PREDICTIVE HOME PRICING ALGORITHM

Group 2

Caitlyn Epley
Arlene Wimbley
Denise Long
Zac Blankenship

OVERVIEW & GOALS

The aim of our project is to explore real estate data to predict home prices. In this project, we examine aspects of homes, such as square footage, number of bedrooms, etc. to predict the value of other homes.

Goal #1

Test multiple types of machine learning models to find an algorithm to accurately predict home prices

Goal #2

Determine which features are most important in determining home prices.

DATA COLLECTION

The original dataset consisted of 2,226,382 rows of data with 12 columns. After reducing the dataset, we were left with 1,360,347 rows and 10 columns. This cleaned dataset was then used for the machine learning algorithm.

Data Removed:

- 'brokered_by', 'street'
- Non-contiguous states
- All rows with null values

Data Used:

- 'status', 'price', 'bed', 'bath', 'acre_lot', 'city', 'state', 'zip', 'house_size', 'prev_sold_date'

- Resources:
    - USA Real Estate Dataset
    - Regions Dataset
    - Geocodes Dataset

DATA PREPARATION

DEPENDENCIES

- Pandas
- SQLAlchemy
- Matplotlib

PREPROCESSING STEPS

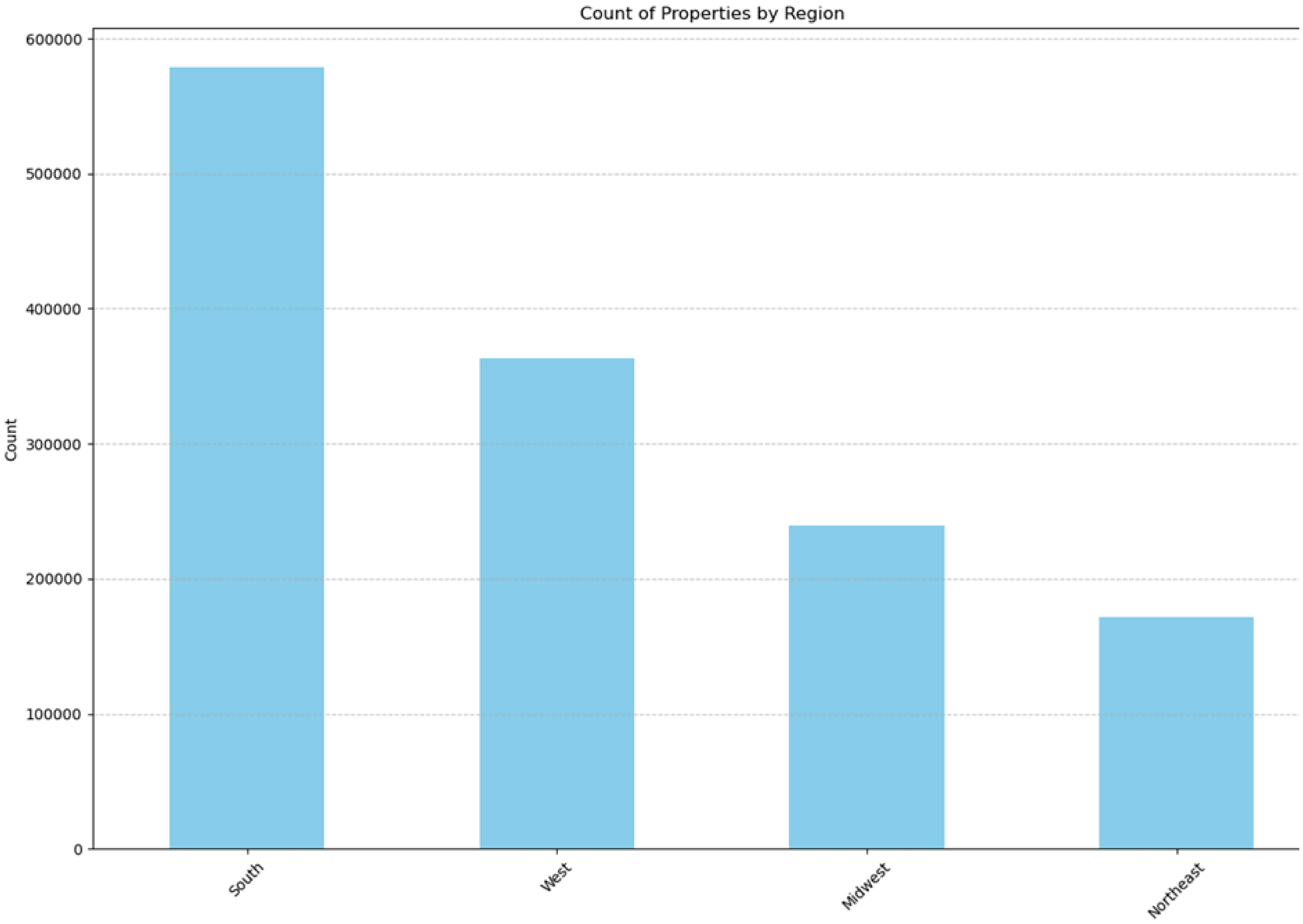
- Read csv and created data frame of the original data file
- Checked for duplicates – none found
- Removed rows with null values EXCEPT for prev_sold_date
- 61 percent of data was kept
- SQLAlchemy used to create SQLite database connect for use with machine learning

