Helping owners understand the rental market.

Presented to:
Airbnb Community Support team

By:
Carl Horwitz

Background

My wife and I were getting ready to go out of town for a week and were discussing how we should explore renting our place on Airbnb to supplement our income.

We started to fill out the listing, but had no idea how our place compares to any other listings, and in turn, no idea how to price it.

I started clicking on listings in our neighborhood, then on similar sized houses, and thought, there has to be a better way.

My goal is to make a tool to help owners understand what they have and how to price their home.

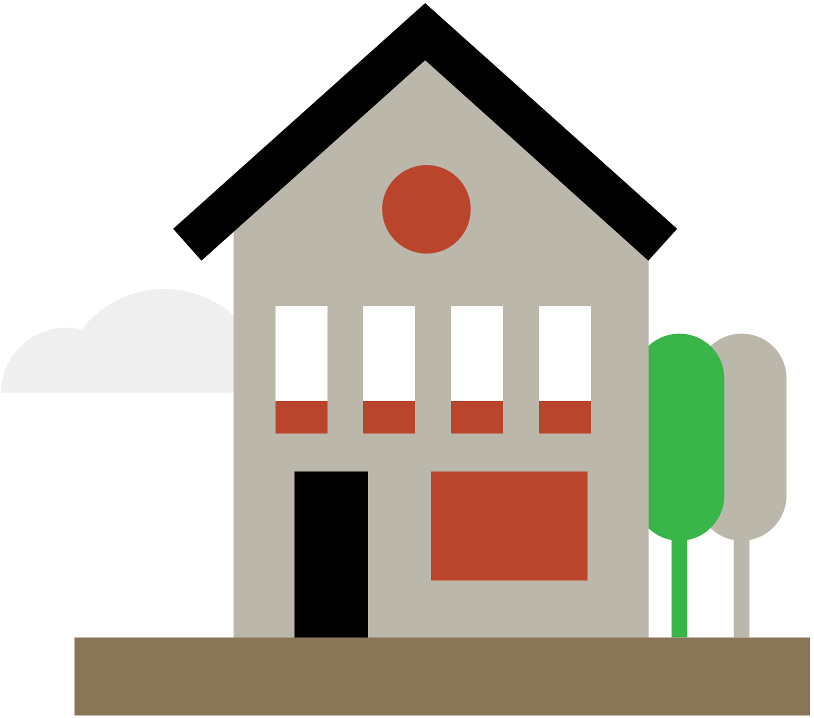


Table of Contents

Data Collection

Inside Airbnb - Listings

01

Downloaded a dataset that contains information about every rental in Denver, 5250 homes with 75 features per home.

Inside Airbnb - Reviews

02

Downloaded a dataset that has every review from 2017 through March of 2023. A quarter million individual reviews across more than 4000 listings

WalkScore API

03

Utilized the latitude and longitude included in each listing to pull the Walk, Bike and Transit Scores using the WalkScore API.

Data Cleaning and Preparation

Ways to make the data more usable

How to deal with nulls and outliers?
ie: reviews, price

Feature Engineering
ie: years listed

Converting data types
ie: percents, dollars, “1 bath”

Created bins
ie: shared status

Exploring trends

Natural language processing (NLP)

Natural Language Processing:

I conducted sentiment analysis on all descriptions using Vader, which is a computational process for determining whether a piece of writing has a positive, negative, or neutral tone.

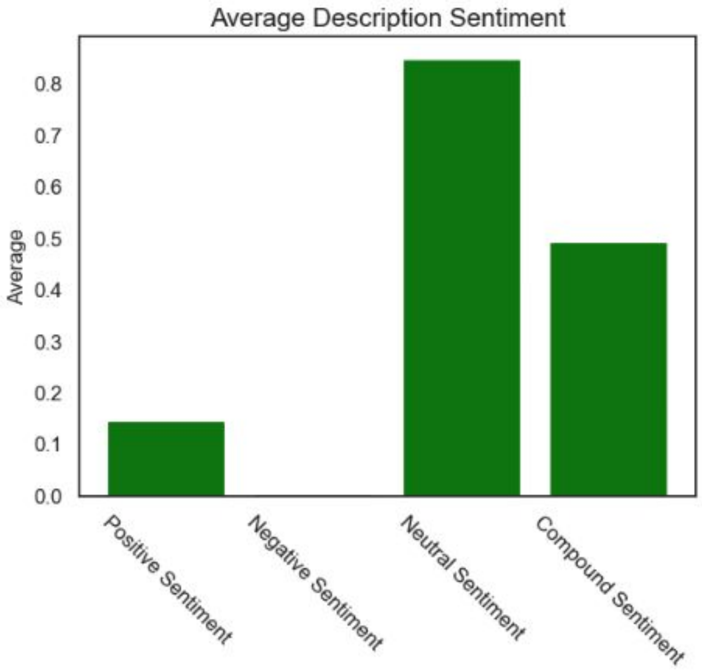
The analysis returns numeric values on a 0-1 scale, with the number indicating the probability that a given input fits into that category.

