# BT5153 Team 11 "The Quintet" Group Project Report: Understanding the Preferences of Airbnb Consumers

**Low Chin Yang** [1 2] **Goh Aik Tio** [1 3] **Daniel Yuan Jiaxing** [1 4] **Nicholas David Gabriele** [1 5] **Li Xueqing** [1 6]

## Abstract

The rising popularity of paid homestay apps, where hosts offer shared accommodation to travelers in exchange for an agreed upon fee, has defined the sharing economy. Airbnb has emerged as the predominant market force with an estimated market capitalization of $31 billion in 2017 (Thomas, 2017). This study aims to discern the key contributing factors – including both directly observable traits, as well as latent features derived from sentiment analysis and natural language processing – that determine the composite score for an Airbnb listing in a particular market.

## 1. Problem Definition

### 1.1. Background

As of February 2020, the Airbnb platform advertises more than seven million listings across 220 countries, making it the most expansive peer-to-peer rental platform in the world (Airbnb Newsroom, 2020). News reports also point to wide adoption among younger consumers. A 2016 survey of 1,000 millennial travelers from the US, UK and China indicated that more than half of the respondents prefer their stay to be in accommodations with "cool, local neighborhoods," and up to 80% prioritized a unique travel experience (Airbnb Citizen, 2016).

The authors of this study acknowledge that there is no shortage of research on travel experience. Indeed, the hospitality sector has amassed troves of data since its early days, during which questionnaires were used by hotel management to obtain customer feedback and satisfaction scores. This study leverages and hopefully expands upon this rich literature. With the advent of online booking and travel platforms, traditional modes of evaluating customer satisfaction have been increasingly replaced by digital reviews and rating systems. Numerous studies have demonstrated how such ratings can influence booking decisions (King et al., 2014), so the business implications of understanding consumer sentiment are readily apparent. As such, it is critical for platform operators like Airbnb, as well as its host partners, to understand the determinants of user-generated review ratings.

### 1.2. Description

The problem is therefore framed as follows: what business insights can be gleaned from the abundance of homestay accommodation data and, specifically, what are the key characteristics that drive (positively or negatively) review scores? Answering this question is an ambitious and formidable task, so it is instructive to divide the problem into smaller, more manageable subcomponents:

- *How can big data be organized and utilized to generate actionable insights for business owners?* If a direct link can be established between some key performance indicator (e.g. revenue growth) and an observable response metric (e.g. composite review score),[1] then the problem reduces to one of supervised learning. A model can then be trained to understand the dynamics of the target variable according to some set of predictive features.
- *Where should the information be sourced, and how should it be processed?* While there are several unprocessed raw variables that can be used to describe the accommodation unit (e.g. number of bedrooms), rental policy (e.g. price, minimum length of stay), or host (e.g. years of experience), perhaps the most feature-rich fields involve raw text (e.g. property description, user-generated commentary), as they may convey pat-

[1]Business Analytics Centre, National University of Singapore, Singapore [2]A0003004L [3]A0191238A [4]A0186487J [5]A0206490J [6]A0186108A. Correspondence to: Low Chin Yang <e0319303@u.nus.edu>, Goh Aik Tio <e0337882@u.nus.edu>, Daniel Yuan Jiaxing <e0320725@u.nus.edu>, Nicholas David Gabriele <E0427318@u.nus.edu>, Li Xueqing <e0320346@u.nus.edu>.

*Proceedings of the $36^{th}$ International Conference on Machine Learning*, Long Beach, California, PMLR 97, 2019. 

[1]The research cited in Section 1.1 suggests that consumer feedback is almost certainly linked to one or more KPIs, but the goal of the study is to generate value-added insights regardless of which KPI is ultimately chosen.

terns and themes that cannot otherwise be expressed. This lends itself naturally to feature engineering via sentiment analysis and natural language processing.

- *Which features consistently explain variation in the composite review score, and why?* In order to establish a fair basis for comparison across candidate models, it is necessary to impose a loss function of some kind.[2] Features that are found to reliably reduce this loss (at least on average) are typically considered to be important drivers, provided that the underlying relationships conform with intuitive reasoning. This can be challenging to assess if the model lacks parsimony. In such cases, regularization techniques are introduced.

In the course of addressing these considerations, it is often necessary to examine other novel subproblems or challenge previously-held assumptions, especially as it concerns feature engineering. This process can be viewed as a control cycle from which a solution is arrived upon incrementally; however, the overarching problem statement as defined above remains the core focus of this study.

## 2. Understanding the Data

### 2.1. Sources

The data for the study is primarily sourced from the third-party website Inside Airbnb, which collects and retains information publicly listed on Airbnb (Cox, 2019). Neither the website nor its owner are affiliated with Airbnb, and the data is made available to the public under the Creative Commons CC0 1.0 license. The data is partitioned by city, spanning 40 major metropolitan areas across the globe, and further subdivided into fact and dimension tables, which can be readily joined.

The written reviews are captured in the native languages of the patrons. This would invariably frustrate the parsing of text-based commentary if the analysis were to encompass all 40 markets simultaneously.[3] Consequently, the approach for this exercise is to narrow the focus to a single market: the city of Singapore.