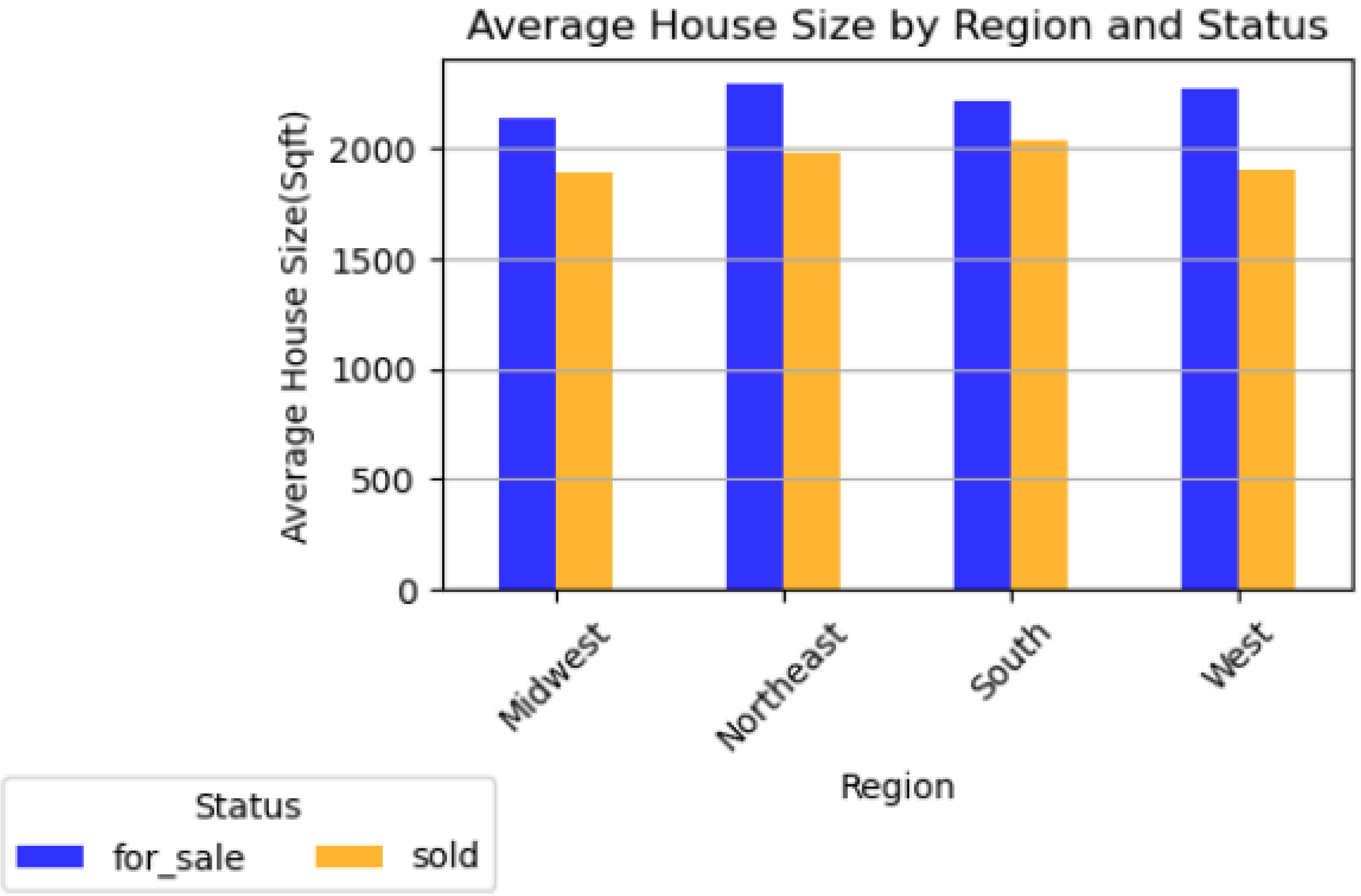
PostgreSQL

THE DATABASE

Relational Database

- Data was structured (columnar)
- Minor datatype changes were necessary
  - zip_code field changed from decimal to varchar
- Virtual Table created (view) adding region, division and longitude and latitude coordinates
- Queries created to answer questions

