

Final Report

Airbnb Price Analysis

Benyin Luo, Sam Kim, Jinwoo Oh

I 320D Applied Machine Learning

Spring 2023

University of Texas at Austin

1st May, 2023



Table of Content

1. [Introduction](#)
2. [Data Processing](#)
3. [Methodologies](#)
4. [Results](#)
5. [Conclusion](#)
6. [Annexure](#)
7. [References](#)
8. [Code](#)



Introduction

Background

A place to stay on a trip is essential. In the past, only hotels used to provide such services, but Airbnb has become a trend now. After COVID-19, the number of people traveling to other countries increased. And "Airbnb saw its revenues increase by 73% in 2021, after a 31% decrease in revenue in 2020 due the coronavirus pandemic shutting down travel. (*Home App Data Airbnb Revenue and Usage Statistics (2023)*, n.d.). There are many reasons why Airbnb continues to become famous, but the main reason is that "they are cheaper and provide better service than hotels (Baker, 2022)."

Target Users

The target users who want to see this analysis are 1) hosts who don't know how to set their Airbnb price, 2) guests who want to know if the Airbnb they picked is overpriced, and 3) other travel platforms who want to understand how the Airbnb pricing works.

Objective

Our primary goal is to predict the price of an Airbnb based on its features. The realization of the reason Airbnb became famous has made us think of a question: what factors make Airbnb's services cheaper and on what basis do hosts set their own Airbnb's price? Thus, we will try to understand the factors influencing Airbnb prices and create a predictive model that estimates the price of an Airbnb listing with its features. This will enable hosts to reasonably price their properties, guests to identify fairly priced accommodations, and other travel platforms to gain insights into Airbnb's pricing strategies.

Data Processing

Data Sources

- We got our dataset from kaggle website:
<https://www.kaggle.com/datasets/airbnb/seattle>
- The data set has 3819 rows and 92 columns.
- Binary, Numerical, Categorical, Text data are combined.

Initial Feature Selection

Initial Columns

[id, listing_url, scrape_id, last_scraped, name, summary, space, description, experiences_offered, neighborhood_overview, notes, transit, thumbnail_url, medium_url, picture_url, xl_picture_url, host_id, host_url, host_name, host_since, host_location, host_about, host_response_time, host_response_rate, host_acceptance_rate, host_is_superhost, host_thumbnail_url, host_picture_url, host_neighbourhood, host_listings_count, host_total_listings_count, host_verifications, host_has_profile_pic, host_identity_verified, street, neighbourhood, neighbourhood_cleansed, neighbourhood_group_cleansed, city, state, zipcode, market, smart_location, country_code, country, latitude, longitude, is_location_exact, property_type, room_type, accommodates, bathrooms, bedrooms, beds, bed_type, amenities, square_feet, price, weekly_price, monthly_price, security_deposit, cleaning_fee, guests_included, extra_people, minimum_nights, maximum_nights, calendar_updated, has_availability, availability_30, availability_60, availability_90, availability_365, calendar_last_scraped, number_of_reviews, first_review, last_review, review_scores_rating, review_scores_accuracy, review_scores_cleanliness, review_scores_checkin, review_scores_communication, review_scores_location, review_scores_value, requires_license, license, jurisdiction_names, instant_bookable, cancellation_policy, require_guest_profile_picture, require_guest_phone_verification, calculated_host_listings_count, reviews_per_month]

- Since there are so many features, we decided to remove columns that are obviously not related to price before feature selection.
- In addition, we decided not to include long text data such as the user's comment. These comments may be factors that affect prices, but it is difficult to train the models with this kind of data.
-

Columns after the Initial Feature Selection

[host_response_time, host_response_rate, host_is_superhost, host_total_listings_count, host_has_profile_pic, host_identity_verified, zipcode, property_type, room_type, accommodates, bathrooms, bedrooms, beds, security_deposit, cleaning_fee, guests_included, extra_people, minimum_nights, maximum_nights, availability_365, number_of_reviews, review_scores_rating, review_scores_cleanliness, review_scores_checkin, review_scores_communication, review_scores_location, review_scores_value, instant_bookable, cancellation_policy, require_guest_profile_picture, require_guest_phone_verification, price]