### 2.2. Collection Methods

Inside Airbnb is maintained by Murray Cox, an Australian-American activist who founded the website in 2016. The site's contributors employ automated algorithms to scrape and collate listing data and metadata from the Airbnb website. While the authors claim that the datasets provided have been appropriately processed and validated, Airbnb has aired criticism for perceived inaccuracies. These disputes involve the calculation of rates, the status of a property as active, and the presence of duplicate listings (Carville, 2019). Despite Airbnb's objections over authenticity, fellow "data activists" such as Tom Slee have been able to independently replicate the work performed by Cox (Katz, 2017). For that reason, Inside Airbnb remains a trusted and comprehensive data source for a proof-of-concept study, and it has been consistently relied upon in preceding studies of rental price and rating scores (Zhu et al., 2019; Wang & Nicolau, 2017).

### 2.3. Description

Within the Singapore segment, Inside Airbnb provides seriatim data for approximately 107,000 reviews corresponding to 7,900 unique listings, which comprise the bulk of information needed for this study. There are also separate tables containing detailed temporal and geographical data; these may be useful for auxiliary purposes like trend and segmentation analysis but, being nonidiosyncratic, are less likely to add predictive power. Table 3 provides a partial list of 40 raw variables that are deemed worthy of consideration for an eventual model, grouped across the following broad categories:

- **Response vs. predictor:** As discussed in Section 2.3, the response variable is the composite review score, known as `review_scores_rating` in the listings table. The response is continuous on a scale of 0 to 100, but this may be easily converted into a classification problem by establishing a threshold separating positive reviews from negative reviews (refer to Section 5.1). Every other raw input field serves as a potential explanatory variable.
- **Data type:** Each variable can be described as categorical (including binary indicators), numeric (including dates/times), or string. For example, `bathrooms` is an integer field counting the number of bathrooms, whereas `amenities` is a string field containing a list of amenities advertised by the host.
- **Broad descriptor:** It is often convenient to designate labels to certain groups of variables to better track how they originate and interrelate. Three broad classifications of review data are introduced, further detailed in Table 3:
  1. *Host attributes:* Describes the features of the hosted property (e.g. `room_type`).
  2. *Rental policy:* Stipulates the terms and conditions of the arrangement (e.g. `house_rules`).

[2] Minimizing the loss function in a high-dimensional feature space is no small feat, and ensemble methods often require some variant of gradient boosting to iteratively converge upon (or "learn") an optimal solution.

[3] Besides the issue of character encoding for non-Latin alphabets, there is also the question of how to meaningfully represent text embedding and parse the syntax of multiple writing systems.

3. *Guest inputs:* Documents interactions on the part the reviewer (e.g. `comments`).

Additionally, seven classifications of listing data are introduced:

1. *Host verification information:* Conveys biographical information about the host, such as `host_verifications` and `host_has_profile_pic`.
2. *Communication:* Evaluates frequency of communication between host and guest, as with `host_response_time` and `host_response_rate`.
3. *Policy of renting:* Describes a broad range of conditions enforced by the host. Examples include `require_guest_profile_picture`, `cancellation_policy`, and `minimum_nights`.
4. *Space offered:* Serves as a proxy for the amount of space available in the unit as indicated by `bathroom_per` and `bedroom_per`.
5. *Information about environment:* Includes characteristics about the imminent surroundings like `neighborhood_overview` and `is_location_exact`.
6. *Price:* Consists of all monetary amounts, specifically `price` and `cleaning_fee`.
7. *Experience related information:* Pertains to experiential data accumulated on Airbnb itself like `host_duration`, `host_listings_count`, and `number_of_review`.

## 3. Data Exploration

As described in Section 2, the dataset available from Inside Airbnb is replete with all of the elements needed to conduct a predictive analytics exercise. However, prior to building an actual model, it is essential to examine the basic structure of the data as well as the relationships present therein. This process is collectively referred to as data exploration, which often acts as the first line of defense to assess data quality, highlight important features, and identify possible constraints.

The raw data possesses several interesting aspects that benefit from further analysis:

- Univariate analysis of `review_scores_rating` suggests that the distribution is strongly skewed which, when considered in the context of a classification problem, makes this an imbalanced dataset, as illustrated in Figure 1. Depending on the class of model, these imbalances can be treated by, for instance, undersampling positive reviews, which constitute the majority class. There is also the question of how to deal with missing values; approximately 21,000 listings do not provide any rating.

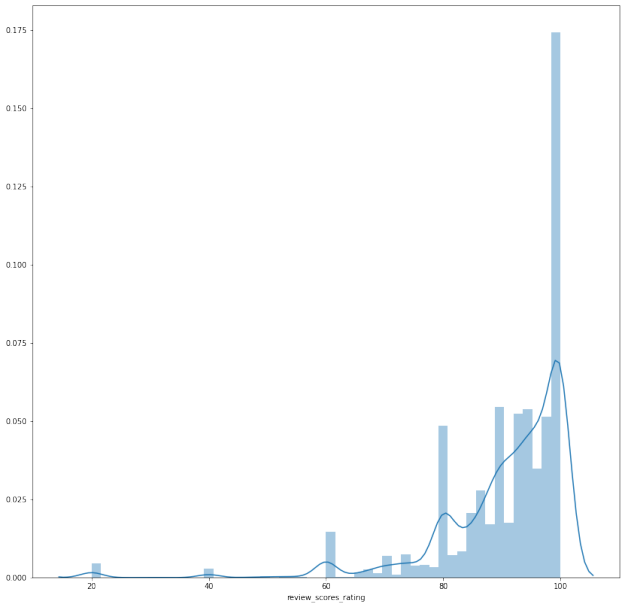


*Figure 1.* Relative frequency histogram.

