Project 2 Summary:
Putting a Price on ARTistic Experiences

Project Design

Using a linear regression model, my initial goal was to find out what current art class instructors are doing to attract customers, and if any of these factors are impacting their success—as measured by ratings. After web scraping Airbnb Experience listings and creating my initial model (minimum viable product), I found that only 2/3 of my sample had a rating and, of these, nearly 70% were 5.0 stars. While I may have had options (e.g., to perform an ordinal logistic regression or Heckman correction), I chose to adjust my target variable to predict price. Insights gained from my model could be used to help small local businesses—like my aunt's flower shop—build and price art classes.

I used Selenium to web scrape Airbnb Experiences: first, I scraped the search page using the filter "craft" and "classes" to find at least 1000 listings. I then collected as much information as possible from that page and their URLs so I could match them with their individual listings. After programming my code to find and pull text from as many unique page layouts as possible, I used Selenium to automate opening and scraping the 1000 individual pages.

After merging my "Start Page" and "Individual Pages" dataframes, cleaning (handling nulls, duplicates, etc.), and performing some EDA with Pandas, I started modeling with Scikit-Learn.

Tools

- Python
  - Data Cleaning & Analysis: Pandas, NumPy, Regular Expressions; Scikit-Learn, StatsModels
  - Data Visualization: Matplotlib, Seaborn
- Web Scraping
  - Selenium

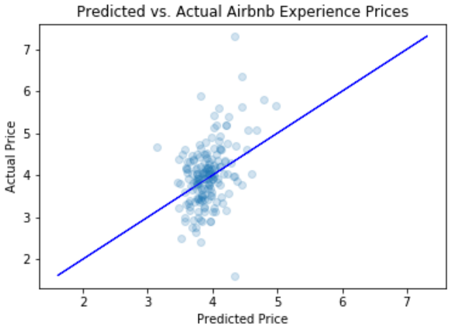
Data

The features I included in my model were:

Variable	Datatype	Range	Description
Language(s)	Int, discrete	1-6	Number of languages offered
Duration of experience	Float	1-14	Hours per experience
Group size	Int, discrete	1-30	Maximum number of people per experience
Includes Drinks	Int, discrete	0, 1 (N/Y)	If the experience includes drinks
Includes Food	Int, discrete	0, 1 (N/Y)	If the experience includes food
Host photos	Int, discrete	1, 2, 5	Number of photos on the listing. Note: #2 was often a sliding video/image that could contain 2-5 photos
Number of tags	Int, discrete	0-4	Number of tags associated with the listing
Title character count	Int, discrete	6-90	Number of characters in the title of the listing; excludes spaces
Title word length	Int, discrete	1-20	Number of words in the title of the listing
Average word length	Float	2.4-31.0	Average number of characters per word in the title of the listing

Before I started modeling, I split my data using a train-test split, so that I could test my model for generalizability (after optimizations and model selection). I then further split my training data into 5 k-folds for cross-validation. For modeling, I began by using Scikit-Learn's linear regression object—which is an Ordinary Least Squares calculation. After visualizing the distributions using Seaborn, I recognized a skewed distribution with price. I log-transformed price—which led to the biggest improvement in my r^2. I performed log transformations for some of the independent variables to normalize their distributions. I compared my OLS results to that of Statsmodels—which showed similar r^2 and provided some additional information, such as adjusted-r^2 and p-values.

Then I constructed a pipeline to test different options: polynomial feature transformation (using degree = 2 or 3), standardscaler, and regularization (lassoCV or ridgeCV). Pipelines are really helpful for streamlining different functions—since the output of one function will be used as the input in the next function. Feature transformations need to go first: Polynomial feature transformations can help increase r^2 by introducing more variable terms for the dependent variables; the result is still a linear regression, however, because there is still a linear relationship between the coefficients and the dependent variable. Polynomial transformations often leads to overfitting, and so regularization is necessary to reduce overfitting. Lasso and Ridge regularizations require preprocessing with standardscaler to center the distributions and standardize the variables. The model that produced the best results for my data was: PolynomialRegression(degree = 2), StandardScaler, and RidgeCV—with an r^2 of 0.150.



