Understanding Customer Satisfaction and Pricing in Airbnb

Combining Text Analysis and Machine Learning

Parker Jones, Vern Walker, Akshat Gadgil, Brody Kasprzak, Sienna Amorese

Abstract: Airbnb listings have gained popularity in the vacation rental and housing industries from its founding in 2008. They are marketed as a more "homely" experience than hotels, by welcoming in travelers to a homeowner's place of residence during vacation, also known as a short-term rental. As such, Airbnbs are scattered throughout destination cities and the surrounding areas, with a large variety of accommodations and spaces. Generally, they offer a much larger variance in experience than the average hotel, and thus, their prices are less standardized and not tied to any brand name or promise. What determines the satisfaction of a stay in an Airbnb?

We took an exploratory analysis approach to this problem, employing natural language processing and two-layer neural networks to perform "word2vec" word embedding analysis. We directly analyzed variables that affected overall satisfaction through ratings and reviews (user-generated content, UGC). We also used regression and random forest modeling techniques to understand the relationships between variables and price, under the assumption that price represents value, and value, experience. Thus, how models explain and predict price will propose a parallel to factors that determine overall experience of the customer. We found that emotional connection is paramount to good experiences, and that price and satisfaction are surprisingly uncorrelated.

1. Introduction

The short-term rental market in cities such as Austin, Texas, has become a highly competitive industry from the inception of Airbnb, the most popular rental service. It offers an enticing alternative from the standardized hotel experience. However, a listing's success is not only driven by well-reported and studied quantitative factors such as amenities, price, and location. Listings can be highly variable in the experience they offer, and as such, the satisfaction of a customer's experience is also variable, and not determined by well-known, accessible factors. Other qualitative aspects influence success, such as listing descriptions and neighborhood context, which play a critical, yet understudied, role in shaping customer expectations and satisfaction.

There is a key gap in the explainability of a listing's price and stay-satisfaction, and as such, we planned to take a deep learning approach to lesser-studied factors, such as user

generated content like reviews and ratings, and the variables most critical to modeling price. Using this methodology, we discover the underlying elements in Airbnb's success on the market, beyond tangible or well-documented factors of a listing.

Our data comes from Inside Airbnb, which describes 10,500 Airbnb listings in Austin, Texas. Each listing has 79 variables of structured and unstructured data. Structured data includes factors such as price, location, amenities, property-type, and host attributes, whereas unstructured data includes listing descriptions and UGC. Using this dataset, we can derive the key-players to customer satisfaction, pricing, and compare more ambiguous general perception with concrete features.

2. Methodology and Background

We used two main methods to determine the underlying drivers of success in Airbnb listings: supervised machine-learning and word2vec analysis. We used three different models to predict price: linear regression, gradient-boosted trees, and random forests. We chose to model price under the assumption that pricing is a direct signifier of value, due to the competitive nature of the short-term rental market. For our word2vec analysis, we cleaned the unstructured data to train word embeddings, using centroid similarity, pearson correlation, and cosine similarity for our exploratory analysis. Finally, we did an LDA analysis to find groups of words that were associated with a specific rating score and sentiment of the Airbnb customer.

2.1 Modeling

For modeling, we first changed missing values to nulls, handled outliers, and standardized variables using Spark DataFrames. We created new variables like "host response time" and "is superhost" to represent information in a way our model could easily understand and predict. We chose 8 predictor variables: "accommodates," "bedrooms," "bathrooms," "host_is_superhost," "has_availability," "room_type," "property_type," and "host_response_time." Finally, we split the data into a training and testing set, with a randomized 80/20 split for error measurement and evaluation. In all models, our continuous target variable, *Y1,* was Price, and all other variables, X_i, $i = 1 \ldots 8$ as predictors. Linear regression estimates coefficients B_i, $i = 0 \ldots 8$ by minimizing the sum of squared residuals of *Y1* using ordinary least squares:

$$Y1 = B_0 + B_1X_1 + B_2X_2 + \ldots + B_nX_n$$

This model finds the "line of best fit" for all data included in the model, and provides a good baseline for predicting price, and seeing what variables are most integral to the outcome using p-values of predictor variables.