- Multivariate relationships can be understood using either statistics (e.g. correlation) or visualizations (e.g. bar charts, heat maps). For instance, Figure 2 suggests that `review_scores_rating` associates positively with both `number_of_reviews` and `price`.[4] When two variables are highly correlated, it is pertinent to ask whether the relationship is spurious or causal. For example, `review_scores_rating` correlates positively with `price`. A statistical test can quantify the nature of this relationship and understand the possible direction of any causality.

- As mentioned in Section 2.3, text inputs may qualify as the most interesting and feature-rich raw variables. A full discussion of feature engineering is deferred to Section 4.4, but in the context of data exploration, it is helpful to conceptualize how text-based reviews influence the overall score. In particular, it is likely that there are certain keywords or phrases that strongly influence the sentiment of the review, and identifying these features is very much within grasp of machine learning.

[4]This may be innocuous, as in the case of a seasoned traveler who is inclined to give positive reviews, or point to nefarious activity, such as "astroturfing" (i.e. the flooding of reviews from inauthentic accounts).

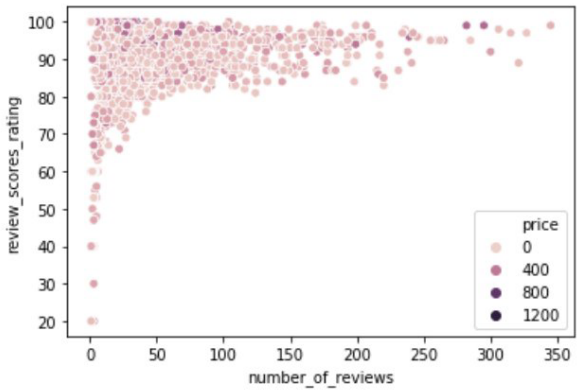


*Figure 2.* Multivariate scatterplot.

## 4. Pre-Processing Steps

### 4.1. Data Cleaning

While text-based fields are highly coveted for constituting a rich, high-dimensional feature space, they are also notoriously difficult to wrangle. Without thorough and effective cleaning, artifacts and syntactical errors in the data are bound to creep into the analysis. At its worst, uncleaned data can result in compile errors, making it impossible to extract any meaningful results. Even in the absence explicit errors, unprocessed text data can undermine the model pipeline at later stages, resulting in misdirection of important patterns and relationships. It is therefore essential that a careful approach to cleaning be taken before any feature generation or model building commences. Fortunately, a variety of Python libraries can be utilized for this purpose.

Some basic cleaning functions applied to the raw text fields include the removal of the following:

- *Numbers:* While numerical values do convey meaning in conjunction with words, they are highly context sensitive. Suppose a reviewer expresses that a property is 11.5 kilometers from the city center. This is useful information, but it is difficult to intuit sentiment from the digits '11.5' alone. In all likelihood, patterns of such high specificity would not recur in other observations, and an observer would need more information (e.g. the word 'far') to assess sentiment.

- *Punctuation marks:* These include periods, exclamation points, question marks, and other special symbols. Punctuation imparts a review with syntactical structure, and may occasionally contribute to sentiment (e.g. exclamation points to indicate emphasis). However, this is again very sensitive to the surrounding context, which is difficult to effectively leverage with open source tools.

- *Stopwords:* Disposable words such as 'a' and 'the' are unlikely to contribute to the overall sentiment of a written review, and it is common practice to remove these prior to model construction. Moreover, less frequent but extraneous words like 'whence' should be similarly ignored, since they do not contribute much to sentiment and may even confound the predictive algorithm, mistaking noise for signal.

- *Whitespace:* Similar to punctuation, whitespace is technically a form of user expression, but too abstract to precisely correlate with sentiment. Typographical analysis may be an interesting area for expansion on this topic, but is considered outside the scope of this work.

The decision is also made to remove observations that do not contain at least 5 characters in the `comments` field. The reasoning is that inarticulate commentary may indicate a lack of thoughtfulness and credibility on the part of the reviewer, or even a form of astroturfing, as mentioned in Section 3. The decision threshold is somewhat arbitrary, but it does ensure that extremely low-information content is excluded from the analysis.

Finally, missing values are treated by imputation. In general, the mean is used to populate missing values for numerical fields like `host_response_rate`, `positive_pct` and `vader_compound_avg` (see Section 4.3). The exception is `comments_count`, which is imputed with zero.

### 4.2. Language Detection

As noted in Section 2.1, reviews may be written in languages other than English. This is especially true for cities like Singapore, where visitors from around the world submit reviews in their native languages.

The challenges posed by multilingualism for text processing are self-evident. If not properly treated, the best-case scenario for predictive algorithms is to compartmentalize the data, establishing vastly different sets of rules for score prediction depending on the source language. More likely, the `comments` field would be ignored in favor of the more salient characteristics like `room_type` and `price`. To prevent this outcome, the instances require either some form of pre-processing (e.g. translation packages) or exclusion from the analysis.

The prospect of translating user reviews across languages without loss of information is extremely daunting, since there are drawbacks and limitations among even the most sophisticated tools.[5] However, language detection packages

[5]Several deep learning algorithms have been developed for language modeling and machine translation using recurrent neural

can be used to identify reviews written in foreign languages so that such observations can be omitted.