Selected Columns Explanation

- host_response_time: Time taken by the host to respond to a guest's message
- host_response_rate: Percentage of messages the host responds to
- host_is_superhost: Whether the host has a "Superhost" status on Airbnb
- host_total_listings_count: Total number of listings the host has on Airbnb
- host_has_profile_pic: Whether the host has a profile picture
- host_identity_verified: Whether the host's identity has been verified by Airbnb
- zipcode: Postal code of the property's location
- property_type: Type of property (e.g., apartment, house, etc.)
- room_type: Type of room being offered (e.g., entire home/apt, private room, shared room)
- accommodates: Number of guests the property can accommodate
- bathrooms: Number of bathrooms in the property
- bedrooms: Number of bedrooms in the property
- beds: Number of beds in the property
- security_deposit: Amount of security deposit required for the property
- cleaning_fee: Cleaning fee charged by the host
- guests_included: Number of guests included in the booking price
- extra_people: Fee for additional guests beyond the guests_included
- minimum_nights: Minimum number of nights required for a booking
- maximum_nights: Maximum number of nights allowed for a booking
- availability_365: Number of days the property is available for booking in a year
- number_of_reviews: Total number of reviews received by the property
- review_scores_rating: Overall rating of the property (out of 100)
- review_scores_cleanliness: Rating for cleanliness of the property (out of 10)
- review_scores_checkin: Rating for check-in process of the property (out of 10)
- review_scores_communication: Rating for communication with the host (out of 10)
- review_scores_location: Rating for the location of the property (out of 10)
- review_scores_value: Rating for the value of the property (out of 10)
- instant_bookable: Whether the property can be instantly booked without host approval

- `cancellation_policy`: The property's cancellation policy (e.g., flexible, moderate, strict)
- `require_guest_profile_picture`: Whether a guest is required to have a profile picture for booking
- `require_guest_phone_verification`: Whether a guest is required to have a phone verification for booking
- `price`: Price per night for the property

Feature Engineering

Drop the Rows with at Least One NULL Entry

By using the `dropna()` function, we dropped all rows with NULL values. As a result, only 1320 rows remain. Although there is only a third left, we think about 1300 rows are enough to train the model.

Numerical Data Transformation

Removed the unit symbol (such as '%', '\$', ',') from the numerical features.

- Transformed columns: `host_response_rate`, `security_deposit`, `cleaning_fee`, `extra_people`, `price`).

Binary Data Transformation

Made t/f data into 1/0 for easier training on a model.

- Transformed columns: `host_is_superhost`, `host_has_profile_pic`, `host_identity_verified`, `instant_bookable`, `require_guest_profile_picture`, `require_guest_phone_verification`

Categorical Data Transformation

Made categorical data into data that can be trained on a model through one hot encoding.

- Transformed columns: `host_response_time`, `zipcode`, `property_type`, `room_type`, `cancellation_policy`

Feature & Label Definition

Since we are predicting Airbnb prices, our label is 'price'. All columns other than Price are features.

Feature Scaling

We used MinMaxScaler because it had better accuracy than StandardScaler when we initially trained models. The scaler is only applied to numerical features.

- Transformed columns: host_response_rate, host_total_listings_count, accommodates, bathrooms, bedrooms, beds, security_deposit, cleaning_fee, guests_included, extra_people, minimum_nights, maximum_nights, availability_365, number_of_reviews, review_scores_value