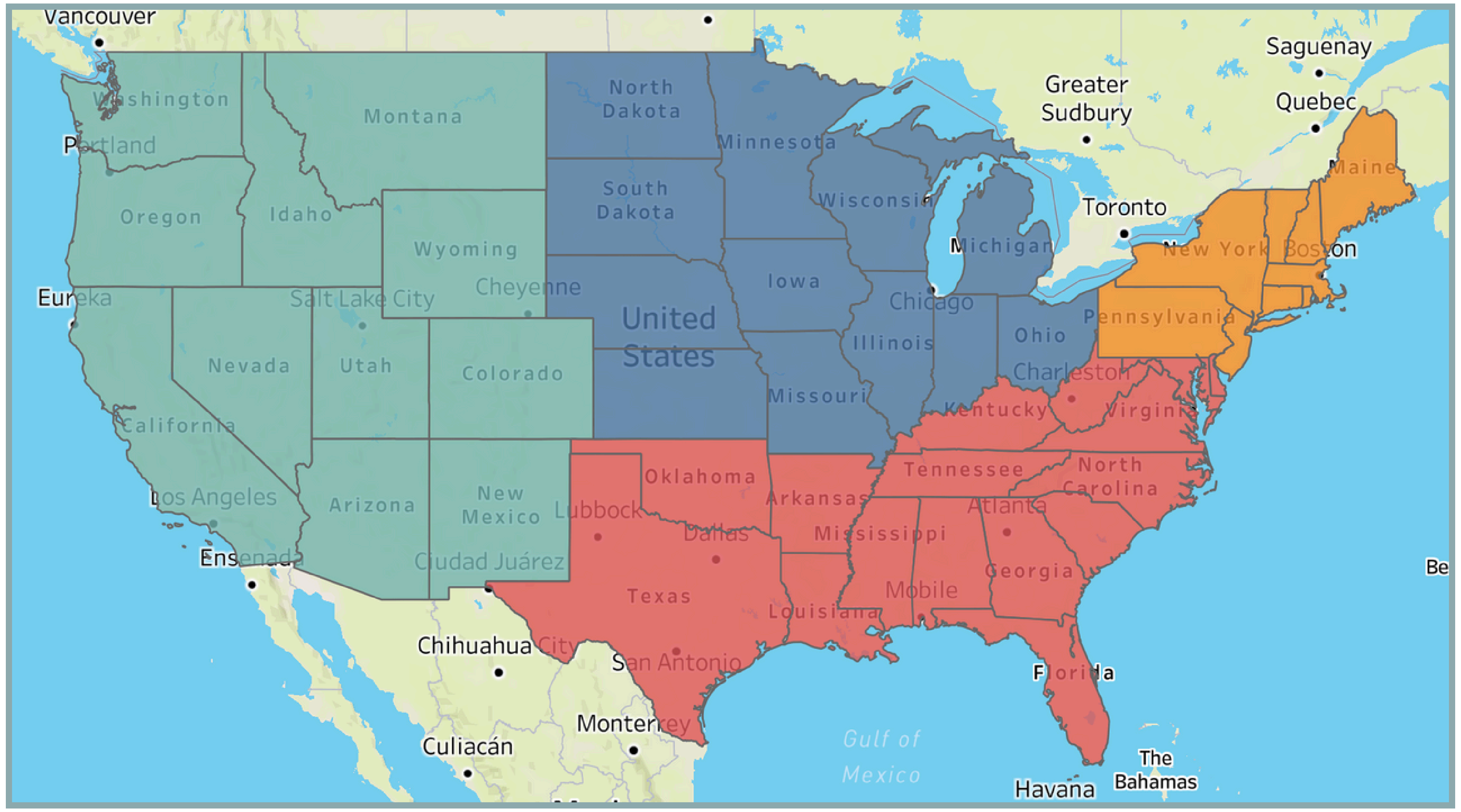

Count of Properties by Region
600000
500000
400000
300000
200000
100000
0
Count
South
West
Midwest
Northeast


Average House Size by Region and Status
Average House Size(Sqft)
2000
1500
1000
500
0
Midwest
Northeast
South
West
Region
Status
for_sale
sold

tableau®

EXPLORATORY DATA ANALYSIS

The below map highlights the regional differences in price per square foot. Gaining insight into these regional differences is essential for comprehending the broader housing market landscape.