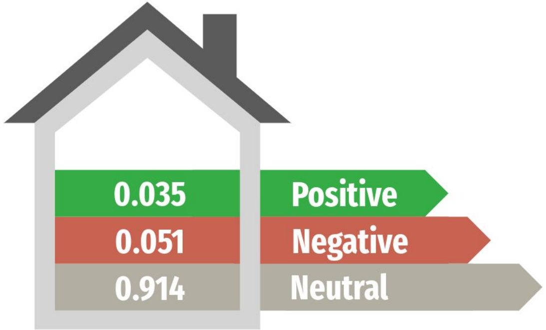
Additionally, Vader combines individual scores to create a compound score, which provides a more "holistic" assessment of sentiment. A compound score of -1 suggests a strong probability of negativity, while a score of +1 suggests a strong probability of positivity.

NLP Example

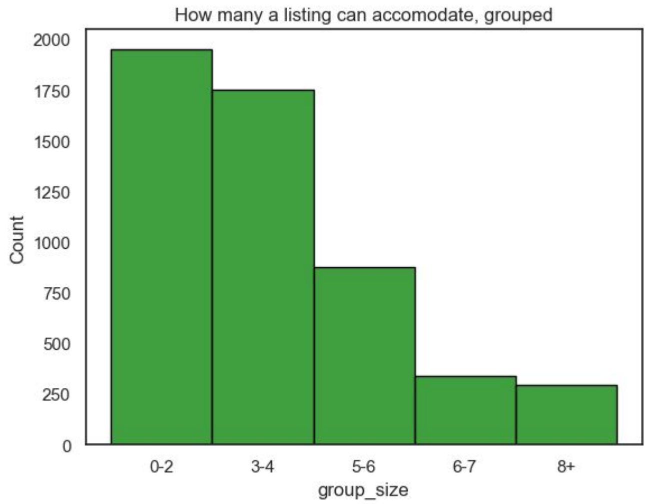
Property Description:
“Fire up the BBQ grill on the deck behind this spacious, renovated home. Find a private place to recharge with a custom curated interior, mid-century furnishings, a covered backyard dining area, a fire pit, and a basement games room. Wake up with a cup of coffee and a newspaper in the sun filled living room.

Unwind with a book and a beverage on the shaded front porch or back patio. Newly redecorated with tons of light, 3 big screen TV's with Netflix, Prime and cable, living room, family room, private office, and free curbside.” parking”