Next, we used two types of specialized decision-tree models: the gradient-boosted tree and random forests. The gradient-boosted tree creates one final tree through iterative gradient descent on previous trees to minimize error (max iterations=100, max depth=5). Random forests average the outcome of several individual trees trained on bootstrapped samples (number of trees=50, max depth=10). Implementing this randomization measure throughout generation often gives better results on large or noisy data. Gradient-boosting and random forests are prone to overfitting and underfitting, respectively, and so are often used to counter the opposite problem on an original tree model. By using both, we can control error and fitting issues by measuring which predicts best on unseen data.

The decision tree algorithm works by finding the optimal information gain at each split in the data. That is, optimizing the most important threshold at any given point to separating the data into significant groups. This is represented by:

$$Gain(A) = Info(D) - Info_A(D)$$

Where the *Info* function minimizes the information needed to classify datapoints *D* into partitions, and thus minimizing randomness or "impurity" in a split of the data *D*. *Info(D)* or entropy, is the mean amount of information needed to identify the class or category of a data point in *D,* and *InfoA(D)* or conditional entropy, is the weighted average entropy of the subsets produced by splitting the data *D* on an attribute *A*. The decision tree then chooses biasedly towards tests with many similar outcomes to formulate a more generalized and accurate result using the gain ratio, then the tree uses the Gini Index (GI) to find the probability that a data tuple in *D* belongs to a specific class or "branch" in the tree. As a result, the algorithm works by choosing splits that minimize the uncertainty in the dataset. From there, the decision tree algorithm can be used in gradient-boosting and random forests.

We determined model success based on four metrics for the regression, and three metrics random forests and gradient-boosted trees: p-values (regression only), R^2, RMSE (root mean squared error), and MAE (mean absolute error). P-values measure the statistical significance of each individual predictor variable in our regression model only: that is, a hypothesis test that evaluates the probability of each individual predictor's coefficient under the null hypothesis that there is no present relationship (the slope is zero). A p-value under 0.05 signifies strong evidence of a relationship that affects the target variable, *Y1*. R^2 measures the amount of variance in the data explained by our model [0, 1], with values approaching 1 signifying a better model. RMSE measures the total error, specifically punishing data points that are further from the regression line by squaring error distances. MAE treats all error distances equally. RMSE is affected more heavily by outliers, and tells us better how well our line fits the data, whereas MAE tells us generally how much error our model accrued overall. Errors were both measured by testing model prediction accuracy on the test dataset.

Our regression model had an R^2 of 0.74, an RMSE of 1725, and an MAE of 404. P-values are seen in Figure 2.1, below. Our gradient-boosted trees had an R^2 of 0.76, an RMSE of 1672, and an MAE of 178. Our random forest had an R^2 of 0.74, an RMSE of 1724, and an MAE of 195. Overall, the gradient-boosted trees seemed to perform best by all measures.

2.2 Word Embedding and LDA

To perform sentiment analysis on the less structured data in our dataset, we used variables such as "perfect", "excellent", and "noise" for positive anchors, and "dirty", "noisy", and "disappointed" for negative anchors to compute word embeddings using deep learning. Then, we used word-level tokenization methods, removing stop words ("is," "and," etc.), punctuation, and other unnecessary information that may obstruct the important results. Then, we employed the word2vec algorithm to train word embeddings on the cleaned content of our dataset.