Above is the predicted vs. actual price (in terms of log of price). The narrow range of the predicted price and the fact that the predicted prices do not fit along the line of perfect fit (actual price vs. actual price) indicates that my model is underfitting. Ways to improve underfitting include: feature transformations, adding features, adding data, and addressing interactions (if applicable). The log and polynomial feature transformations did improve my model to an extent. Time constraints and limited numerical data on Airbnb Experience listings hindered my ability to collect more data and explore interactions. Additionally, the economics of pricing is rather complicated, and many factors in pricing (e.g., cost of supplies, labor costs; consumer willingness to pay) is very limited—in terms of public/data accessibility—and industry-specific. However, in my additional research of Airbnb Experiences, I found that may have been biases in my dataset that may make it less generalizable to the entire population anyway: selection bias and survival bias.

What I Would Do Differently & Future Work

More efficient web scraping, by automating challenges I addressed manually while scraping Airbnb Experiences:

- The infinite scroll feature (for the full page of search results, only 18 listings will appear). While there are ways to automate scrolling with Selenium, I ended up manually scrolling the page to load 1000 listings.
- While Airbnb Experiences is fairly new (established in 2016), some listings have very different layouts—which I overcame by using many if/else statements.
    - Of note: There is no way of sorting the page of listings and each time the same page is loaded, sometimes the listings change in order, content, and listing information. This introduces challenges in retrieving the same sample, sorting to find different listings, and sometimes the features I was pulling. The filter options were also limited to the following: dates, number of guests, price ($1-$100+), time of day, and language offered.
- Airbnb Experiences often omits web elements for features that do not apply to a given listing. For example, if there is no "food" included in the experience, there may not be a web element corresponding to that feature at all—which can cause my code to error out. I used if/else statements to resolve this issue as well.

Project approach:

- Research more before I select my project
    - Look for potential biases in the data

-
    - Look for examples of how more experienced subject matter experts would approach the problem (e.g., economists for pricing models)
    - Find a dataset with more features, and ideally more diverse datatypes (e.g., more continuous data features to predict a continuous dependent variable)
    - Approach my research with potential actionable takeaways for my end-user
    - Have a back-up project plan
- Organized my files better throughout the project
    - I found myself jumping around my notebook and creating cells/going out of order, which caused me to make some misinterpretations and unnecessary troubleshooting

If I had more time, I would:

- Try to better understand my coefficients after performing log transformations, polynomial regression, standard scaler, and regularizations. The feature engineering and regularizations drastically changed my coefficients (sometimes my coefficients were alternating between negative and positive signs)—which made interpretability of my model very challenging. Without this interpretability, I was not able to make actionable recommendations based on my model for my end business user.
    - Some options that I would look into are: Lime, Shapley
- Search for interactions:
    - Food-drink interactions? Maybe individually they do not have a significant effect, but together they may have an effect.
    - Tags and host photos interactions? Maybe hosts who invest more time in advertising through their pages have higher prices because their business costs are higher.
    - Duration of experience and food/drink interactions? Maybe longer experiences offer food and drinks, while shorter experiences don't.
- Explore types of drinks:
    - While not specified in all of the listings, maybe experiences that include alcohol have a higher price due to the higher costs.
- Further filter through types/categories of experiences:
    - Maybe some features had more impact in certain types of classes (e.g., paint nights may include drinks while workshop activities may not call for food or drinks)
- Dive deeper into actual languages offered at each experience:
    - I can't think of how it may affect my model, but it may be nice to know which languages are popular in a given location.
- Add more features:
    - Location: Cost of living can be a factor into businesses' costs. Unfortunately, Airbnb Experiences listings only contain the city name, experiences are not offered in very many cities, listings per city is not very high--especially in the category of "crafts" + "classes".
    - Using NLP of the listings:
        - To better understand the differences across experiences, and possibly find other features to include in my model (e.g., actual equipment or materials involved, complexity of the activit(ies), etc.)
        - To get a better sense of ratings, I could turn from (or combine) rating values with guest sentiment through the qualitative comments of their reviews. Or to collect more information about the host's costs: perhaps I can get more information from the reviews that give more details about the experience themselves (e.g., number of hosts (teacher(s), staff) per session, location details (size of space; studio vs. residential, downtown vs. suburban), materials used)
- Find and use a less biased dataset:
    - Since Airbnb Experiences has an application process for hosts, the quality of the experiences is biased—making my sample not random or representative of the entire population. There is also a high likelihood that survival bias is present—hosts who do not perform well may stop offering their experience. While this may not have caused my model to underperform since it affected both my training and test data equally, using a sample that is less prone to bias may have more generalizability to all art classes.