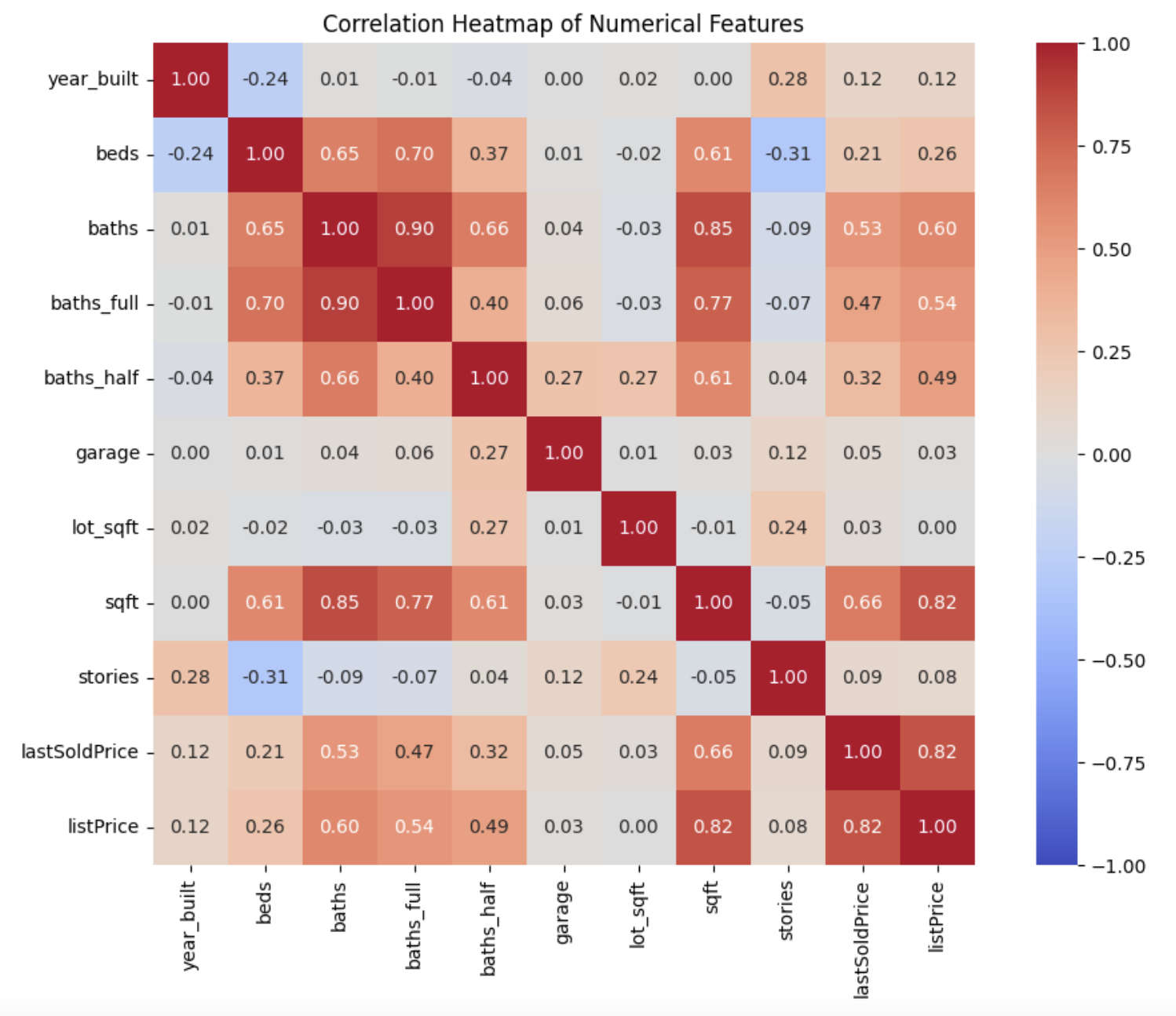
**Data Bootcamp Final Write Up**

**Introduction**

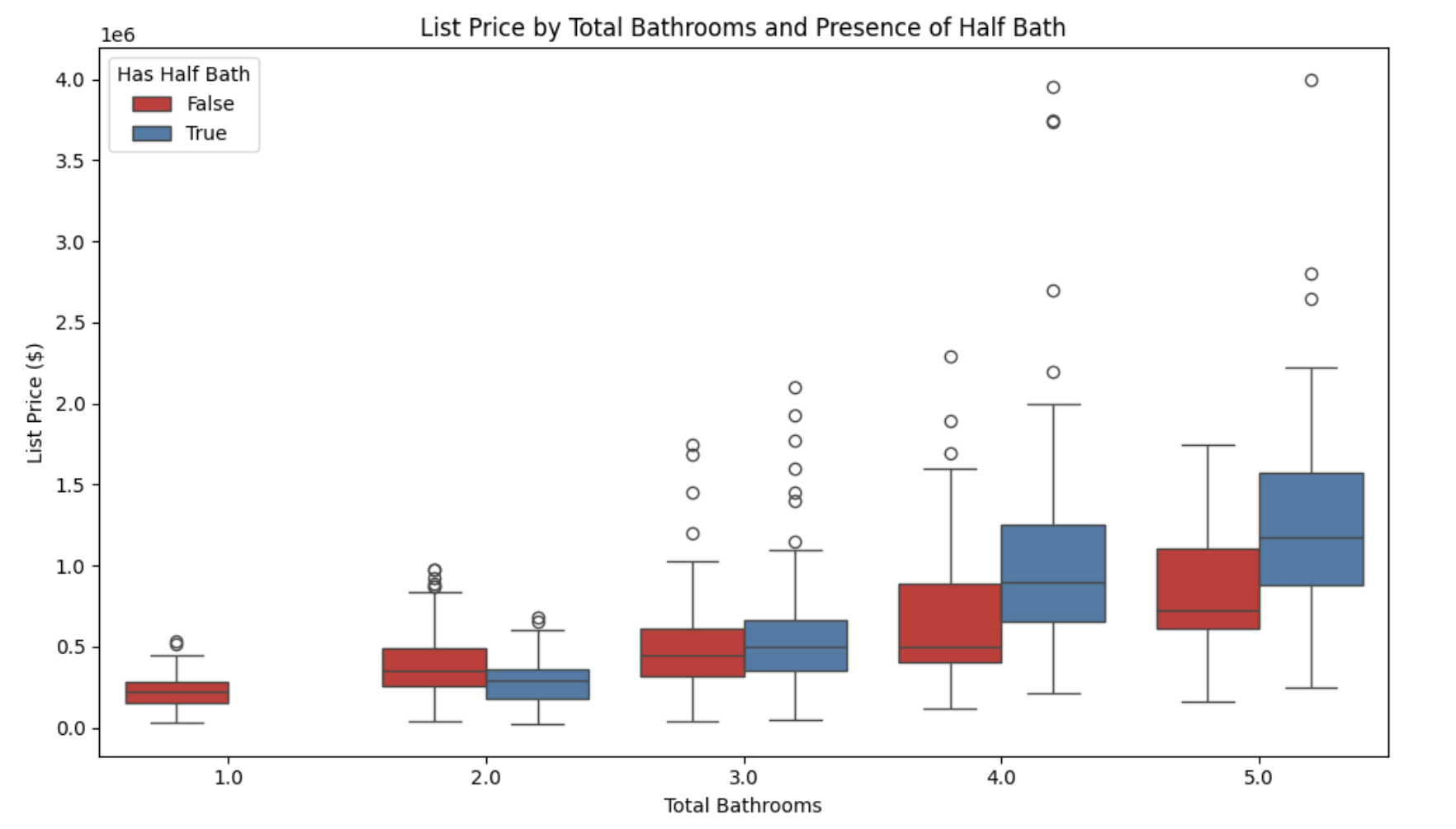
My project uses the Kaggle dataset titled ‘Real Estate Data Chicago 2024’. This dataset consists of real estate listings in Chicago. The data contains metrics that would typically be displayed on sites like Zillow or Redfin, for example: current listing price and most recent previous sale price. The data set also includes property features such as square footage and the number of bedrooms and bathrooms. My goal was to use this data to build models that return a rational reference price. When placing properties on the market, sellers could use this price as guide when setting their initial ask price. After exploring relationships between features and listing prices, I found that models using the number of baths, half baths, square footage, and last sale price produced the most accurate predictions. In the end, the second-degree polynomial model I constructed outperformed the others, including linear regression and KNN, with the lowest test error.