*Table 1.* Language identification tools.

| PACKAGES | FASTTEXT | LANGDETECT |
|---|---|---|
| DETECTED LANGUAGES | 98 | 32 |
| TOP LANGUAGE | ENGLISH | ENGLISH |
| OBSERVATIONS | 83,450 | 84,936 |

Table 1 shows the results of two language prediction algorithms. The FASTTEXT classifier is a custom language identification package, which is trained on a downloaded language dataset containing short sentences each tagged with a language, whereas LANGDETECT is an out of the box Python package (Kinnunen, 2018). When applied to the Airbnb data, the majority of reviews are classified as English for either package, although FASTTEXT detects more instances of foreign languages.

It is worth considering the consequences of false positives and false negatives in the context of language identification. A false positive is where the review is identified as English but is actually written in a foreign language, whereas a false negative is where the review is identified as foreign but is actually written in English. The latter is of particular concern as this needlessly eliminates valid data from which useful information can be gleaned.[6]

To minimize the prevalence of false negatives, observations are retained if the review is identified as English by *either* package, and only discarded if both packages detect a foreign language. Consequently, the number of observations retained following the language identification step is $88{,}440$, which is higher than would be the case if either package were used in isolation.

### 4.3. Sentiment Analysis

As explained in Section 1.2, the prediction of Airbnb ratings naturally lends itself to supervised learning, with the target field being the composite score. The rating system is the most direct way to measure the sentiment of the review, but the text-based `comments` field also avails itself to latent representation of sentiment.

There are compelling reasons to evaluate sentiment independently of the target variable. The first is that including a sentiment score as a predictor serves as a sanity check: it is expected that a higher review score confers more positive sentiment. If not, this may portend inconsistencies or anomalies in the data. Another reason is that a missing response value can be imputed from the sentiment score if it acts as a reasonable proxy. This kind of self-supervised learning can be especially effective in cases where the response data is sparsely populated or unreliable.

It is therefore prudent to compute a polarity score from the `comments` field to gauge the overall sentiment of a review. To do this, the Valence Aware Dictionary and sEntiment Reasoner (VADER) library is leveraged. The VADER package is a transfer learning tool that has shown impressive capability and versatility in a variety of social media applications (Pandey, 2019). The algorithm assigns a compound score on the interval $[-1, 1]$, with $-1$, $0$, and $1$ representing negative, neutral, and positive sentiments, respectively.

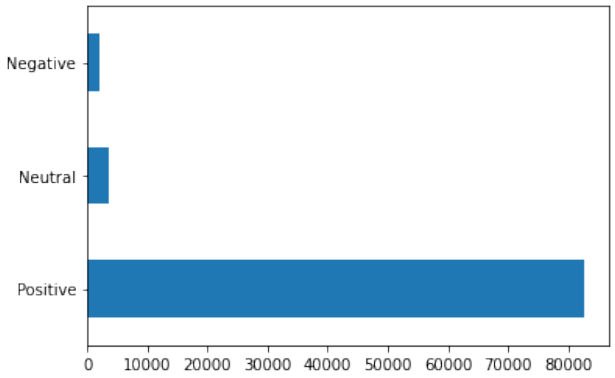


*Figure 3.* Frequency of VADER sentiment classifications.

As evident from Figure 3, the vast majority of reviews convey positive sentiment, as measured by the VADER compound score. This score is included as a feature along with those discussed in Section 4.4.

### 4.4. Feature Generation

Given the context provided in Section 3 and the high volume of data embedded in the `comments` field, a significant amount of time is dedicated to feature engineering. This process transforms the original data to enhance signal among the predictors, and employs techniques such as transformations, tokenizations, TF-IDF (*Term Frequency times Inverse Document Frequency*) scores, hand-tuned features, and autoencoder features.

**Transformations.** Several predictive modeling classes such as $k$-nearest neighbors (KNN) and support vector machines (SVM) require some concept of distance to render a prediction. This can be problematic if the numerical fields differ in scale, as the loss function becomes sensitive to

networks. When translated back and forth between the source and target languages, some form of degeneracy is commonly observed.

[6]False positives are also problematic to the extent that a machine learning algorithm will attempt to interpret foreign text, but should not dramatically impact findings so long as the quantity remains small.

those predictors involving larger units. To circumvent this, each numerical value can be mapped on the unit scale $[0, 1]$ for a given predictor, so that 0 represents the minimum and 1 represents the maximum. This has the added advantage of establishing parity with one-hot encoded variables when regularization techniques are performed.

**Tokenizations.** Raw text strings can be converted into tokens using the open source library NLTK. Words are assembled via regular expressions following the cleaning steps (e.g. removal of punctuation marks) applied in Section 4.1. This ensures that words with similar meaning but slightly different syntax are counted as a single instance.

**TF-IDF scores.** There are countless methods to engineer text-based data, but a blind application of the most common approaches – such as a bag-of-words – is unlikely to inspire meaningful results without a more concentrated effort. At a minimum, the stopwords noted in Section 4.1 should be removed. Even more importantly, some shortlist of meaningful nouns ('pool', 'breakfast') and adjectives (e.g. 'spacious,' 'clean') needs to be cultivated. One way to construct such a list is to rank the features according to some measure, such as the TF-IDF score, defined as $TF_i = \frac{f_i}{\max_k f_k}$, that tracks the frequency of a term $i$ in relation to the maximum number of occurrences of any other term $k$ (Leskovec et al., 2014).