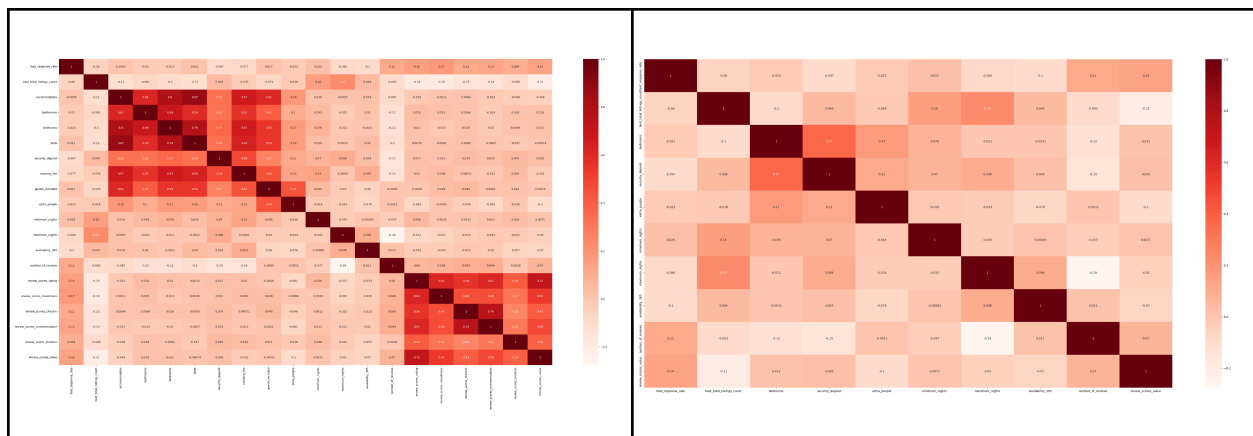
Methodologies

Feature Selection

We tried three different feature selection methods but ended up only using the correlation heatmap as it is the only method we are confident to use.

1. Correlation Heatmap

The heat map on the upper left shows that features “accommodation” is strongly correlated with “beds”; “bathroom” strongly correlated with “bedrooms”; “bedroom” strongly correlated with “beds”. Other than the top left, the review score ratings are also strongly correlated.



2. Variance Threshold

We also tried the variance threshold but it didn't really workout because we are not sure what the correct threshold number should be. After removing all the extra features, the model accuracy actually went down.

c) Variance Threshold

```
|:  from sklearn.feature_selection import VarianceThreshold

    selector = VarianceThreshold()

    selector.fit(numerical_features_df)

    print(selector.feature_names_in_)
    print(selector.variances_)

    selector = VarianceThreshold(threshold=0.02) # we can change the threshold
    selector.fit(numerical_features_df)
    print(f"Selected features: {selector.get_feature_names_out()}")

    numerical_features_reduced = selector.transform(numerical_features_df)

    ['host_response_rate' 'host_total_listings_count' 'bedrooms'
    'security_deposit' 'guests_included' 'extra_people' 'minimum_nights'
    'maximum_nights' 'availability_365' 'number_of_reviews'
    'review_scores_value']
    [0.0166639  0.01733266 0.0204812  0.009947   0.01514009 0.00504607
    0.00658619 0.20877071 0.10591089 0.01294517 0.01003139]
    Selected features: ['bedrooms' 'maximum_nights' 'availability_365']
```

3. Chi-squared test

For the Chi-squared test, we unfortunately did not find any numerical features that have p-values <0.05, this means that none of the features that are significantly dependent on labels.

c) Variance Threshold

```
|:  from sklearn.feature_selection import VarianceThreshold

    selector = VarianceThreshold()

    selector.fit(numerical_features_df)

    print(selector.feature_names_in_)
    print(selector.variances_)

    selector = VarianceThreshold(threshold=0.02) # we can change the threshold
    selector.fit(numerical_features_df)
    print(f"Selected features: {selector.get_feature_names_out()}")

    numerical_features_reduced = selector.transform(numerical_features_df)

    ['host_response_rate' 'host_total_listings_count' 'bedrooms'
    'security_deposit' 'guests_included' 'extra_people' 'minimum_nights'
    'maximum_nights' 'availability_365' 'number_of_reviews'
    'review_scores_value']
    [0.0166639  0.01733266 0.0204812  0.009947   0.01514009 0.00504607
    0.00658619 0.20877071 0.10591089 0.01294517 0.01003139]
    Selected features: ['bedrooms' 'maximum_nights' 'availability_365']
```


In the end, we chose to go with a correlation heat map. We dropped the following features ["review_scores_rating", "review_scores_cleanliness", "review_scores_checkin", "review_scores_communication", "review_scores_location", "beds", "accommodates", "cleaning_fee", "bathrooms"] as heatmap suggested

Models

We are predicting a continuous numerical value with multiple independent variables so it is a multivariate regression problem. Although we're less concerned with whether the model can predict the value exactly or not, we still used k-fold cross validation to see the accuracy of each model. We are just going to focus more on using error metrics to evaluate the predictive skills of each regression model.