**Exploratory Data Analysis**

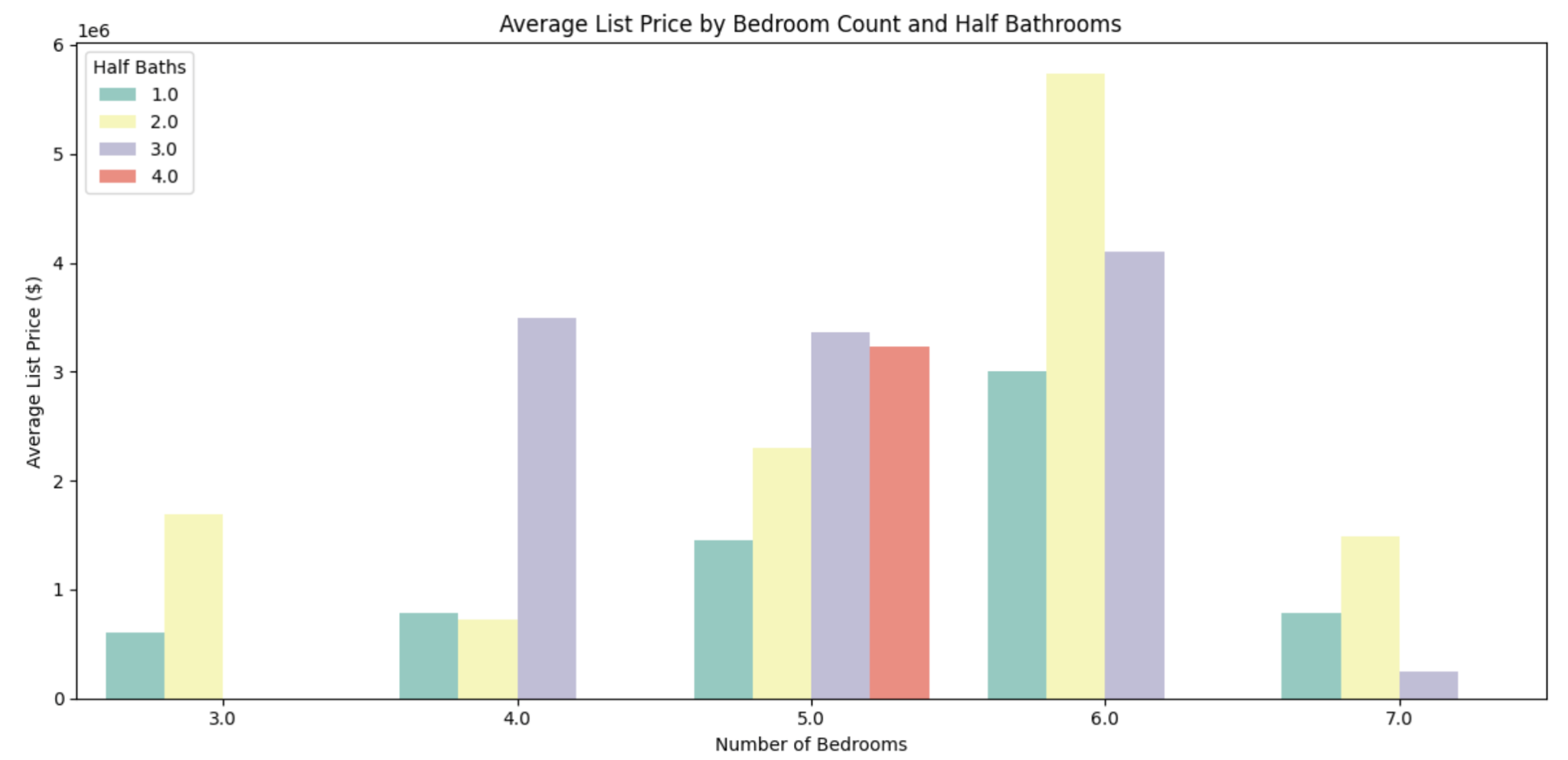
I began my analysis by constructing a heat map to explore the correlations between the numerical features in the dataset. For my project, I knew that I wanted to focus on how the features related to list price, as this was the target variable that I would be predicting for the modeling portion of the assignment.