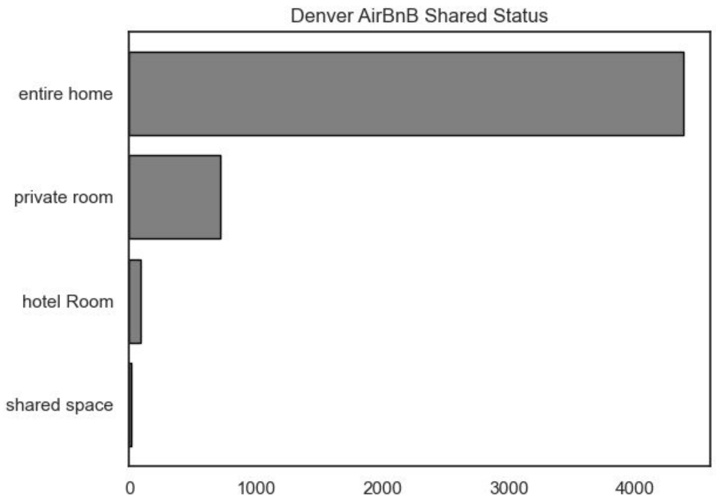


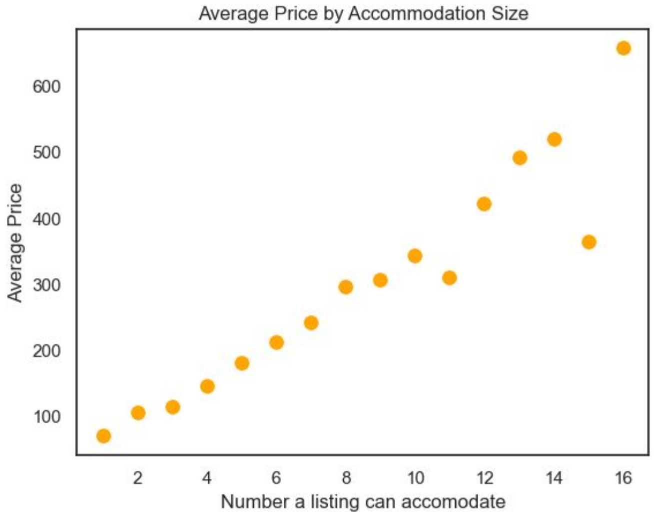


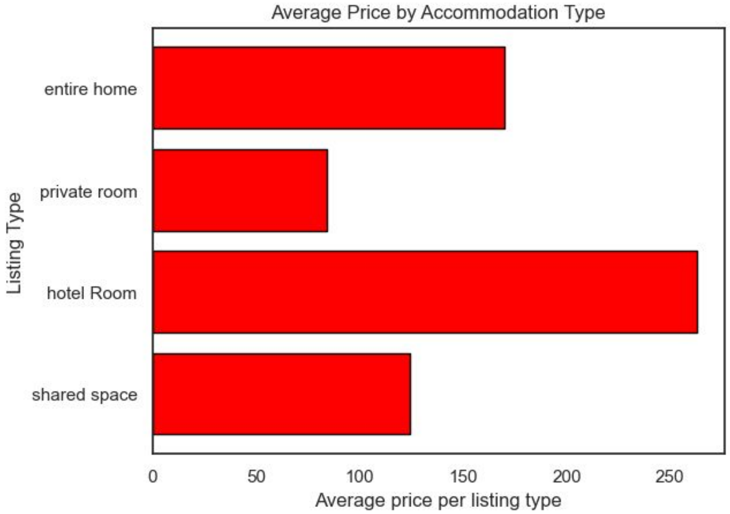
0.22 Compound Score



Type and Size of Accomodations