Word2vec uses a two-layer neural network on one-hot encoded words. For our purposes, we used continuous bag-of-words embedding (CBOW) due to its emphasis on recurring words and our smaller dataset. Any particular word, X_i, $i = 1 \ldots n$, is represented in a vector of length n, otherwise known as the "vocabulary size," where the words are encoded as binary values. That is, the presence of the word Xi is represented by a 1 in a word-specific index, and all other indices are represented by a 0. Then, this vector is fed into a two-layer neural network with a non-activation hidden layer of size m and a softmax output layer of size n that predicts the probability that every word is the correct word given surrounding context for every entry 1…n. At the end of training, the hidden layer represents commonalities between words, with nodes acting as a "weight" of each characteristic. These trained weights act as word embeddings for each word in the data.

Once we obtain the vector weights for each word, we used three different methods to evaluate embeddings and sentiment analysis. First, we computed cosine similarity between vectors in the vector space of all words to measure word associations. Cosine similarity between two vectors, A and B is computed by:

$$Cos(\Theta) = \frac{A \cdot B}{||A||\,||B||}$$

We hand-picked values of one vector, A, as (perfect, excellent, amazing, loved, wonderful) to represent positive sentiment, and (dirty, noisy, disappointed, broken, terrible) to represent negative sentiment, also known as "anchor words." By computing the similarities between our anchor words and the word embeddings of all words, we discovered which words were commonly associated with positive and negative sentiments, as well as the correlation to the score of listings which included these words, as seen in figures 2.1 and 2.2.

```
Words semantically similar to positive anchors (perfect, excellent, amazing, loved, wonderful):
  fantastic            sim=0.905  score_corr=+0.0971
  fabulous             sim=0.858  score_corr=+0.0829
  awesome              sim=0.841  score_corr=+0.0717
  incredible           sim=0.820  score_corr=+0.0986
  terrific             sim=0.810  score_corr=+0.0643
  great                sim=0.809  score_corr=+0.0694
  phenomenal           sim=0.740  score_corr=+0.0665
  lovely               sim=0.734  score_corr=+0.0952
  outstanding          sim=0.721  score_corr=+0.0769
  superb               sim=0.711  score_corr=+0.0613
  delightful           sim=0.651  score_corr=+0.0688
  exceptional          sim=0.651  score_corr=+0.0891
  unbelievable         sim=0.644  score_corr=+0.0322
  spectacular          sim=0.635  score_corr=+0.0523
  stellar              sim=0.634  score_corr=+0.0390
  unbeatable           sim=0.629  score_corr=+0.0317
  ideal                sim=0.625  score_corr=+0.0588
  brilliant            sim=0.616  score_corr=+0.0316
  adorable             sim=0.584  score_corr=+0.0503
  gorgeous             sim=0.573  score_corr=+0.0830
```

```
Words semantically similar to negative anchors (dirty, noisy, disappointed, broken, terrible):
  horrible             sim=0.779  score_corr=-0.0774
  gross                sim=0.745  score_corr=-0.0583
  noticeable           sim=0.743  score_corr=-0.0302
  dusty                sim=0.740  score_corr=-0.0440
  unbearable           sim=0.739  score_corr=-0.0235
  bad                  sim=0.722  score_corr=-0.0621
  strange              sim=0.704  score_corr=-0.0269
  messy                sim=0.703  score_corr=-0.0257
  usable               sim=0.691  score_corr=-0.0197
  disgusting           sim=0.685  score_corr=-0.0862
  poor                 sim=0.684  score_corr=-0.0709
  damaged              sim=0.676  score_corr=-0.0526
  moldy                sim=0.672  score_corr=-0.0384
  unsettling           sim=0.671  score_corr=-0.0272
  visible              sim=0.671  score_corr=-0.0245
  crooked              sim=0.671  score_corr=-0.0083
  disruptive           sim=0.669  score_corr=-0.0086
  useable              sim=0.665  score_corr=-0.0122
  wobbly               sim=0.664  score_corr=-0.0131
  overwhelming         sim=0.663  score_corr=-0.0141
```

Figures 2.1, 2.2: Cosine similarity output

Next, we used Centroid Similarity Differential, which splits the listings into vectors. The top 25% and bottom 25% by score. All the listing vectors in each group are averaged together. This gives two points for the dimensional space. One representing the high-rated listings and one for the low-rated. For every word embedding vector, it calculates the distance between the word and the centroid using cosine similarity, scoring words closer to the centroid as having a higher relevance, with the score. However, the vectors of particular words are often skewed by similarity.