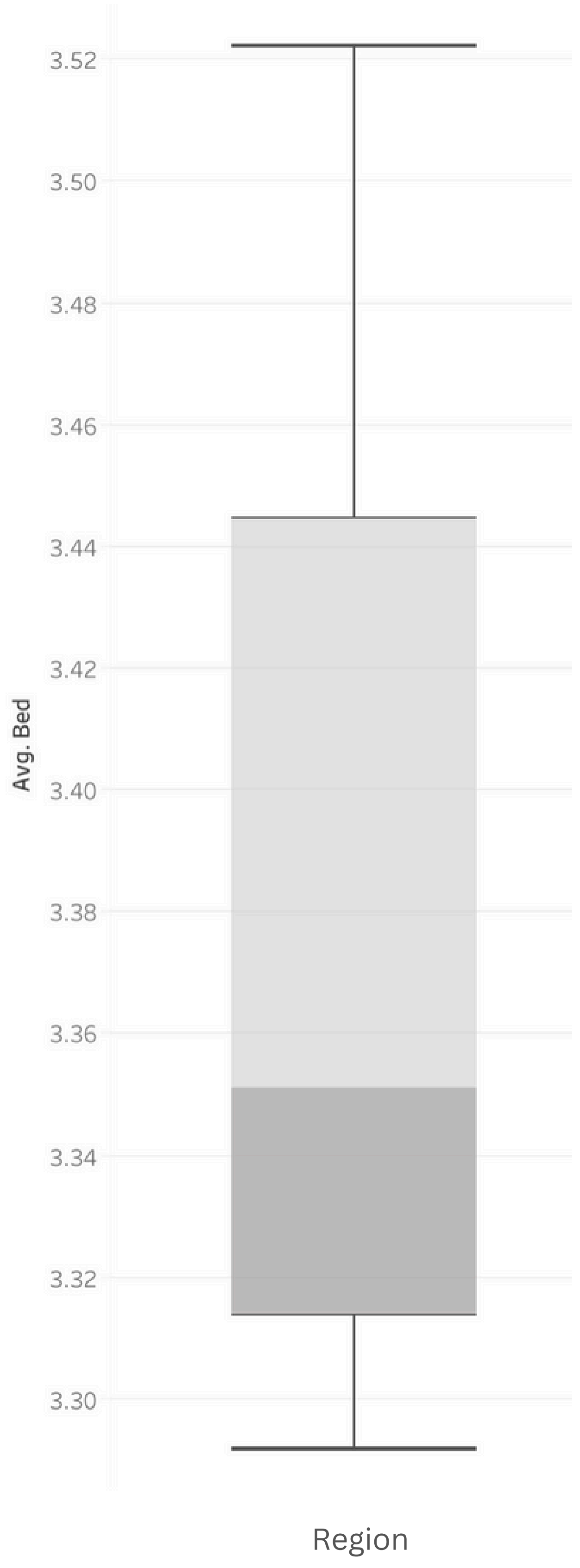


Midwest
Avg. Price: $304,345

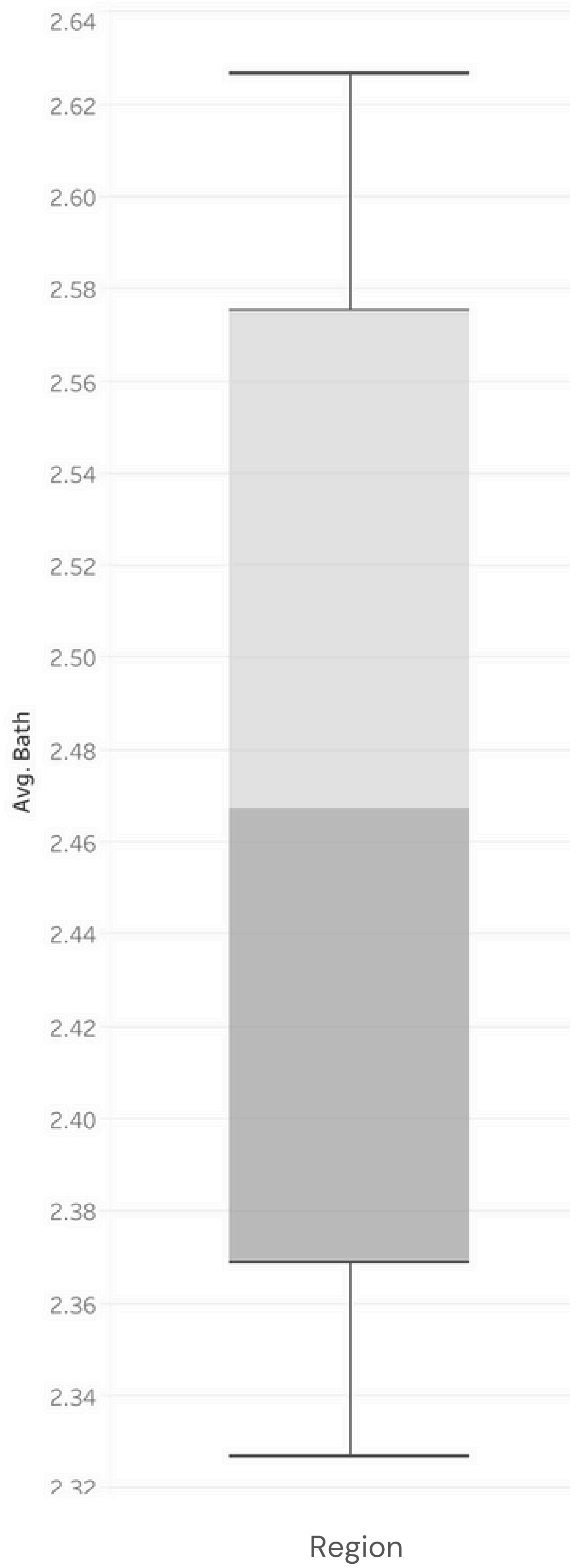
South
Avg. Pric: $479,362

Northeast
Avg. Price: $527,761

West
Avg. Price: $859,441

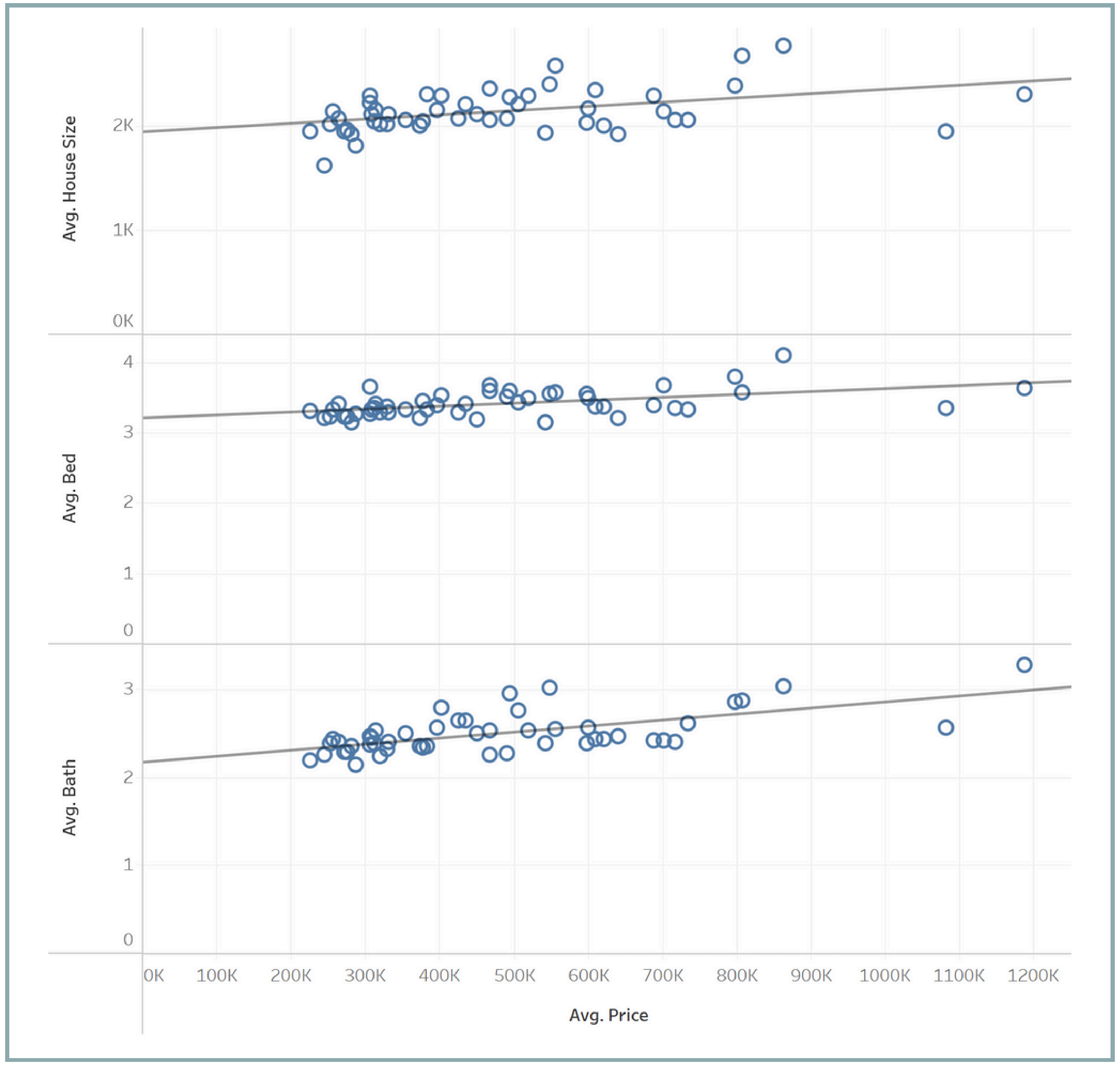


- **Avg. Number of Beds**
  - Fairly similar in all regions implying a consisten demand for 3-4 bedrooms homes across the US regions.