For the purpose of text mining, priority is given to the `comments` field, which contains free-form commentary provided by the reviewer. Other text-based fields like `space` and `amenities` are less relevant to the sentiments of the consumer – which may well be captured by other host attributes – and are thus devoted less attention.

**Hand-tuned features.** While many of the aforementioned features are procedurally generated over the raw inputs, it is worth considering how these inputs may be specially transformed or combined to extract better signal. It is often the case that such hand-tuned features appeal to one's intuitive understanding and domain knowledge, and previous studies offer useful guidance in the context of Airbnb data. Examples include indicator variables for whether certain information is available (e.g. `transit_info` or `access_info`), `bathrooms` and `bedrooms` standardized for the number of accommodates, and the duration of the host's tenure on Airbnb measured in days (Zhu et al., 2019).

**Autoencoder features.** Features may also be generated autonomously using deep learning or transfer learning techniques. One such tool is autoencoder, which is a form of dimensionality reduction. A neural network model is trained to compress a high-dimensional input space into a much smaller feature space which, upon deconvolution, is able to reconstruct as much of the original input space as possible.

The dimension and performance of the autoencoder (as measured by loss) are left to the discretion of the modeler. There are inevitable tradeoffs between performance and parsimony, and an elbow plot can be constructed to determine an inflection point, beyond which additional complexity yields diminishing returns. As determined from Figure 4, three dimensions appears to strike a reasonable balance.

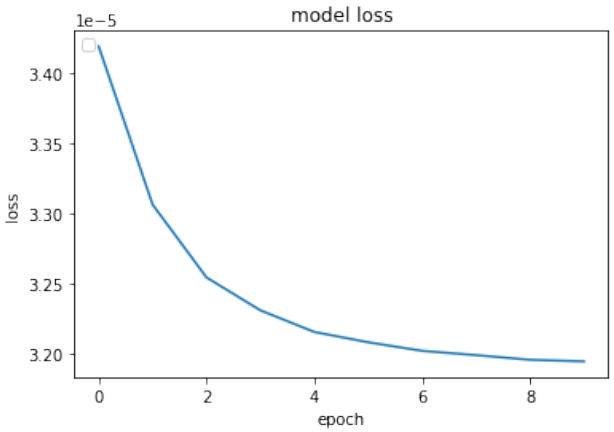


*Figure 4.* Elbow plot of autoencoder.

## 5. Modeling & Validation

### 5.1. Machine Learning Methods

The modeling process, as outlined in Section 1.2, requires the selection of an appropriate model class and evaluation framework. The prediction of review scores can be framed as either a regression or classification problem. A regression problem is preferred since the original response variable (i.e. rating score) is numeric. Coercing the response variable to a class requires some choice of threshold value, which is somewhat arbitrary and prone to information loss. This determination naturally influences the evaluation criteria discussed in Section 5.2.

Before any models are developed, it is advisable to partition the data into mutually exclusive and exhaustive training, validation, and testing subsets. Each of these are to be used for parameter estimation, hyperparameter tuning, and performance benchmarking, respectively. For this study, 20% of the original data is held out for testing purposes, while the remaining 80% is used for training and $K$-fold cross-validation where appropriate.

The autoencoder features described in Section 4.4 are especially useful in wrangling high-dimensional text data. This, in effect, represents a form of unsupervised learning, similar

to clustering or principal component analysis, both of which achieve dimensionality reduction on the feature space. This may alleviate the computational burden and provide other exploratory insights.

From there, a supervised ensemble model, such as a random forest of classification or regression trees, can be constructed based on the training data. Advanced techniques such as gradient boosting improve upon previous iterations by placing added emphasis on weaker predictions, while the structure of the model can be tuned via cross-validation. A final determination of model selection takes into account performance on the test data as well as overall parsimony and interpretability.

Rather than rely on a single model class, it is pragmatic to consider a broad range of predictive models. The following models are considered and, where indicated, tuned via cross-validation:

- *Decision tree:* A decision tree recursively partitions the feature space by splitting the data into subsets so as to minimize impurity at each node. The procedure terminates when a control parameter (e.g. maximum depth) has been invalidated or no further improvement can be made, and the resulting leaf nodes are the basis for new predictions. While easy to interpret, the greedy nature of decision trees tends to result in overfitting.

- *KNN:* The KNN algorithm is a form of instance-based learning where the observations themselves are used as the basis for prediction. While most often used for classification problems, it equally applies to regression. KNN offers the elegance of simplicity as it requires no training, but involves significant processing time at the prediction stage and often suffers from the curse of dimensionality. A grid search is performed to tune the model.

- *Logistic regression:* A logistic regression is an extension of the multiple linear regression framework where the linear component is linked to a logit response function. It has the advantage of high interpretability, with each coefficient for a given predictor representing its additive or multiplicative contribution to the response.

- *Elastic net regression:* This is a form of regularized regression, which constrains parameters via a complexity penalty. This involves scaling the sum of the squared coefficients (as in ridge regression), their absolute values (lasso), or a linear combination thereof (elastic net, which is a generalization of ridge/lasso). All three variations are attempted.

- *Random forest:* A type of ensemble method, random forests consist of a collection decision trees independently constructed from bootstrapped data and randomly selected predictors. While generally more robust than single decision trees, random forests are still prone to overfitting, and increased predictive power often comes at the expense of transparency. Nevertheless, with appropriate pruning and tuning, a random forest model can achieve impressive performance.