Top 30 words most associated with HIGH-rated listings:

word	high_sim	low_sim	diff
leah	0.081988	-0.207830	0.289817
joel	0.199563	-0.078315	0.277878
mitch	0.070902	-0.204136	0.275038
nicole	0.163491	-0.105286	0.268777
wonderful	0.351291	0.084485	0.266806
ellen	0.051715	-0.212182	0.263897
christie	0.157965	-0.105276	0.263241
richard	0.015249	-0.247930	0.263179
kate	0.105247	-0.153692	0.258939
shelley	0.047862	-0.209695	0.257558
nicholas	0.110115	-0.146602	0.256717
mikes	0.109701	-0.146322	0.256023
steph	0.103898	-0.151716	0.255614
erik	0.126492	-0.129070	0.255562
nikki	0.165399	-0.089963	0.255362
chad	0.114572	-0.139820	0.254393
bryttie	0.052768	-0.198675	0.251443
craig	0.110009	-0.141092	0.251101
carl	0.029491	-0.220237	0.249727
thanks	-0.023138	-0.271953	0.248815
billy	0.148546	-0.099868	0.248414
maria	0.144629	-0.103462	0.248091
eva	0.221820	-0.024475	0.246295
trish	0.074136	-0.171995	0.246131
juli	0.044743	-0.200847	0.245590
jane	0.166501	-0.078642	0.245143
terrific	0.291655	0.046962	0.244693
jess	0.044525	-0.198624	0.243149
thea	0.152425	-0.088823	0.241248
neil	0.162900	-0.077633	0.240533

Top 30 words most associated with LOW-rated listings:

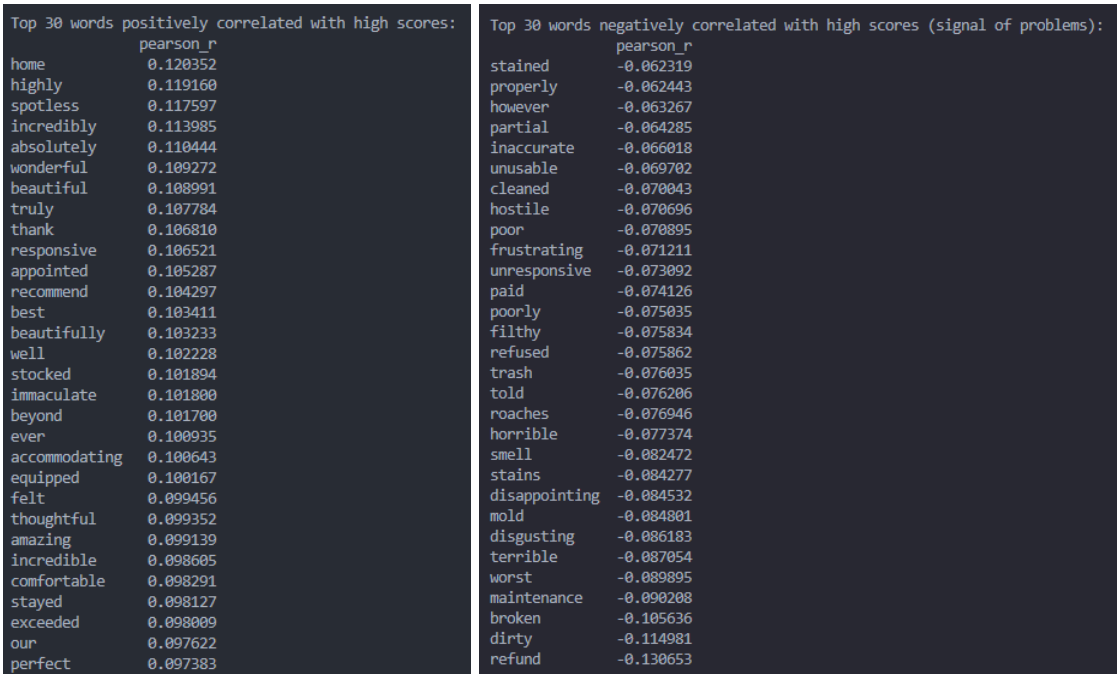
word	high_sim	low_sim	diff
apparently	-0.272590	0.069791	-0.342382
blind	-0.130390	0.209596	-0.339986
maybe	-0.381926	-0.045131	-0.336795
causing	-0.226410	0.105525	-0.331935
nasty	-0.121290	0.209574	-0.330863
sometimes	-0.266790	0.063281	-0.330070
cover	-0.213899	0.110987	-0.324886
cause	-0.352945	-0.029471	-0.323474
railing	-0.081908	0.241528	-0.323436
therefore	0.005880	0.327091	-0.321211
malfunctioning	-0.106154	0.213943	-0.320097
assuming	-0.346700	-0.027216	-0.319484
smelly	-0.044547	0.269847	-0.314394
irritating	-0.175165	0.139041	-0.314206
letdown	-0.224140	0.089642	-0.313782
bc	-0.183998	0.129692	-0.313690
trap	-0.173471	0.137313	-0.310783
sucked	-0.350627	-0.040670	-0.309957
intermittently	-0.001087	0.307664	-0.308751
random	-0.170466	0.137003	-0.307469
mess	-0.160633	0.145033	-0.305666
guess	-0.404027	-0.100602	-0.303425
strange	-0.133377	0.169748	-0.303125
that	-0.100993	0.202026	-0.303019
chemicals	-0.178674	0.124234	-0.302908
lingering	-0.125180	0.177449	-0.302629
flushing	-0.030472	0.271934	-0.302406
stairwell	-0.157938	0.143755	-0.301693
covering	-0.054939	0.245350	-0.300289
tripped	-0.166568	0.133457	-0.300025

Figures 2.3, 2.4: Centroid similarity differential output

Finally, we used the Pearson Correlation to determine which words were most positively correlated with higher scores. The Pearson Correlation *r* is computed by two vectors, *A* and *B:*

$$r = \frac{\sum_{i=1}^{n}(A_i - A_{bar})(B_i - B_{bar})}{\sqrt{\sum_{i=1}^{n}(A_i - A_{bar})^2 \sum_{i=1}^{n}(B_i - B_{bar})}}$$

which returns a correlation measure *r* in [-1, 1], signifying correlations with high scores of listings, as seen in figures 2.5 and 2.6.



Top 30 words positively correlated with high scores:

	pearson_r
home	0.120352
highly	0.119160
spotless	0.117597
incredibly	0.113985
absolutely	0.110444
wonderful	0.109272
beautiful	0.108991
truly	0.107784
thank	0.106810
responsive	0.106521
appointed	0.105287
recommend	0.104297
best	0.103411
beautifully	0.103233
well	0.102228
stocked	0.101894
immaculate	0.101800
beyond	0.101700
ever	0.100935
accommodating	0.100643
equipped	0.100167
felt	0.099456
thoughtful	0.099352
amazing	0.099139
incredible	0.098605
comfortable	0.098291
stayed	0.098127
exceeded	0.098009
our	0.097622
perfect	0.097383