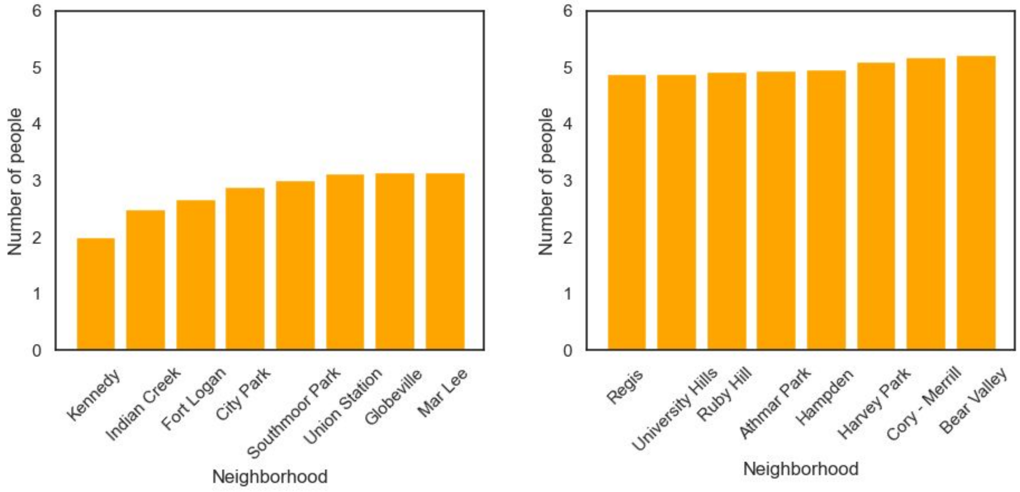




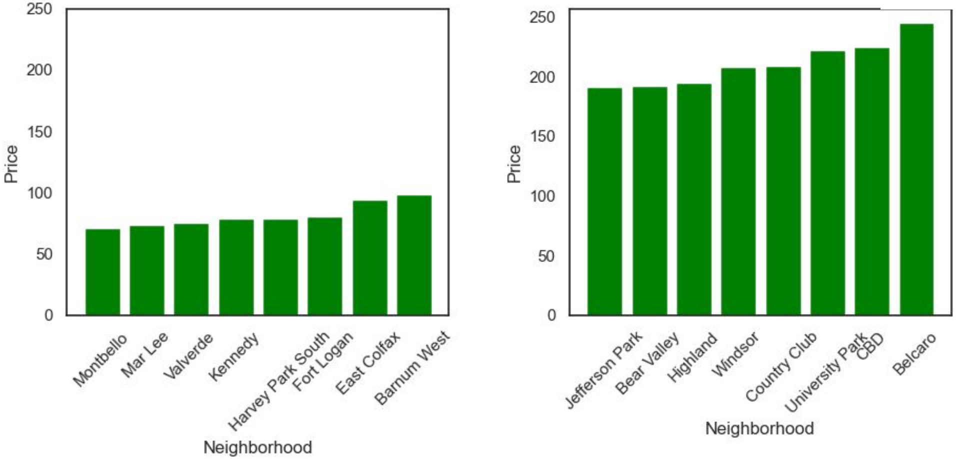


Location: Neighborhood

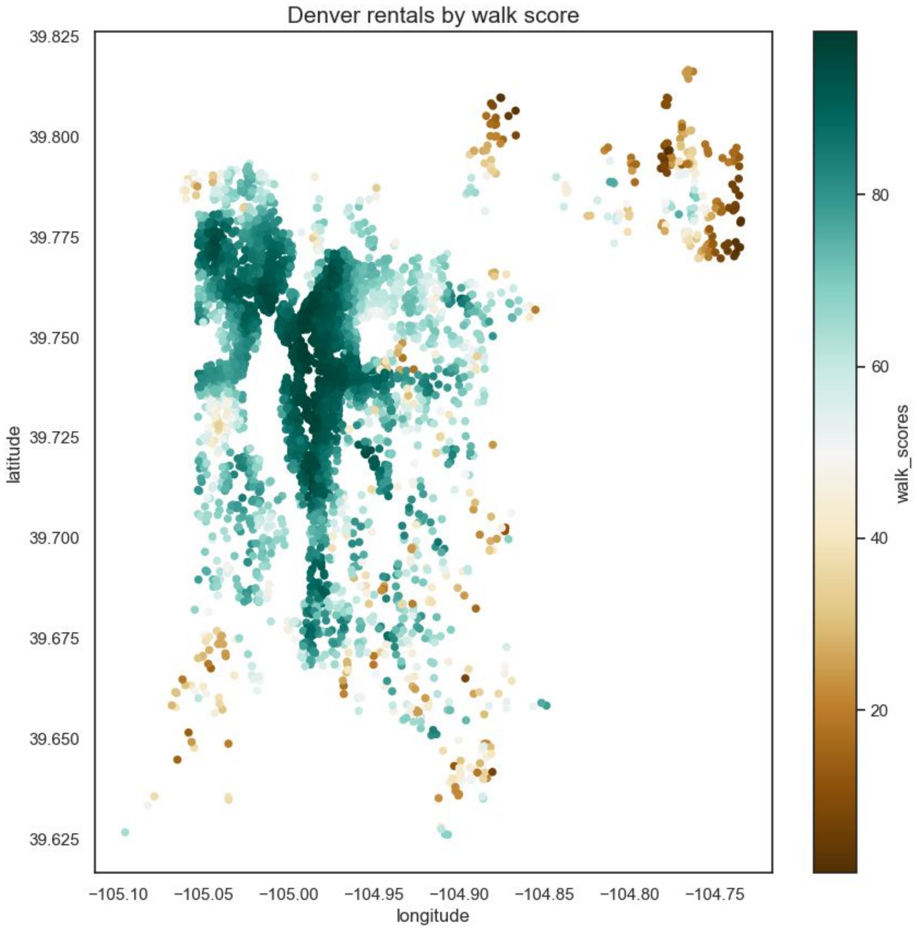
Average number of guests by neighborhood