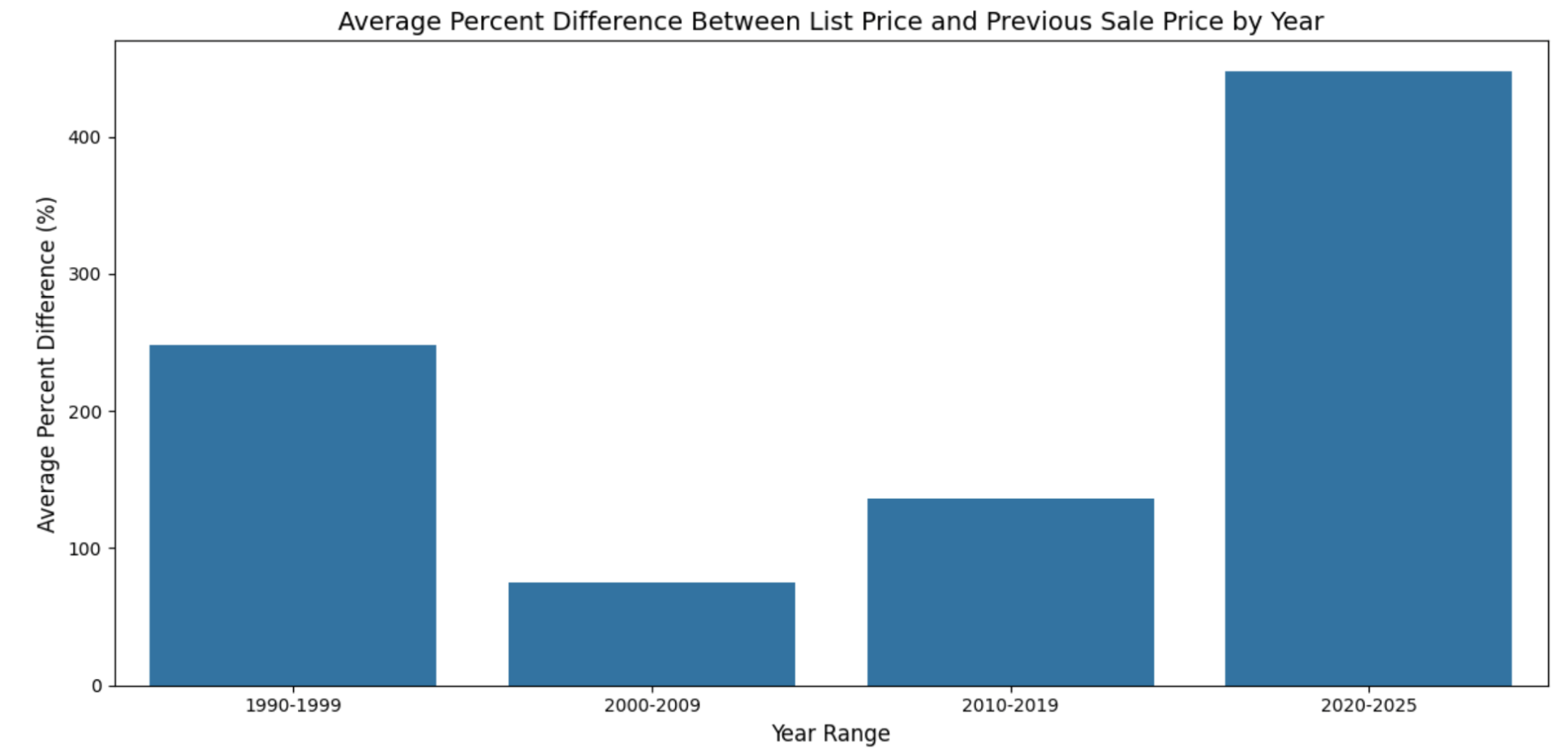
When observing the heatmap I found that the list price had relatively strong, positive correlations with beds, baths, number of full baths, number of half baths, square footage, and last sale price. In other words, as these features increase, list prices tend to increase as well. These findings guided the remainder of my EDA analysis, which included visualizing specific features—like half bathrooms—to better understand how they affect list price. Aside from last sale price and square footage, the column with the highest correlation with list price was the half bathroom column. I decided to explore what the impact of the presence of a half bathroom had on the list price of the property by plotting boxplots of the price for each number of bathrooms.