We will look at to evaluate each model:

1. Mean Squared Error (MSE)
2. Mean Absolute Error (MAE)
3. The Pearson correlation coefficient
4. Cross-Validation

Features	<ul style="list-style-type: none">• Numerical: "host_response_rate", "host_total_listings_count", "accommodates", "bathrooms", "bedrooms", "beds", "security_deposit", "cleaning_fee", "guests_included", "extra_people", "minimum_nights", "maximum_nights", "availability_365", "number_of_reviews", "review_scores_value"• Categorical nominal: "host_response_time", "zipcode", "property_type", "room_type", "cancellation_policy"• Binary: "host_is_superhost", "host_has_profile_pic", "host_identity_verified", "instant_bookable", "require_guest_profile_picture", "require_guest_phone_verification"
----------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Linear regression

Pros:

1. Interpretable: Linear regression model is simple and interpretable. We can simply look at the coefficients in a linear regression to tell the strength and direction of the relationship between the independent variable and the dependent variable

2. Handles continuous and categorical variables: Linear regression model can handle both continuous and categorical independent variables, making it a versatile technique for regression problems.

Cons:

1. Sensitive to outliers: Linear regression model is sensitive to outliers in the data. Outliers can have a large effect on the regression line, leading to biased coefficients and poor model performance
2. Assumption of normality: Linear regression assumes that the residuals are normally distributed. Violation of this assumption can lead to biased coefficient estimates and inaccurate predictions'
3. Overfitting: Linear regression models can overfit the data if the model is too complex. This can lead to poor generalization to new data
4. Assumes linear relationship: Linear regression model assumes that the relationship between dependent variable and the independent variable is linear. This assumption may not hold true for all datasets, leading to poor model performance

	Mean Squared Error	Mean Absolute Error	Pearson correlation coefficient
Before feature selection	3229	37	0.755
After feature selection	3915	42	0.699

Decision Tree Regression

Pros:

- Non-linear relationships: Decision tree regression can handle nonlinear relationships between the input features and the target variable.
- No assumptions of linearity: Decision tree regression does not make any assumptions about the linearity of the relationship between the input features and the target variable.
- Robust to outliers: Decision tree regression is robust to outliers in the data since the extreme values never cause much reduction in the Residual Sum of Squares because they are never involved in the split.

Cons:

- Overfitting: Decision tree regression is prone to overfitting the training data if the tree is too deep or the minimum number of samples required to split a node is too small.

- Instability: Decision tree regression is an unstable algorithm because small changes in the data can lead to significant changes in the structure of the decision tree.
- Limited extrapolation: Decision tree regression is not good at extrapolating beyond the range of the input data. .

	Mean Squared Error	Mean Absolute Error	Pearson correlation coefficient
Before feature selection	3086	34	0.763
After feature selection	4836	39	0.645

Support Vector Regression

Pros:

- Non-linear relationships: Support Vector Regression can handle nonlinear relationships between the input features and the target variable.
- Robust to outliers: Support Vector Regression is robust to outliers in the data.
- Good at handling high-dimensional data: Support Vector Regression can handle datasets with a large number of input features.
- Can be used with different kernel functions: Support Vector Regression can use different kernel functions to map the input data into a higher-dimensional space.

Cons:

- Complexity: Support Vector Regression can be computationally expensive and difficult to implement for large datasets.
- Kernel selection: Support Vector Regression requires careful selection of the kernel function and other hyperparameters to achieve good performance
- Interpretability: Support Vector Regression is not as easily interpretable as linear models.

	Mean Squared Error	Mean Absolute Error	Pearson correlation coefficient
Before feature selection	7864	55	0.208
After feature selection	6843	48	0.560

Random Forest Regression

Pros:

1. Non-linear relationships: Random Forest Regression can handle nonlinear relationships between the input features and the target variable.
2. Robust to outliers: Random Forest Regression is robust and it can handle datasets with noisy and inconsistent data.
3. Handles high-dimensional data: Random Forest Regression can handle datasets with a large number of input features.
4. Low bias, high variance: Random Forest Regression has low bias and high variance, which can lead to better performance than other models when dealing with complex datasets.