- **Avg. Number of Bathrooms**
  - Similar to beds, baths are similar across the board except for in the South where it’s possible for a regional preference for more bathrooms.

- **Avg. Price per Sqft**
  - This had the largest range of regional differences, indicating some regions may have unique external factors that have a higher affect on price.



- The linear regression indicates that a (weak) relationship exists does between average price and house size, number of bedrooms, and number of baths.

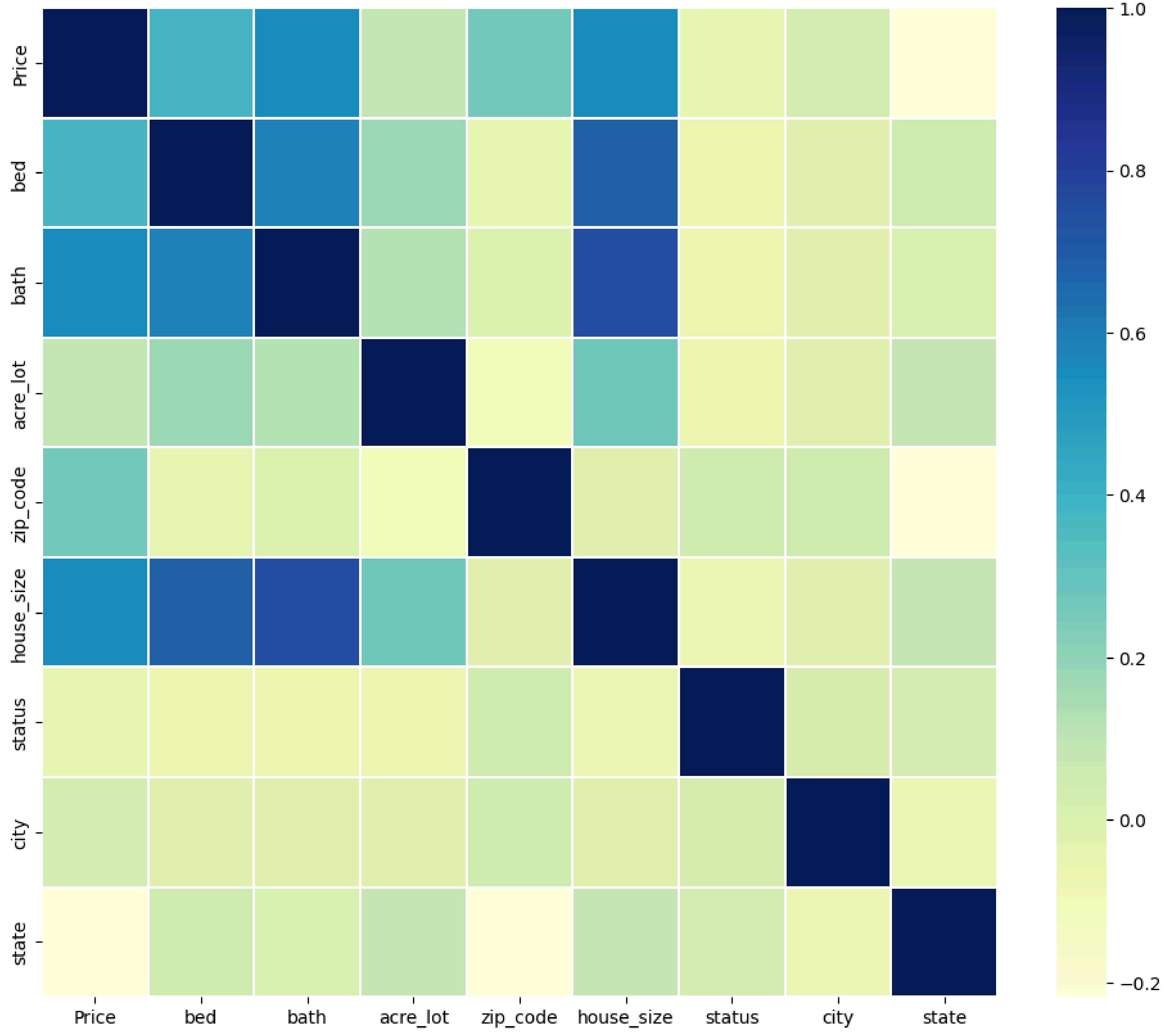
- Given the common knowledge that a larger house often costs more, it's very possible that a linear regression is not the best model for capturing the complexities of these relationships.