Top 30 words negatively correlated with high scores (signal of problems):

	pearson_r
stained	-0.062319
properly	-0.062443
however	-0.063267
partial	-0.064285
inaccurate	-0.066018
unusable	-0.069702
cleaned	-0.070043
hostile	-0.070696
poor	-0.070895
frustrating	-0.071211
unresponsive	-0.073092
paid	-0.074126
poorly	-0.075035
filthy	-0.075834
refused	-0.075862
trash	-0.076035
told	-0.076206
roaches	-0.076946
horrible	-0.077374
smell	-0.082472
stains	-0.084277
disappointing	-0.084532
mold	-0.084801
disgusting	-0.086183
terrible	-0.087054
worst	-0.089895
maintenance	-0.090208
broken	-0.105636
dirty	-0.114981
refund	-0.130653

Figure 2.5, 2.6: Pearson correlation output

Finally, we performed LDA (Linear Discriminant Analysis) on the data to examine the probability distribution of certain words within a topic. Our LDA identified 6 topics from the word embeddings and calculated the mean review score and standard deviation for each topic, as shown in Figure 2.7. The “word cloud” (Figure 2.8) depicts the clusters of words that were placed into and how common they were, relatively, through size.

```
— Topic word profiles —
  Topic 0: clean, comfortable, neighborhood, location, definitely, recommend, perfect, responsive, quiet, needed
  Topic 1: group, pool, perfect, location, recommend, space, responsive, amazing, beautiful, beds
  Topic 2: beautiful, loved, amazing, perfect, definitely, recommend, wonderful, space, comfortable, hosts
  Topic 3: clean, location, easy, definitely, comfortable, close, recommend, check, responsive, super
  Topic 4: location, perfect, clean, space, comfortable, close, restaurants, recommend, neighborhood, easy
  Topic 5: location, apartment, clean, restaurants, easy, walking, walk, perfect, downtown, check

— Mean review score by topic —
  Topic 2 | score=4.929 +/-0.142 | n=842 | [beautiful, loved, amazing, perfect, definitely, recommend]
  Topic 0 | score=4.928 +/-0.130 | n=2,644 | [clean, comfortable, neighborhood, location, definitely, recommend]
  Topic 4 | score=4.917 +/-0.112 | n=1,009 | [location, perfect, clean, space, comfortable, close]
  Topic 1 | score=4.876 +/-0.212 | n=1,216 | [group, pool, perfect, location, recommend, space]
  Topic 5 | score=4.831 +/-0.228 | n=1,009 | [location, apartment, clean, restaurants, easy, walking]
  Topic 3 | score=4.652 +/-0.492 | n=2,189 | [clean, location, easy, definitely, comfortable, close]
```