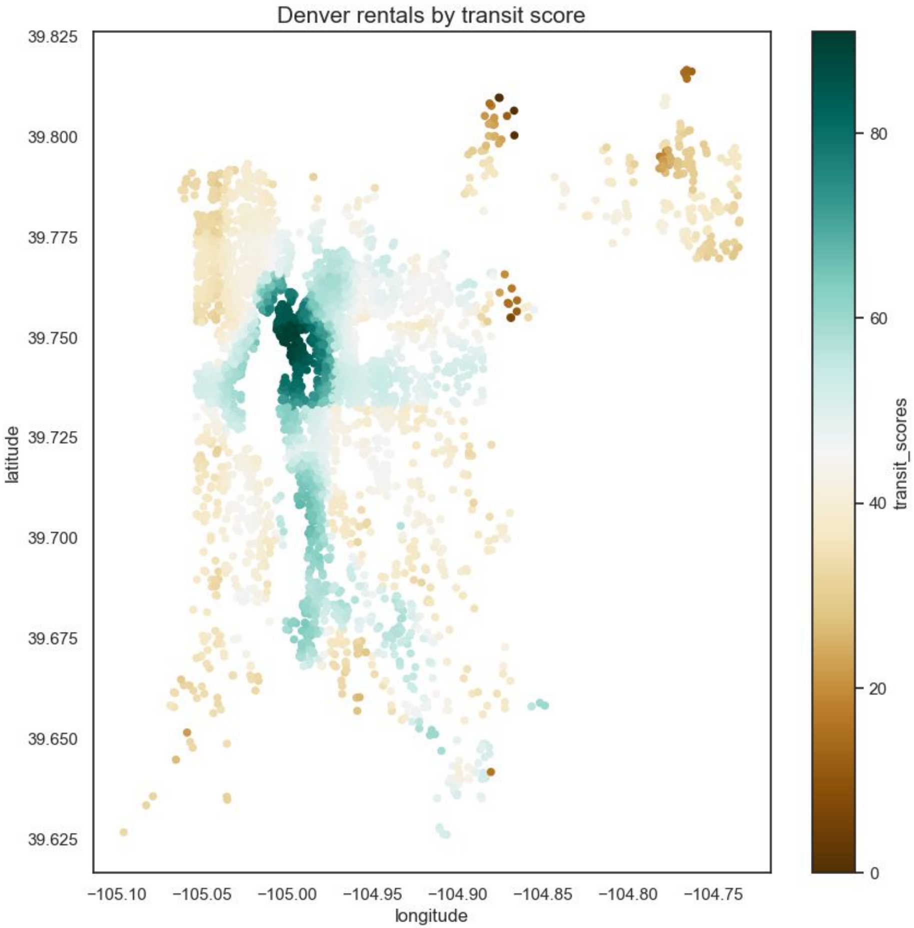


Average price by neighborhood

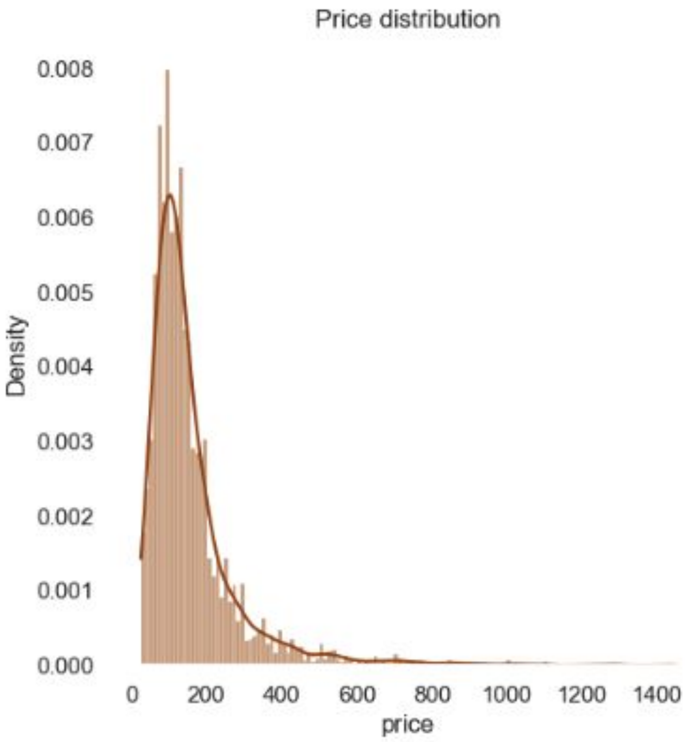


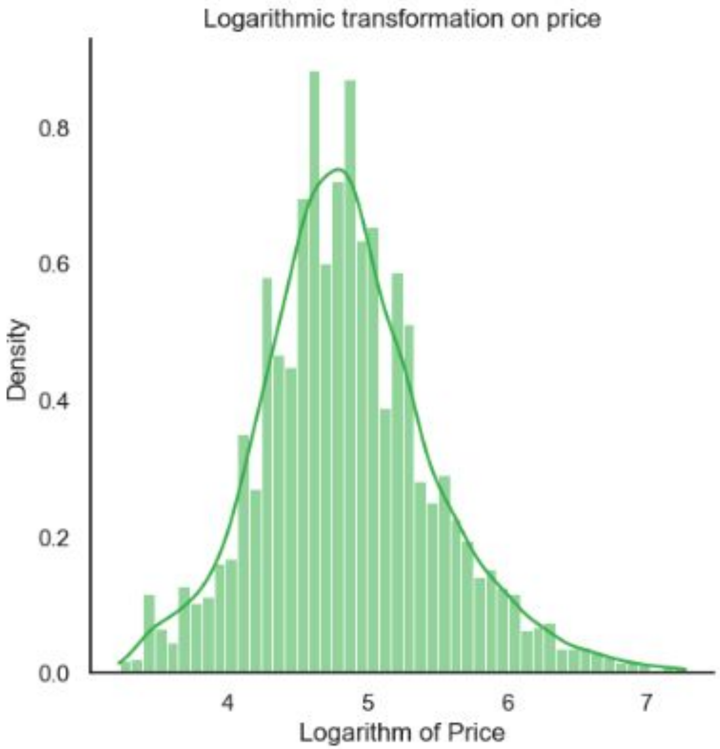
WalkScoreAPI





Price distribution





Modeling

Initial regression models Target = Price	**Training RMSE**	**Testing RMSE**
Baseline	134	134
Neural Network	83	88
Gradient Boost	43	89