- *Gradient boosting machines:* Tree-based ensemble models can be trained to focus on weak learners and optimized via GBMs, such as XGBoost, LightBGM, and AdaBoost. Each of these algorithms leverage novel techniques and depend on a range of hyperparameters that control overfitting and convergence speed. All three are applied to the feature set and tuned to varying degrees.

- *Neural network:* A multi-layer perceptron model leverages nonlinear activations which, when passed through multiple hidden layers, may identify complex, abstract patterns in the input data. Deep learning models are notable for possessing high model capacity, but the resulting predictive power is often difficult to explain and interpret.

- *AutoML:* Automated machine learning is a general term describing the usage and tuning of all of the aforementioned techniques in a single, end-to-end process. While AutoML tools such as H2O necessarily bypass domain knowledge, they may serve as effective baselines with which to compare hand-tuned models.

### 5.2. Evaluation Criteria

In order to provide a fair benchmark for evaluation across the models, an objective performance metric is needed. For classification problems, this typically includes accuracy, precision and recall. For regression problems, this can be achieved via goodness-of-fit measures, such as root mean square error (RMSE) and mean absolute error (MAE), for which the goal is to minimize the respective quantity. RMSE can be viewed as a scale-dependent measure of accuracy, but it is sensitive to outliers. Formally, these evaluation criteria are defined as follows, where $N$ is the number of observations, $y_i$ is the observed response and $\hat{y}_i$ is the predicted response:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \quad (1)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \quad (2)$$

Table 2 provides the RMSE and MAE for each of the candidate models listed in Section 5.1 when predictions are rendered on the test dataset. From this, a number of observations can be gleaned:

- The tuned random forest model outperforms its rivals both in terms of RMSE and MAE with hyperparameters set as follows:
  - *Number of trees:* 1400
  - *Number of selected features:* Square root of total
  - *Maximum depth:* 80
  - *Minimum observations for splitting nodes:* 2
  - *Minimum observations in child nodes:* 4
  - *Bootstrap:* Without replacement
- In general, tuning helps performance across all model variants. This is to be expected when cross-validation is employed, as optimal performance on the validation sets is likely to result in improved performance on the test dataset.
- Decision trees, multi-layer perceptrons, and instance-based models do not perform as well under either measure. There is likely a significant degree of bias and memorization of the training data, resulting in overfitting when expanded to unseen observations. This phenomenon is especially common for neural networks, owing to high model capacity.

AutoML serves as a useful benchmark to assess the overall adequacy of the hand-tuned models. The number of models considered by the AutoML algorithm is limited to 20 in the interest of runtime. It ranks second and outperforms the remaining models including GBMs. This provides some assurance that the best performing model has been exhaustively tuned.

## 6. Insights

### 6.1. Variable Importance

On the official Airbnb website, the review score is assessed across six dimensions: accuracy, cleanliness, check-in, communication, location and value. Therefore, it is reasonable to expect that raw variables with obvious linkages to these criteria should act as strong predictors. For example, `host_response_rate` should influence the communication score, while `price` should affect the value score. It is also conjectured that descriptive data about a listing, such as the host attributes and guest inputs (as defined in Section 2.3), should inform customer satisfaction to some degree.

For tree-based methods such as the tuned random forest, variable importance can be quantified by computing the mean reduction in impurity for each feature. Using this criteria, several key features are identified as being important to predict the Airbnb rating score, as illustrated in Figure 5. These include a mix of both raw variables and engineered features, such as price, the number of listings/reviews, experience of the host, and the response rate.[7]

*Table 2.* Performance metrics across models.

| MODEL | RMSE | MAE |
|---|---|---|
| DECISION TREE | 16.56 | 9.67 |
| KNN | 12.12 | 8.01 |
| ELASTIC NET | 12.02 | 8.03 |
| LIGHT GBM | 11.27 | 7.11 |
| RANDOM FOREST | 11.41 | 7.17 |
| LOGISTIC REGRESSION | 11.96 | 8.15 |
| NEURAL NETWORK | 12.21 | 8.36 |
| RIDGE/LASSO | 11.96 | 8.15 |
| TUNED KNN | 11.62 | 7.77 |
| TUNED RANDOM FOREST | 10.82 | 6.87 |
| TUNED XGBOOST | 11.36 | 7.42 |
| TUNED LIGHT GBM | 11.05 | 7.15 |
| TUNED ADABOOST | 12.04 | 8.38 |
| AUTOML | 10.96 | 7.00 |

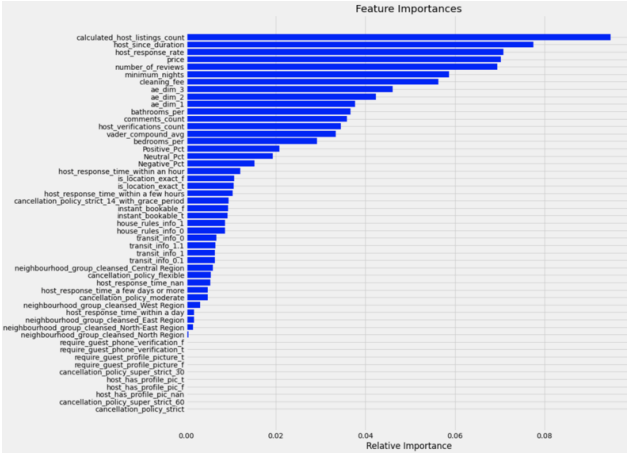


*Figure 5.* Key features ranked by mean impurity reduction.

