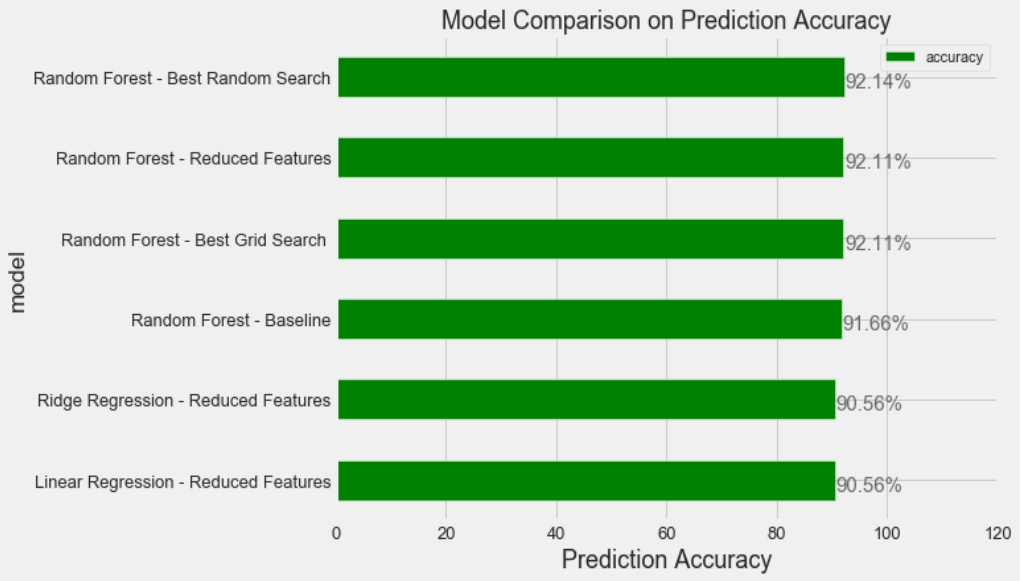
Since we already know that from the first beginning that the linear regression has a relatively good performance, we want to keep the model as simple as it can be, so we will use the reduced feature set to refit the linear regression and ridge regression.

```
Ridge Regression with Cross Validation:
alpha: 0.0001 | Avg Train MAE : 0.879700 | Avg Validation MAE: 0.897100
alpha:  0.001 | Avg Train MAE : 0.879700 | Avg Validation MAE: 0.897100
alpha:   0.01 | Avg Train MAE : 0.879700 | Avg Validation MAE: 0.897100
alpha:    0.1 | Avg Train MAE : 0.879700 | Avg Validation MAE: 0.897100
alpha:      1 | Avg Train MAE : 0.879700 | Avg Validation MAE: 0.897000
alpha:   10.0 | Avg Train MAE : 0.880700 | Avg Validation MAE: 0.897700
best alpha: 0.1
```



The cross-validation indicates when alpha = 0.1 we obtain the best ridge model. However, as a close examine the cross-validation results, we spot that the training MAE and the testing MAE don't vary much with the regulation term. Besides, the linear regression and ridge regression gain the same performance (**MAE as 0.891, MSE as 1.33 and Test Accuracy as 90.56%**), so the regularization is not necessary. We will move forward the linear regression in this section to access the prediction performance. The over and under prediction issue looks the same with random forest, however, the residual plot looks more symmetrical.

4.3 Conclusion & Recommendation



We visualize the accuracy for all the models above and summary the performance in the tables below,

Model	Feature Numbers	Test MSE	Test MAE	CV - MAE	Test Accuracy	Overfitting
Random Forest - Best Random Search	96	1.02	0.74	-	92.14%	-
Random Forest - Best Grid Search	96	1.04	0.75	0.46	92.08%	No
Random Forest - Reduced Features	72	1.03	0.75	-	92.10%	-
Linear Regression - Reduced Features	72	1.33	0.89	0.90	90.56%	No
Ridge Regression - Reduced Features	72	1.33	0.89	0.90	90.56%	No

Based on the results above, the **Random Forest with reduced features** would be our final model, which has,

- best Prediction accuracy – 92.10%,
- no overfitting issue,

Read the detailed machine learning analysis in the Jupyter notebook.

5. Conclusion

In this project, we first explored the Airbnb listing dataset with 93 intake features, to understand their influence on the yearly revenue, as a created business metric, which is a highly skewed numerical feature.

5.1 Exploratory Data Analysis Conclusion

We explore 5 types of information available in the dataset and by the univariate analysis, correlation analysis, and data visualization. Only part of the features influences the yearly revenue. Following is a quick summary of exploratory data analysis:

1. Yearly Revenue: The yearly revenue is created by price, occupancy rate. And the revenue statistics indicating a healthy premium for actively managed listings for Seattle Airbnb Market.

2. Host Activeness on the Platform: The global business trends show that the whole market place keeps boosting while hosts keep joining in the platform. However, The yearly revenue and other business metric don't have a dramatic increase over these years. And the active days on the platform reveals most of the listing centered around 1,000 days, and the yearly revenue varies among them.

3. Listing Accommodates: The accommodates features show a weak correlation with the yearly revenue. And the close cross-walk examination of these features indicates trends are diverse within each class. Five items out of thirty amenities contribute to a different yearly revenue with statistical significance.

4. Interactive Relationship: The guests' side interaction (review rating) influence the yearly revenue more than the hosts' side.

5. Geographical Information: Geographical location sectioning the region with different levels of yearly revenue.

5.2 Modeling Conclusion

After exploring the dataset, we used five supervised regressor algorithms () to train the predictive model and proceed forward the model optimization with two algorithms.

1. Preprocessing Data: The highly correlated features are dropped, and the normalization and standardization are applied. We split the train and test data at 80% : 20%. Besides, the test dataset is scaled by the train set parameters to prevent information leaking.

2. Model Optimization Techniques: The random search, grid search is applied for hyperparameters tuning with cross-validation.

3. Model Evaluation Metrics: Mean absolute error, mean square error and accuracy are used to access the model performance.

4. Model Comparison: We finalize the random forest with reduced features as the final prediction model for further implementation.

5. Model Limitation: The final diagnostic visualization shows that the model has the overprediction and underprediction, it has more prediction on the tails and less prediction on the center of the yearly revenue than it supposed to. Also, the model contains a fairly large number of features, and the single features all has small important, taking that into account, it would be hard to create a reliable and robust metric to preview the whole trends, which means further implementation, we have to collect a fairly large number of features to gain information.