Random Forest	51	90
Extra Trees	69	93
XGBoost	28	92
KNeighbors	84	105
Decision Tree	86	106
Bagged XG Boost	44	81
Bagged Gradient Boost	76	70
Bagged Decision Tree	54	89

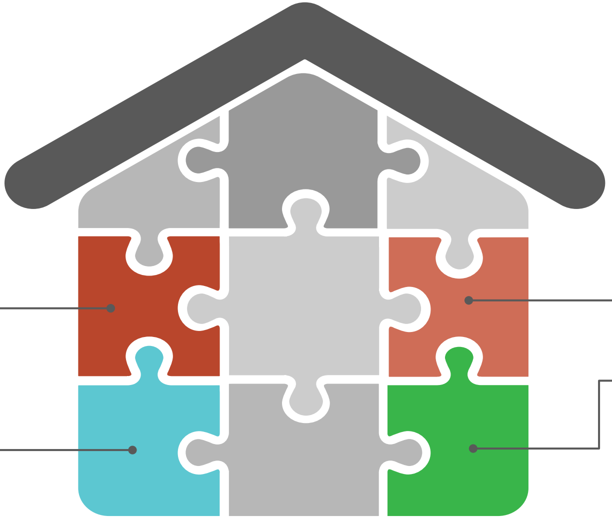
- I ran a few models using a logarithmic transformer on the target. This yielded similar, but slightly higher scores.