It is no surprise that the top-performing model takes into account information relating to the host's historical records. It is reasoned that more experienced hosts (i.e. those with higher values of `host_since`) garner better ratings. Moreover, hosts who demonstrate care and commitment – such as those who respond to reviewer feedback – perform bet-

[7] Recall that reviews tend to be positive, and so these features tend to separate the very best reviews from the rest.

ter on average. Lastly, listings with more comprehensive descriptions of the hosted property (e.g. `transit_info`) contribute to higher review scores, indicating attention to detail on the part of the host. A key takeaway is that there is no substitute for experience, and hosts that manage multiple listings over many years are more likely to elicit positive responses.

### 6.2. Visualization

The dependencies between important features can best be understood with bivariate statistics and visualizations, including heat maps, which measure the strength of linearity between two variables as measured by Pearson correlation. In Figure 6, high correlation between the top three features can be observed. This suggests that the most successful hosts exhibit many of the same behaviors, which largely concern experience and responsiveness.

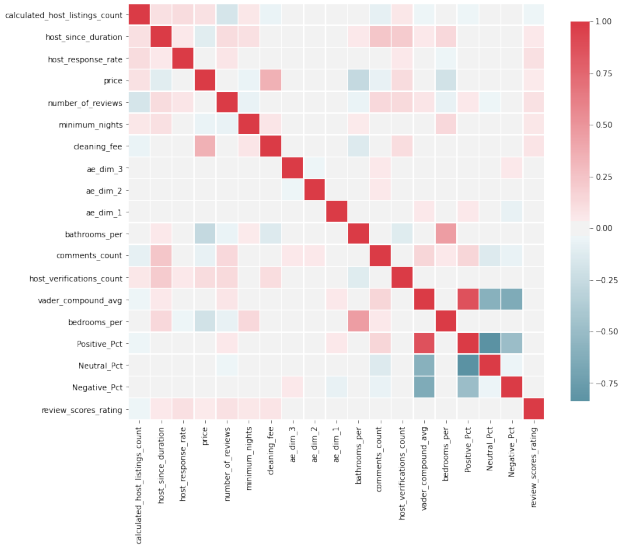


*Figure 6.* Correlation between top features.

It is also evident that the number of reviews are linked to several other key variables. This is because reviews tend to be favorable, and so there is some positive reinforcement between the volume of reviews and their underlying characteristics.

### 6.3. Challenges

Inside Airbnb aggregates review scores at the listing, rather than reviewer, identification level. This does not present an issue for the problem statement, which is concerned with the general movements of scores. However, a finer level of granularity could produce insights into the reviewers themselves, much like the corresponding host attributes.[8]

## 7. Conclusions

This study aims to predict listing review scores and their determinants. Results indicate that the best-performing model can predict review scores with a variance of 7%. The top five predictors include the number of listings belonging to the host, number of years of host experience, host's response rate, listing price and number of reviews for a listing.

The results highlight several noteworthy implications. First, among the six dimensions of review score mentioned in Section 6.1, the top predictors suggest that consumers favor "communication" and "value" among possible traits. In particular, consumers value host experience more than any other single characteristic, despite not factoring into the six dimensions of review score. Therefore, the results of this study suggest that consumers prefer seasoned hosts, whereas inexperienced hosts should seek to improve customer satisfaction by prioritizing response rate.

Additionally, the number of reviews plays a significant role. This study does not, however, find a strong connection between textual information and the review score. Sentiment predictors also rank lower in the feature importance plot. In other words, a listing with a high volume of (mostly negative) reviews can obtain a higher overall review score compared to another listing with very few (but positive) reviews. While this result is surprising, one possible reason is that in the digital arena, consumers pay more attention to the gross number of reviews rather than the content of any individual review. Moreover, the authors note a significant imbalance between the number of positive, neutral and negative comments in the dataset. While this analysis does not draw definitive conclusions on the impact of text-based information and sentiments on the review score, it is evident that hosts should solicit customer feedback whenever possible.

Peer-to-peer rental platforms have firmly established themselves as the bedrock of the sharing economy. This study leverages publicly sourced information to meaningfully quantify guest satisfaction for Airbnb listings in Singapore. Future research may expand upon this work to draw comparisons for other cities around the world. Review score prediction may even be extended to far-reaching industries like e-commerce, digital entertainment, and online travel agencies, providing insights for business owners to elevate their standards of service.

[8]For example, with access to more refined reviewer data, it might be possible to flag instances of astroturfing (see Section 3) or gauge a reviewer's propensity to express extreme sentiments.

## References


Airbnb Citizen. Airbnb and the rise of millennial travel, 2016. URL https://www.airbnbcitizen.com/wp-content/uploads/2016/08/MillennialReport.pdf.

Airbnb Newsroom. Fast Facts, 2020. URL https://news.airbnb.com/fast-facts/.

Carville, O. Meet Murray Cox, the man trying to take down Airbnb, 2019. URL https://www.bloomberg.com/amp/news/articles/2019-05-23/meet-murray-cox-airbnb-s-public-enemy-no-1-in-new-york.