Cons:

1. Overfitting: Random Forest Regression can overfit the training data if the number of trees is too high or the depth of the trees is too great.
2. Complexity: Random Forest Regression can be computationally expensive and difficult to interpret for large datasets.
3. Limited extrapolation: Random Forest Regression is not good at extrapolating beyond the range of the input data.

	Mean Squared Error	Mean Absolute Error	Pearson correlation coefficient
Before feature selection	2894	34	0.778
After feature selection	3867	36	0.688

Neural Network Regression

Pros:

1. Non-linear relationships: Neural Network Regression can handle nonlinear relationships between the input features and the target variable.
2. Robust to outliers: Neural Network Regression is robust to outliers in the data..

3. Can learn complex patterns: Neural Network Regression can learn complex patterns and interactions between variables that other models may not be able to capture.
4. Handles high-dimensional data: Neural Network Regression can handle datasets with a large number of input features.

Cons:

1. Overfitting: Neural Network Regression can overfit the training data if the network is too complex or the training data is too small.
2. Computational power: Neural Network Regression can be computationally expensive and difficult to interpret.
3. Requires large amounts of data: Neural Network Regression requires a large amount of training data to achieve good performance.
4. Black box: Neural Network Regression is often seen as a "black box" because it can be difficult to understand how the network arrived at a particular prediction. This can make it difficult to explain the model to others.

	Mean Squared Error	Mean Absolute Error	Pearson correlation coefficient
Before feature selection	2955	33	0.773
After feature selection	5728	47	0.590

K-fold Cross validation

```
Mean cross validation accuracy for model LR = -1.6751450035211996e+20
Mean cross validation accuracy for model DTR = 0.44269924952690093
Mean cross validation accuracy for model SVR = 0.022969727261593055
Mean cross validation accuracy for model RFR = 0.5579788458170938
Mean cross validation accuracy for model MLP = 0.30864678383190275
Best model is RFR with 5-fold accuracy of 0.5579788458170938
```

Results

We ran a train/test split of 90/10 for our models.

```
from sklearn.model_selection import train_test_split

features = airbnb_features_df.to_numpy()
labels = airbnb_label_df.to_numpy()

x_train, x_test, y_train, y_test = train_test_split(features, labels, test_size=0.10, random_state=42)

print (f"Training: Features' shape [no. of examples * feature vector size] = {x_train.shape}")
print (f"Training: Label's shape [no. of examples * 1] = {y_train.shape}")

print (f"Test: Features' shape [no. of examples * feature vector size] = {x_test.shape}")
print (f"Test: Label's shape [no. of examples * 1] = {y_test.shape}")
```

```
Training: Features' shape [no. of examples * feature vector size] = (1188, 75)
Training: Label's shape [no. of examples * 1] = (1188, 1)
Test: Features' shape [no. of examples * feature vector size] = (132, 75)
Test: Label's shape [no. of examples * 1] = (132, 1)
```

The mean squared error (MSE) and mean absolute error (MAE) measure the magnitude of error between the training and test data. Well performing models tend to have a lower MSE and MAE which indicate that predicted values more closely resemble actual values. The Pearson correlation coefficient ranges from -1 to 1, where 1 indicates a perfect positive linear relationship, 0 indicates no relationship, and -1 indicates a perfect negative linear relationship. The higher the Pearson correlation coefficient, the better the performance of the model.

Based off the results of the mean squared error (MSE), mean absolute error (MAE), and the Pearson correlation coefficient testing of the training and test data, it can be observed that before feature selection, all of the regression models have a moderate to a strong positive linear relationship since the MSE and MAE are relatively low and the Pearson correlation coefficient is over 0.75, with exception of the Support Vector Regression Model

	Mean Squared Error	Mean Absolute Error	Pearson correlation coefficient
Before feature selection	7864	55	0.208

However the Linear Regression, Decision Tree Regression, Random Forest Regression, and Neural Network Regression models' predictive performance has decreased after feature selection since the mean squared error (MSE) and mean absolute error (MAE) slightly increased while the Pearson correlation coefficient slightly decreased. The only model that benefitted from our feature selection via the correlation heatmap is the Support Vector

Regression, which saw a significant increase in its Pearson correlation coefficient from 0.208 to 0.560 and relatively small decreases in its MSE and MAE, but a decrease nonetheless.