I ran a bunch of initial regression models to model the price of a listing.
I then took the top scoring highest scoring models (lowest RMSE) and did further model refinement (Grid Searching over parameters, adding neuron layers...) to find the best model with the best parameter.

Top performers	**Testing RMSE**
Bagged XG Boost	54.0
Bagged Gradient Boost	57.6
Bagged Extra Trees	86.3
Neural Network	87.1

Best Model: Ensemble Bagging Regressor with XGBoost Regressor as base_estimator

Standard Scaled: Numeric features
One Hot Encoded: Categorical features
Grid searched over a pipeline to find the best parameters:



Regularization: L2 (Ridge)

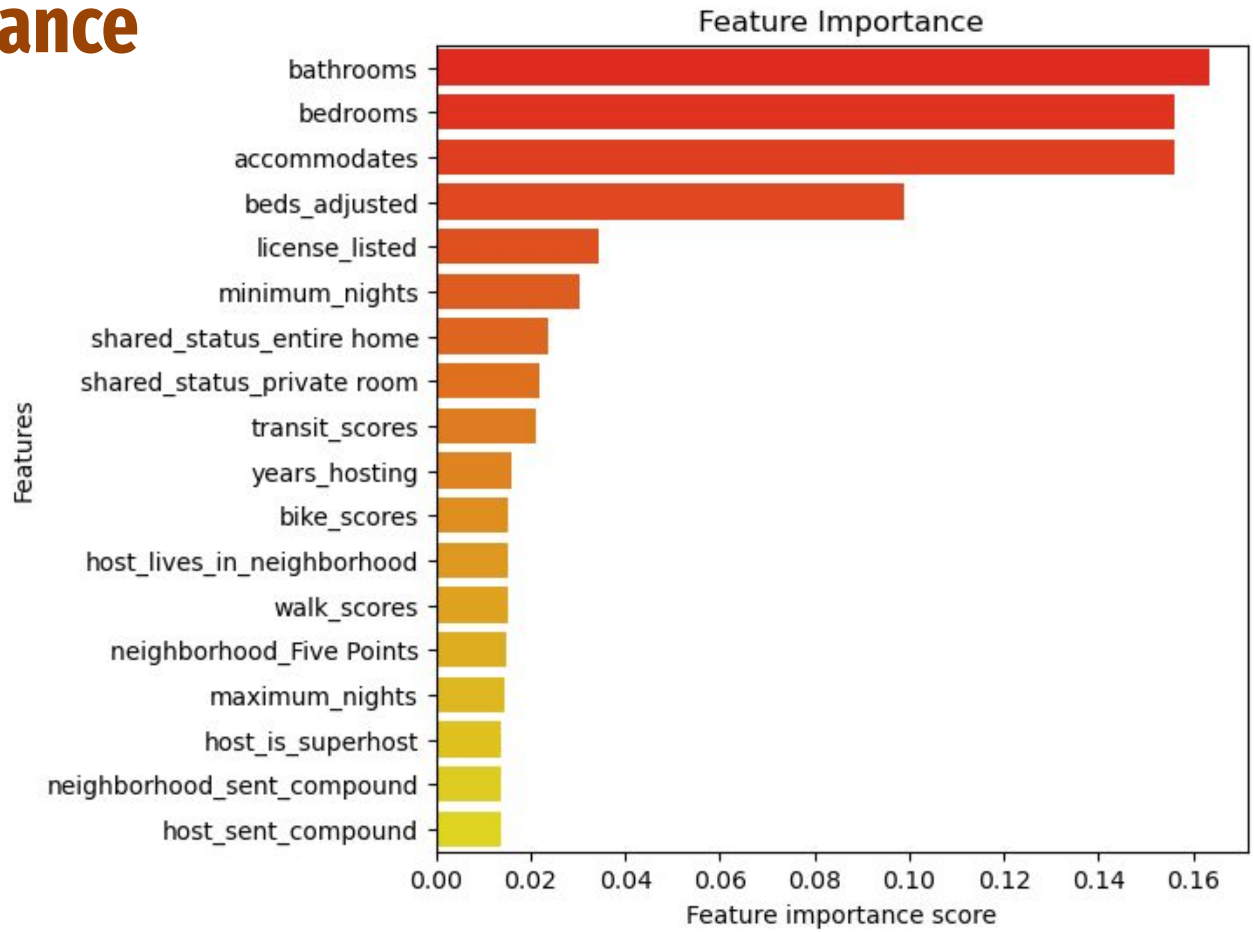
n_estimators: 30

Learning Rate: 0.01

Max Depth: 4

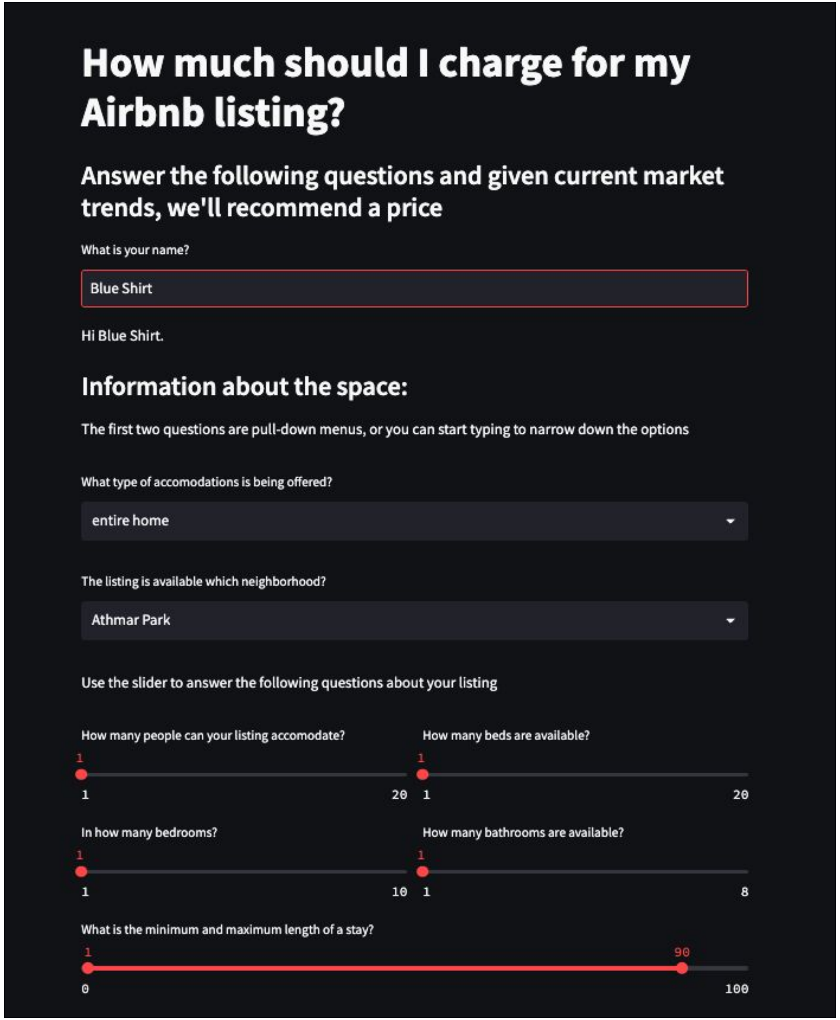
Reduced the baseline RMSE from 133 to 54, a 60% improvement.

Feature Importance



Streamlit

Using this model, I made an app where users can put in information about their listing, upload their descriptions… and the app will make a recommendation on a listing price. At the same time, it will provide some information about other rentals in their neighborhood