I used the presence (or absence) of a half-bathroom as the hue to explore the price differences. The plot shows that having a half-bathroom is associated with higher list prices only when a property has a total of three or more bathrooms. This suggests that half-bathrooms begin to add value once a home already has at least two full bathrooms. This conclusion makes sense—once basic needs are met, additional half baths can increase a property's appeal by providing extra amenities without taking up much space. The box plot analyzing the impact of the presence of half bathrooms on list price led me to another question: What is the ideal number of half bathrooms a property may have to maximize value? I chose to answer this question by creating a bar plot that displays the average prices for each number of bedrooms, using the number of half baths as the hue.

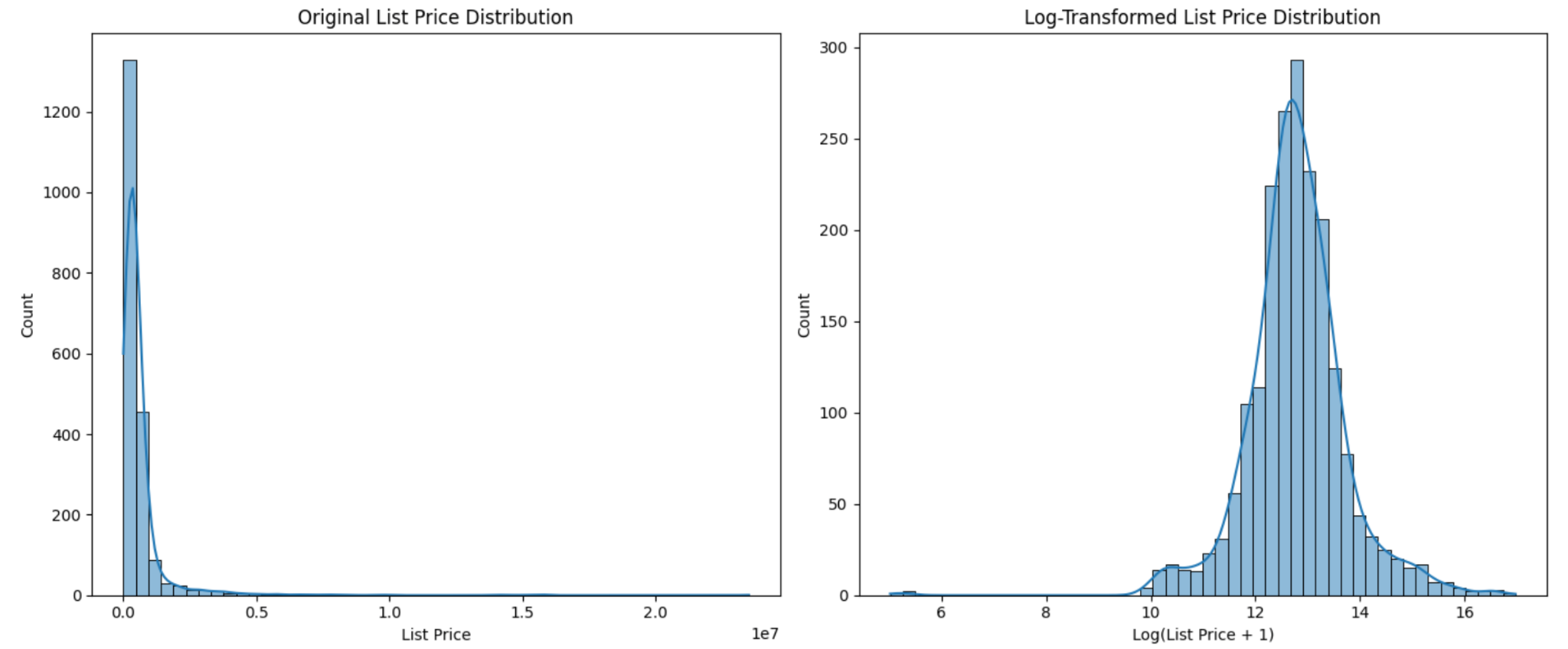


This visualization allowed me to explore what the ideal number of half baths would be, depending on the number of bedrooms. This visualization suggests that two to three half baths appear to be ideal for value maximization across different bedroom counts. We also observe that having too many half baths can diminish value. This is illustrated by the smaller average prices seen when moving from 3 to 4 half baths for a 5-bedroom property, or from 2 to 3 half baths for a 6-bedroom property. I found it interesting that the ideal number of half bathrooms for a 5-bedroom property was 3, while the ideal for a 6-bedroom property was 2. I'd imagine there are several factors contributing to this discrepancy, such as the size of the property, buyer preferences, or the way space is utilized in larger homes. The column most strongly correlated with the list price is the last sale price. To analyze how effective this feature might be in predicting current list price, I investigated whether the year of the last sale influenced how much of a premium is commanded today.



I was slightly surprised to find that properties with the most recent sales also showed the largest increase from previous sale price to current listing price. This suggests that owners, who bought more recently and are now looking to sell, expect to earn more on their initial purchase than those who bought longer ago. I imagine there could be a couple of reasons for this. One theory is that people who have held onto properties for shorter periods may be house flippers. These individuals might have purchased homes at lower prices, quickly renovated them, and are now re-listing them for a significant markup. Interestingly, homes that last sold in the 1990s and are back on the market show greater price increases than those last sold between 2000 and 2019. This suggests that while the "fix and flip" strategy may account for the sharp rise in list prices among more recently sold homes, the expected time correlated price appreciation trend still holds when looking over a broader time horizon.

For the final step in my EDA analysis, knowing that linear models assume a normally distributed target variable, I took a look at the distribution of list prices.



As one may observe in the histograms above, it is clear that using the log-transformed list price is optimal because it is far more normally distributed than the original distribution. Using the transformed variable will likely enable our models to predict more accurately.