After feature selection, the most significant model with the lowest MSE and MAE and one of the highest Pearson correlation coefficients is the Random Forest Regression model.

	Mean Squared Error	Mean Absolute Error	Pearson correlation coefficient
After feature selection	3867	36	0.688

After feature selection, the Linear regression model has a higher Pearson correlation coefficient of 0.699 but also has a higher MSE and MAE of 3915 and 42.


With this information and the 5 K-fold Cross Validation evaluation accuracy scores, we found that the Random Forest Regression model is our best performing model with the highest accuracy of 56%. Our models do not showcase the best performance due to low accuracy and suboptimal MSE, MAE, and Pearson correlation coefficient values, which could potentially be adjusted or improved with varying data splits or more extensive feature selection.

Conclusion

Based on our group's collective effort to approach our goal of predicting the price of an Airbnb based on its features, we conducted a thorough process of applying machine learning concepts into this project. After finding a dataset that represented our prompt with an ample amount of features and data points we cleaned the dataset by conducting a preliminary selection of features and implementing pandas tools to perform feature engineering such as feature definition, feature transformation, and feature scaling.

We then converted the data into a 90/10 training and test split to determine the mean squared error, mean absolute error, Pearson correlation coefficient, and K-fold cross validation values via regression models such as Linear Regression, Decision Tree Regression, Support Vector Regression, Random Forest Regression, and Neural Network Regression.

We then attempted feature selection with correlation heatmapping, threshold variance, and chi-squared testing but only saw meaningful results from correlation heat mapping and adjusted our features. After feature selection we ran the regression models once again. In conclusion the best regression model for determining price according to our selected features is the Random Forest Regression model. Although the model is not anywhere near perfect, it seems to make reasonably accurate predictions.



Reflecting upon the analysis process and our results, some limitations to take into consideration with our analysis is that we failed to take into account outliers which could have skewed the culmination of our results and brought in bias. The accuracy of our models were all subpar and our values of MSE, MAE and Pearson correlation coefficients worsened after our feature selection which may suggest overfitting to training data and is an area of concern. A few things we could have done differently in our analysis is to have adjusted the data splitting percentage and altered parameters within our regression models. Although we ended up utilizing regression to determine a price of an Airbnb based on its features, it was considered to give our label price a binary threshold to approach our prompt as a classification problem instead and to implement classification models (Logistic regression, Gradient boosting) alongside applied machine learning techniques such as dimensionality reduction methods (Principal component analysis) or implementing an ROC curve.

Annexure

Q. If you were to pass the project onto someone else, what would you suggest and what would you do differently?

A. If we were to pass the project onto someone else we would first suggest to be more selective of features and maybe even consider finding a different dataset since our results showcased heavy overfitting and suboptimal accuracy. We would suggest to look into more areas of error such as looking for outliers and testing different data splits. Another suggestion would be to consider changing the label price to a binary value and discovering a threshold by implementing an ROC curve in order to see if classification models would discover less overfit and more accurate results with this dataset.

Q. Why did you choose random forest refashion instead of the linear regression model while it actually gives a better correlation coefficient?


A. We also used cross validation to evaluate the model accuracy however we got a negative value for the linear regression model using cross validation hence we decided to choose the random forest which has the second highest correlation coefficient and the highest accuracy for cross validation.

References

Baker, L. (2022, August 30). *Reasons Why Airbnb Is Getting More Popular*. News from Wales.

Retrieved April 29, 2023, from

<https://newsfromwales.co.uk/reasons-why-airbnb-is-getting-more-popular/>



Home App Data Airbnb Revenue and Usage Statistics (2023). (n.d.). Business of Apps. Retrieved April 29, 2023, from <https://www.businessofapps.com/data/airbnb-statistics/>

Code

Project Github Link: https://github.com/starJin2003/Airbnb_Pricing_Analysis

All information/code used for the project is in the github. Read the Readme file please.