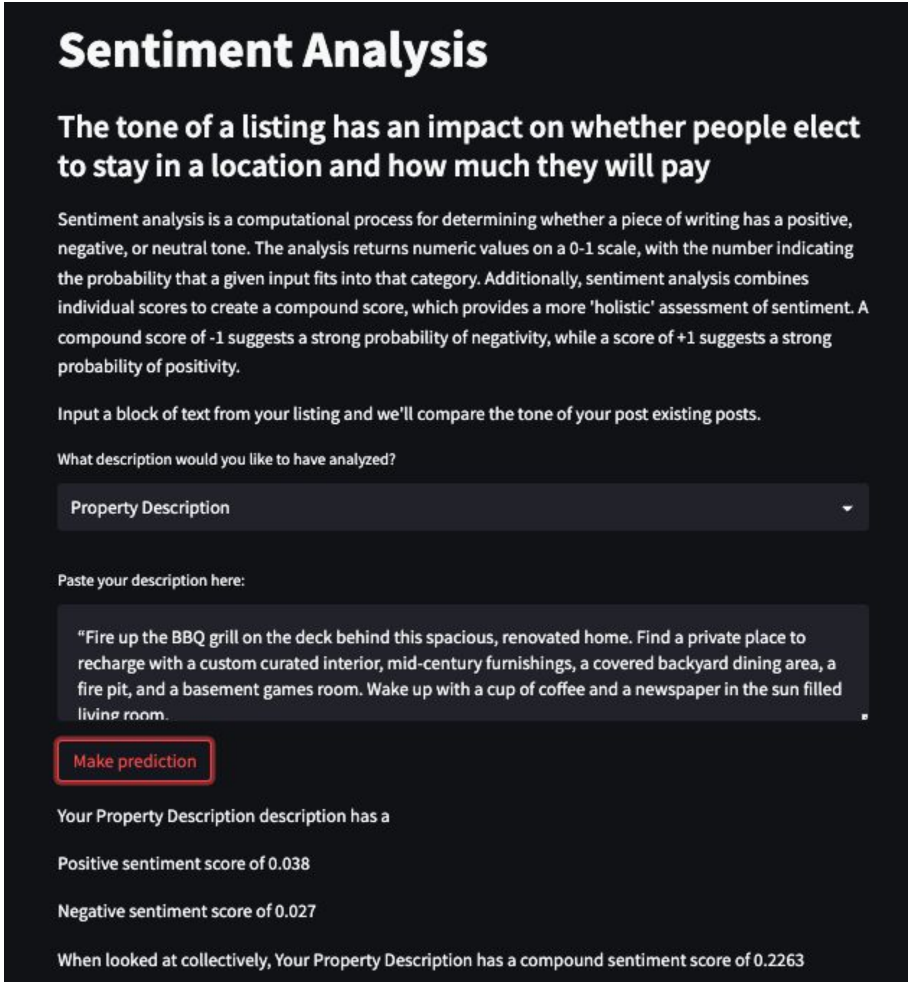


Streamlit

While I was in the process of making a Streamlit app I also decided to incorporate

a) Review Analysis/Comparison

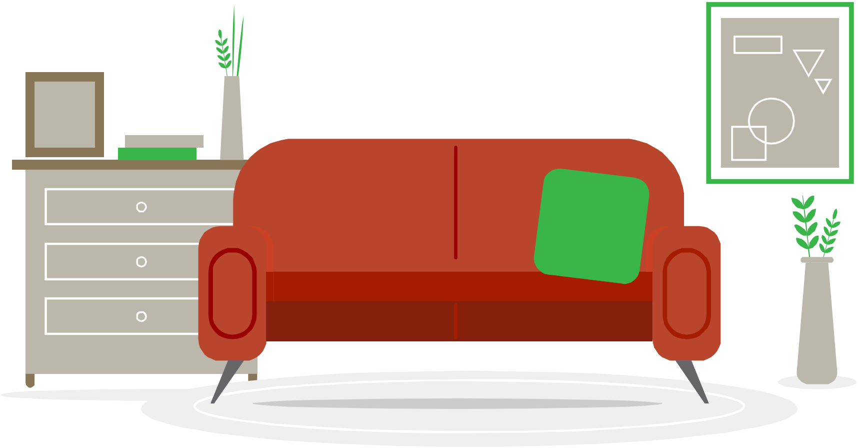
b) Sentiment Analysis/Comparison



Conclusion

My model provides a starting point for Airbnb listers to get a better sense of how they should price their property. With more time, it could be improved.

The Streamlit app is a fully functioning platform.



Next Steps

Find occupancy data

Incorporate a time element, only considering listings that have been rented recently

Weekends and Holidays

Not every time is equal, would need to gather fluctuations in prices.

Amenities

Given how the data is (not) organized
I didn't have time to sort through that

Recommender System

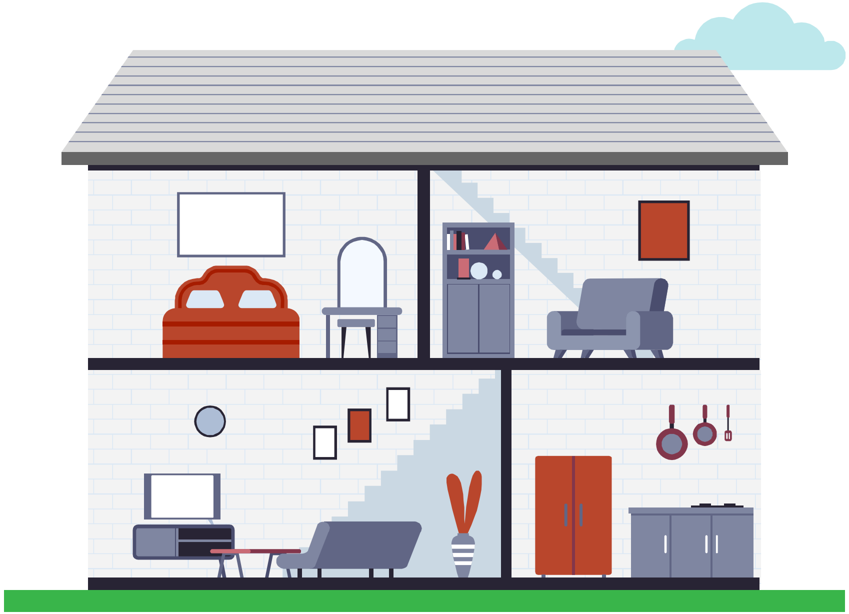
Utilize an unsupervised learning model to cluster houses for user/lister comparison.

Other Markets

Extend to model other metro areas.

Thank you for your time.

Any Questions?



Thanks to:
GA Instructors: Katie Sylvia and Tim Book
GA Classmates
Slidesgo

chatGPT