Cox, M. Inside Airbnb. Adding data to the debate, 2019. URL http://insideairbnb.com.

Katz, M. A lone data whiz is fighting Airbnb – and winning. *Wired*, February 2017. ISSN 1059-1028. URL https://www.wired.com/2017/02/a-lone-data-whiz-is-fighting-airbnb-and-winning/.

King, R., Racherla, P., and Bush, V. What we know and don't know about online word-of-mouth: A review and synthesis of the literature. *Journal of Interactive Marketing*, 28(3):167–183, 2014.

Kinnunen, T. Classifying text with fastText in pySpark, January 2018. URL https://futurice.com/blog/classifying-text-with-fasttext-in-pyspark.

Leskovec, J., Rajaraman, A., and Ullman, J. D. *Mining of Massive Datasets*. Cambridge University Press, USA, 2nd edition, 2014. ISBN 1107077230.

Pandey, P. Simplifying sentiment analysis using VADER in Python (on social media text), November 2019. URL https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f.

Thomas, L. Airbnb just closed a $1 billion round and became profitable in 2016, March 2017. URL https://www.cnbc.com/2017/03/09/airbnb-closes-1-billion-round-31-billion-valuation-profitable.html.

Wang, D. and Nicolau, J. L. Price determinants of sharing economy based accommodation rental: A study of listings from 33 cities on Airbnb.com. *International Journal of Hospitality Management*, 62:120–131, April 2017. ISSN 0278-4319.

Zhu, L., Cheng, M., and Wong, I. A. Determinants of peer-to-peer rental rating scores: The case of Airbnb. *International Journal of Contemporary Hospitality Management*, 31(9):3702–3721, January 2019. ISSN 0959-6119.

*Table 3.* Listing of input fields from raw data.

| VARIABLE NAME | DESCRIPTION |
|---|---|
| **DEPENDENT VARIABLE** | |
| REVIEW_SCORES_RATING | RATING (SCALED FROM 0 TO 100) |
| **INDEPENDENT VARIABLES** | |
| **CATEGORICAL** | |
| *Host Attributes* | |
| HOST_RESPONSE_TIME | RESPONSE TIME (E.G. 'WITHIN A DAY') |
| COUNTRY_CODE | COUNTRY CODE |
| IS_LOCATION_EXACT | EXACT LOCATION (E.G. 'Y') |
| PROPERTY_TYPE | PROPERTY TYPE (E.G 'TOWNHOUSE') |
| ROOM_TYPE | ROOM TYPE (E.G. 'SHARED ROOM') |
| HOST_HAS_PROFILE_PICTURE | HOST PHOTO (E.G. 'N') |
| HOST_IDENTITY_VERIFIED | HOST VERIFICATION (E.G. 'Y') |
| *Rental Rules* | |
| INSTANT_BOOKABLE | ALLOWS INSTANT BOOKINGS (E.G. 'N') |
| CANCELLATION_POLICY | CANCELLATION POLICY (E.G. 'FLEXIBLE') |
| REQUIRE_GUEST_PROFILE_PICTURE | REQUIRES GUEST PHOTO (E.G. 'N') |
| REQUIRE_GUEST_PHONE_VERIFICATION | REQUIRES GUEST VERIFICATION (E.G. 'Y') |
| **NUMERIC** | |
| *Host Attributes* | |
| HOST_SINCE | START DATE OF HOST |
| HOST_RESPONSE_RATE | HOST RESPONSE RATE |
| BATHROOMS | NUMBER OF BATHROOMS |
| BEDROOMS | NUMBER OF BEDROOMS |
| *Rental Rules* | |
| ACCOMMODATES | NUMBER OF PEOPLE ACCOMMODATED |
| PRICE | ROOM RATE OF LISTING |
| SECURITY_DEPOSIT | SECURITY DEPOSIT IN DOLLARS |
| CLEANING_FEE | CLEANING FEE IN DOLLARS |
| GUESTS_INCLUDED | NUMBER OF GUESTS |
| EXTRA_PEOPLE | SURCHARGE FOR EXTRA GUESTS |
| MINIMUM_NIGHTS | MINIMUM LENGTH OF STAY |
| MAXIMUM _NIGHTS | MAXIMUM LENGTH OF STAY |
| *Guest Inputs* | |
| AVAILABILITY_X | AVAILABILITY RATE FOR PAST X DAYS |
| REVIEWS_PER_MONTH | NUMBER OF MONTHLY REVIEWS |
| **STRING** | |
| *Host Attributes* | |
| NAME | NAME OF PROPERTY |
| SUMMARY | BRIEF DESCRIPTION OF PROPERTY |
| SPACE | FACTUAL DESCRIPTION OF PROPERTY |
| NEIGHBORHOOD_OVERVIEW | DESCRIPTION OF SURROUNDINGS |
| NOTES | ADDITIONAL HOST COMMENTS |
| TRANSIT | TRANSPORTATION OPTIONS |
| ACCESS | ACCESSIBILITY OPTIONS |
| INTERACTION | EXPECTED INTERACTION WITH HOST |
| HOST_ABOUT | BIOGRAPHY OF HOST |
| AMENITIES | DESCRIPTION OF AMENITIES |
| *Rental Rules* | |
| HOUSE_RULES | HOUSE RULES (E.G. 'NO SMOKING') |
| *Guest Inputs* | |
| COMMENTS | FREE-FORM TEXT REVIEW |