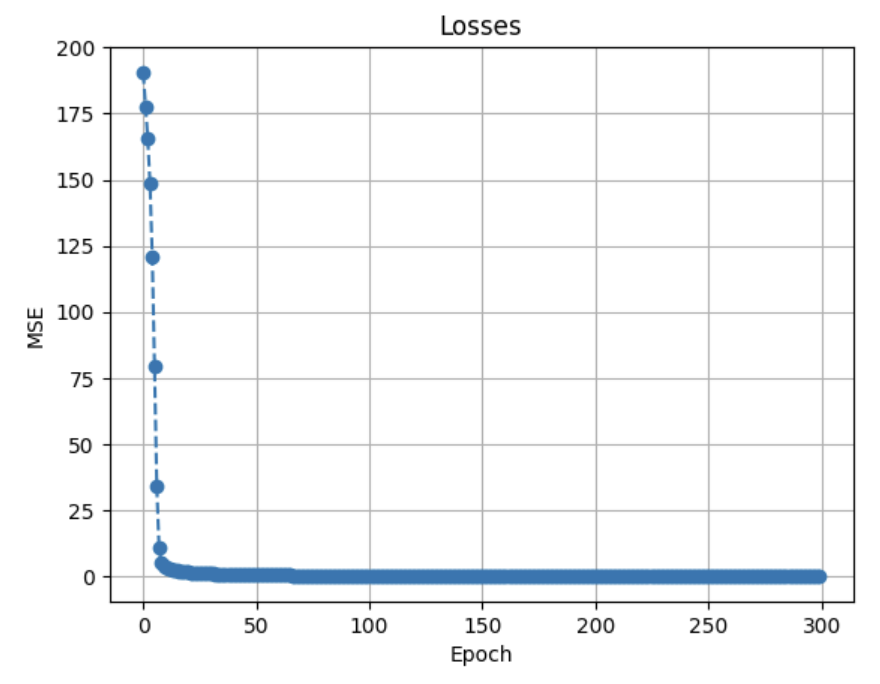
**Predictive Modeling**

After completing EDA, I tested several models to predict list price using the log-transformed list prices. The first model I built was a linear regression model. After experimenting with a combination of different features, I found that using 'baths', 'baths\_half', 'sqft', and 'lastSoldPrice' produced the best performing linear regression model. The MSE values produced represent squared errors in the predictions, which are in terms of the log of the price. With a train MSE of 0.189 and a test MSE of 0.233, one may assume that because the train MSE is not significantly lower than the test, the model is not badly overfit. A Test MSE of 0.233 implies that the mean squared error between the predicted log of the price and actual log of the price is about 0.233 on average. By taking the square root of this value, we get 0.483, which is the root mean squared error. This implies that the model's log of the price is off by about 0.483. Converting this back to price by taking exp(0.483) we learn that the model's predictions are off by 62%. For instance, when predicting the price of a $1,000,000 dollar home, the model may predict anywhere from 617,283 to 1,620,000. While this may seem like a large range, the linear regression model performs significantly better than the baseline (predicting the mean of the target variable for each prediction), whose predictions are off by 153% rather than 62%.

The next model I chose to create is a polynomial model. After experimenting with varying degrees of polynomials on the same features I used for my linear regression model, I was able to achieve MSE values that signified the polynomial model performs more optimally than the linear regression model. This model produces a train MSE of 0.1373 and a test MSE of 0.1464. This equates to a test RMSE of 0.3826 on the log of the price, an indication of a exp(0.3826) = 47% error in price predictions. This would mean that the polynomial model may predict a price of anywhere between $680,272 and $1,470,000 for a $1,000,000 house. While there is still a considerable level of inaccuracy in the range of possible predictions, the polynomial model appears to perform with over 10% more accuracy than the linear regression model, and significantly better than the baseline.

After creating a polynomial model, I constructed a knn model. I chose to use 9 neighbors for my model as this seemed to produce the most optimal results. In the end, this model ended up being my worst performing model, with a train MSE of 0.228, and 0.4356. Not only do these results signify more potential for overfitting compared to the others, but its accuracy is significantly worse, with a test MSE that is nearly .3 greater than that of the polynomial model and much closer to the baseline MSE.

Finally I decided to construct a neural network. After trying several different learning rates, I found that a learning rate of around .007 consistently performed the best.



The loss function of neural network loss starts close to 200 and decreases to about 0.138 after 300 epochs. The consistent decrease in loss indicates the model is effectively learning. The final training loss value of 0.138 performs on par with the best performing model I constructed, the polynomial model, which had a train MSE of 0.137.

**Conclusion and Next Steps**

My models performed significantly better than the baseline (which simply predicts the mean list price); however, to function as a practical tool for predicting optimal real estate listing prices it would need to be enhanced. The most obvious improvement would be to obtain a much larger data set. Also, the data set I used was essentially a snapshot of current market listings; thus, it did not directly include mis-pricing indicators. I would need to license an additional dataset of recent prior transactions to capture time-to-sale data and correlated final sale price data. Because the value of real estate is heavily influenced by precise location, including a dataset that allows me to easily group based on neighborhood would also serve to further improve accuracy. In future projects, I must also experiment with other models such as Random Forest, or more complex neural networks.