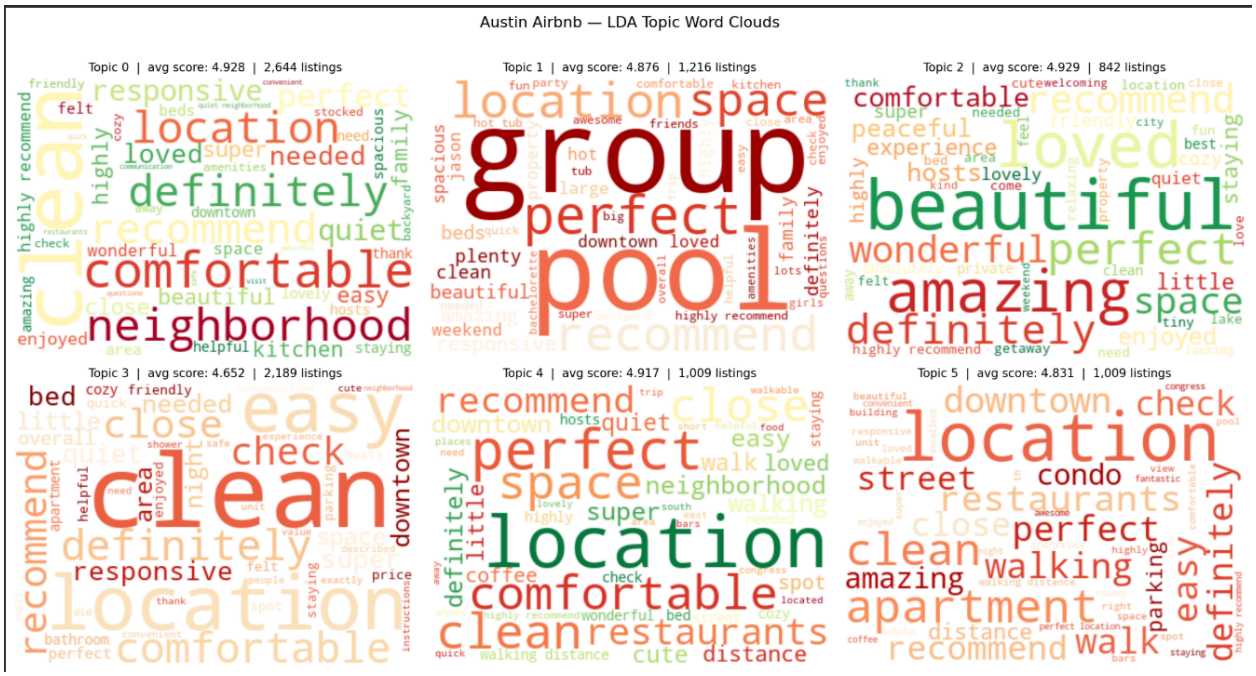


Figure 2.7, 2.8: Linear discriminant analysis output and "word clouds"

3. Results

Through our modeling and word2vec analysis, we find several compelling results about how Airbnb satisfaction and value are determined. Trees appear to perform better than simple linear regression on the dataset, which suggests nonlinear relationships between our predictor variables and price. By all of our chosen metrics, the gradient-boosted tree appeared to perform significantly better than a random forest, meaning that iterative error mitigation, rather than bootstrapped samples, fit best. Due to the high variability of Airbnbs, a potential explanation could be that the gradient-boosting algorithm is needed to fit well to diverse data, while the further randomness introduced by a random forest further obscures an optimal prediction due to lack of data, rather than successfully mitigating manageable noise.

Pricing appears to be primarily driven by tangible, physical features, such as amenities. However, this study of price appears to be integrally flawed when trying to decipher the customer satisfaction of a listing. As seen by the chart in Figure 3.1, sample sizes tend to be relatively even, while prices have a vast range. However, the difference in average review score seems to be insignificant and uncorrelated with price. Perhaps the key takeaway is that while price is determined mainly by amenities, it is the expectations of an Airbnb stay, given the context of described amenities and price, that matters more than anything. Review scores may tie more closely to whether or not an Airbnb matches its description and value proposition. This theory makes sense, given how competitive Airbnb's market has become, that prices describe expectations well because they are managed by the very thing. However, the satisfaction of these preconceived notions about an Airbnb is not necessarily measured well by price.

Price Quartile	Category	Price Range	Avg Review Score	Sample Size (n)
Q1	Budget	$8 – $85	4.7724	2,228
Q2	Mid-Low	$86 – $130	4.842	2,247
Q3	Mid-High	$131 – $215	4.8733	2,214
Q4	Premium	$216 – $50,000	4.8769	2,212

Figure 3.1: review scores by price and number

Our word2vec analysis gave us more insight on what customers consider important in Airbnbs, and what factors influence the satisfaction of these expectations. Almost all results highlight cleanliness or lack thereof as huge drivers of satisfaction/dissatisfaction. Words such as "spotless" had notable (but milder) positive associations with score, whereas words such as "dirty" or "filthy" were commonly the most significant negative associations (Figures 2.2, 2.4, and 2.6). The centroid similarity differential also picked up negative correlations with accessibility and usability of a site, mentioning railings, or tripping, which builds on the hypothesis that the quality of the stay is highly dependent on an assumption of a clean, usable, and orderly space.

Notably, the word "refund" had the highest negative contribution to score, at -0.131, in the Pearson correlation output (Figure 2.6). This may suggest that, despite all issues with an Airbnb stay, problems with refund (not offered, or a lack of a system to appeal) for an undesirable experience is the biggest driver for negative reviews and ratings. People with negative experiences who cannot receive a refund will likely leave negative feedback as their only form of recourse against an unjust or unsatisfying experience.

Positive associations were less telling. Most words associated with positive review scores were general exclamations of positive sentiment such as "beautifully," or "wonderful." However, as found in Figure 2.3 with the centroid similarity differential output, reviews with high scores are disproportionately associated with names ("Leah," "Joel," etc.). This suggests that hosts are critical when an Airbnb customer is reporting the positive aspects of their stay. Host friendliness, punctuality, and accessibility may not be large drivers of dissatisfaction, but a positive host experience seems common and notable when a person had a good experience.

The highest positive association in the Pearson correlation output was the word "home," which may suggest a high occurrence of sentiments akin to "it feels like home." Given Airbnb's brand promise as a more unique and "homely" experience compared to hotels, this could be intertwined with the expectations of guests, and commonly reported in positive experiences.

When comparing the two, we find that cleanliness is a huge negative factor in determining review score, while positive reviews mention cleanliness less overall, and that this is

likely an expectation rather than a pleasant addition. Furthermore, positive experiences focus more on emotional language than on specifics of the stay. How an Airbnb and a host connect emotionally with a person matters far more in positive experiences. Heavy and negative emotional language is also used in negative correlations, such as “trash,” “nasty,” and “letdown.”

Furthermore, the LDA analysis shows us groupings of common positive aspects of an Airbnb. Group 0 appears to describe location and amenities in terms of overall coziness or comfort, with an emphasis on words like “kitchen,” “location,” “neighborhood,” and “comfort.” Group 1 includes words like “pool,” “spacious,” and “recommend,” which may describe pleasant amenities above-and-beyond expectations. Group 2 has general positive sentiment, with words like “beautiful,” or “amazing,” and group 3 describes an organization and orderliness of the stay, including “clean,” “easy,” and “comfortable.” Group 4 describes a similar sentiment to group 0, with a bigger emphasis on location. Finally, Group 5 includes “location,” “walking,” “restaurants,” and “parking,” which probably refers to walkability and location relative to nearby attractions.

Group 3 had the lowest of all positive review scores at 4.6, enforcing the idea that cleanliness and orderliness are more of an expectation than a notable positive experience, and that it may not have as big of a positive impact as a negative one. However, group 3 and group 0 had approximately ~1000 more samples than other groups, meaning that although cleanliness had a lesser effect on positive ratings, it was mentioned very frequently, along with general comfort.

Overall the groups were quite similar. Location had a large presence throughout positive reviews, and people most often mentioned abundance of space, cleanliness, nearby restaurants, and a pool as great contributors to their good experiences with an Airbnb listing.

Further analysis could be done with mathematical comparison of listing reviews and listing descriptions, or other more targeted analysis to diagnose specific questions in the data. As is expected, Airbnb’s large variance in properties amounts to large noise and a larger data requirement to diagnose the more specific influencing factors.

5. Conclusion

Airbnb pricing and expectations are primarily driven by property features, location, and other amenities, while customer satisfaction hinges on experience, emotional connection, and host accessibility and friendliness. Traveling is not a cheap expense, and as a short-term rental company, the ability to satisfy a high price tag and the expectations that come with it, is the most important determiner of listing success. That should not be so surprising, given the value propositions of Airbnb and how important a positive stay is in a vacation experience. As with many aspects of life, it is the subjective and emotional perception of value that resonates most in treasured experiences.

6. Sources

Inside Airbnb. *Austin, Texas Data (Listings and Reviews CSV Files, 16 Sept. 2025)*. Inside Airbnb, https://insideairbnb.com/get-the-data/. Accessed 17 Apr. 2026.