Machine Learning

TRAINING PROCESS

Correlation Matrix

The features that have the most impact on the price are house size (~0.55) and the number of bathrooms(~0.55). The features that have the least impact are state (~-0.21) and sold status (~-0.05)



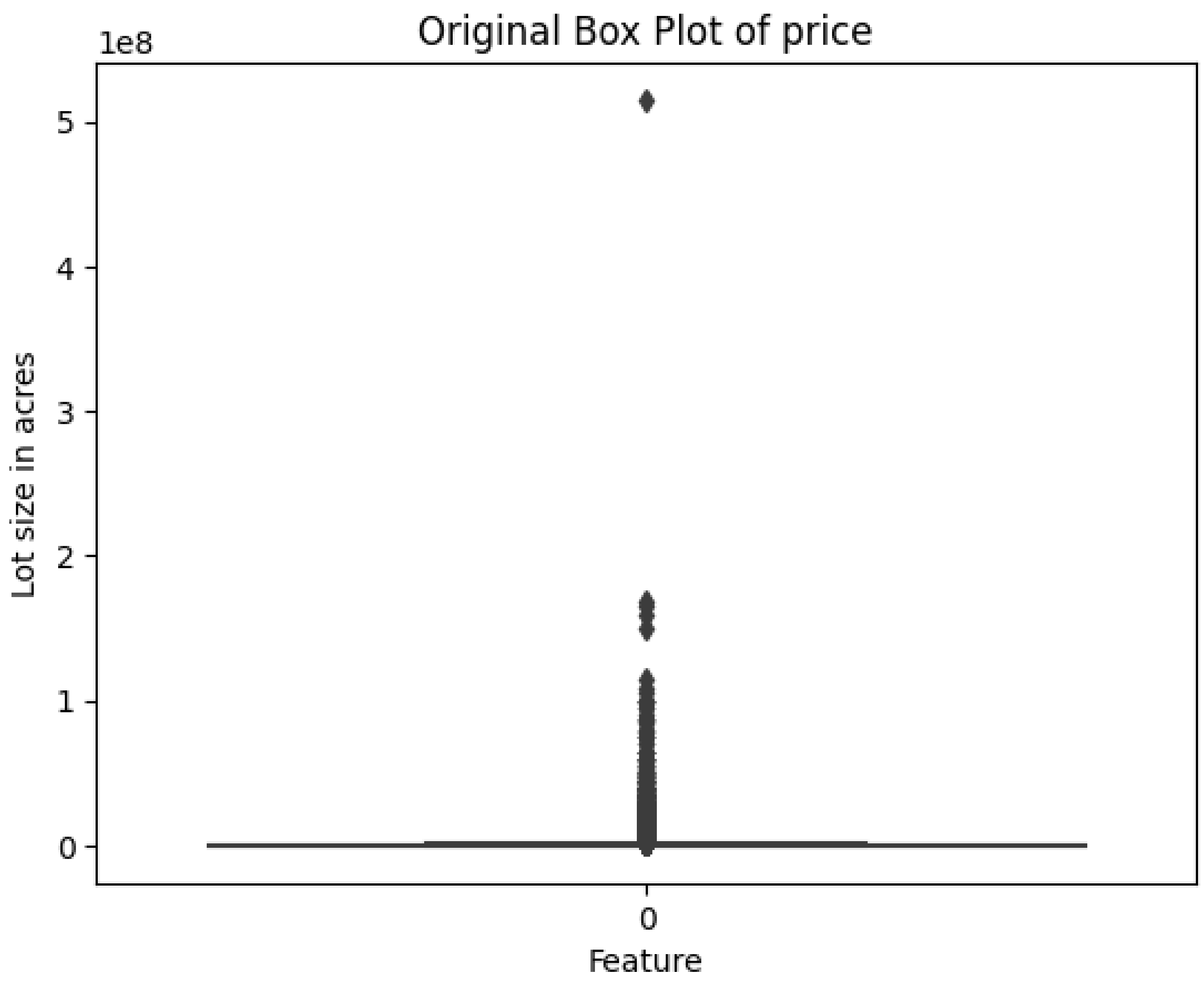
Removing Outliers

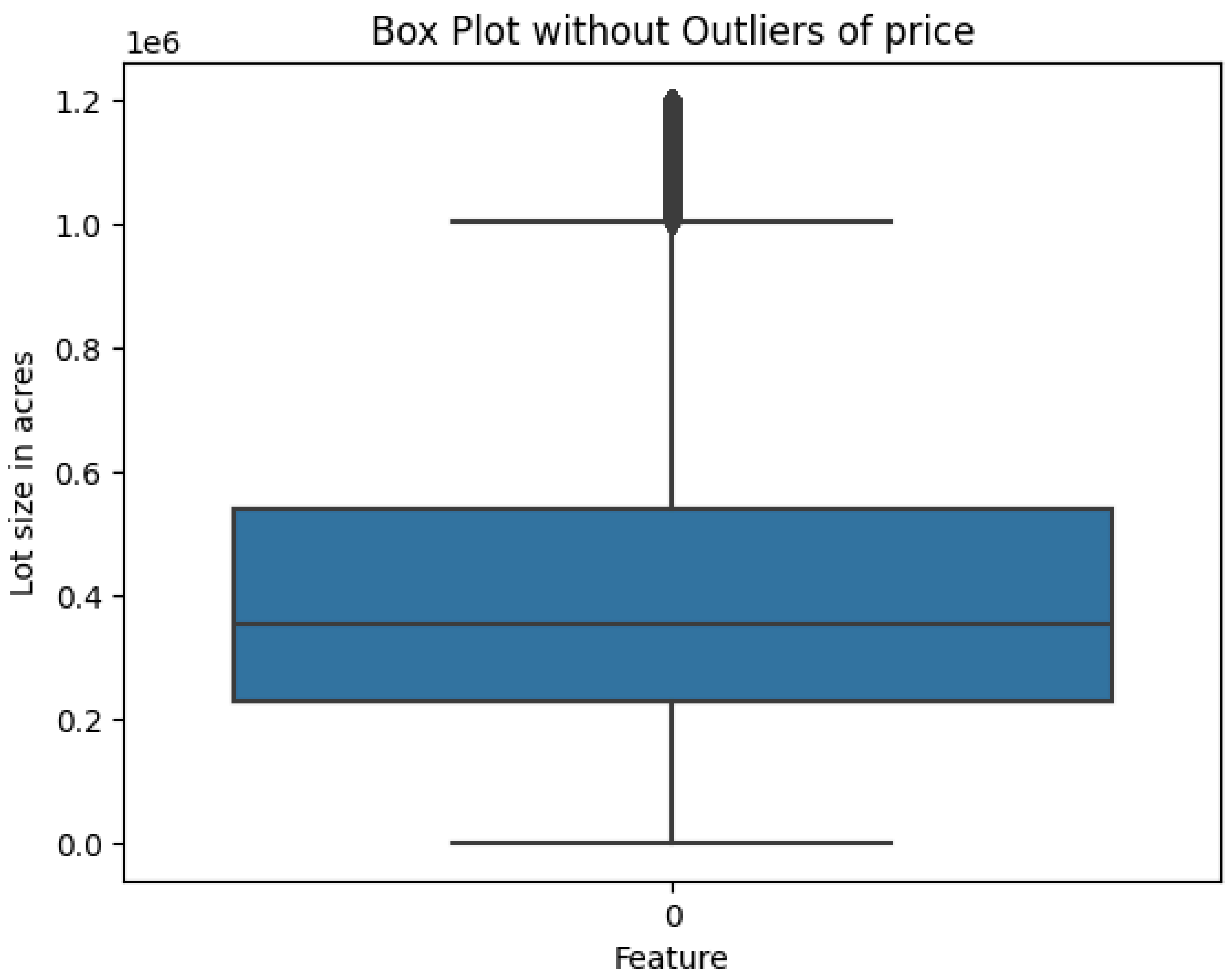
To improve the fit of the machine learning models, rows that contain extreme outliers need to be filtered out. Removing these reduced the dataset to 935,674 points. To remove outliers, box and whisker plots were used to show the spread of data and the outliers

Limits:

- Price < $1,200,000
- Beds < 6
- Baths < 6
- Lot size < 0.5 acres
- House size < 4,000 sqft

Before and After Removing Outliers

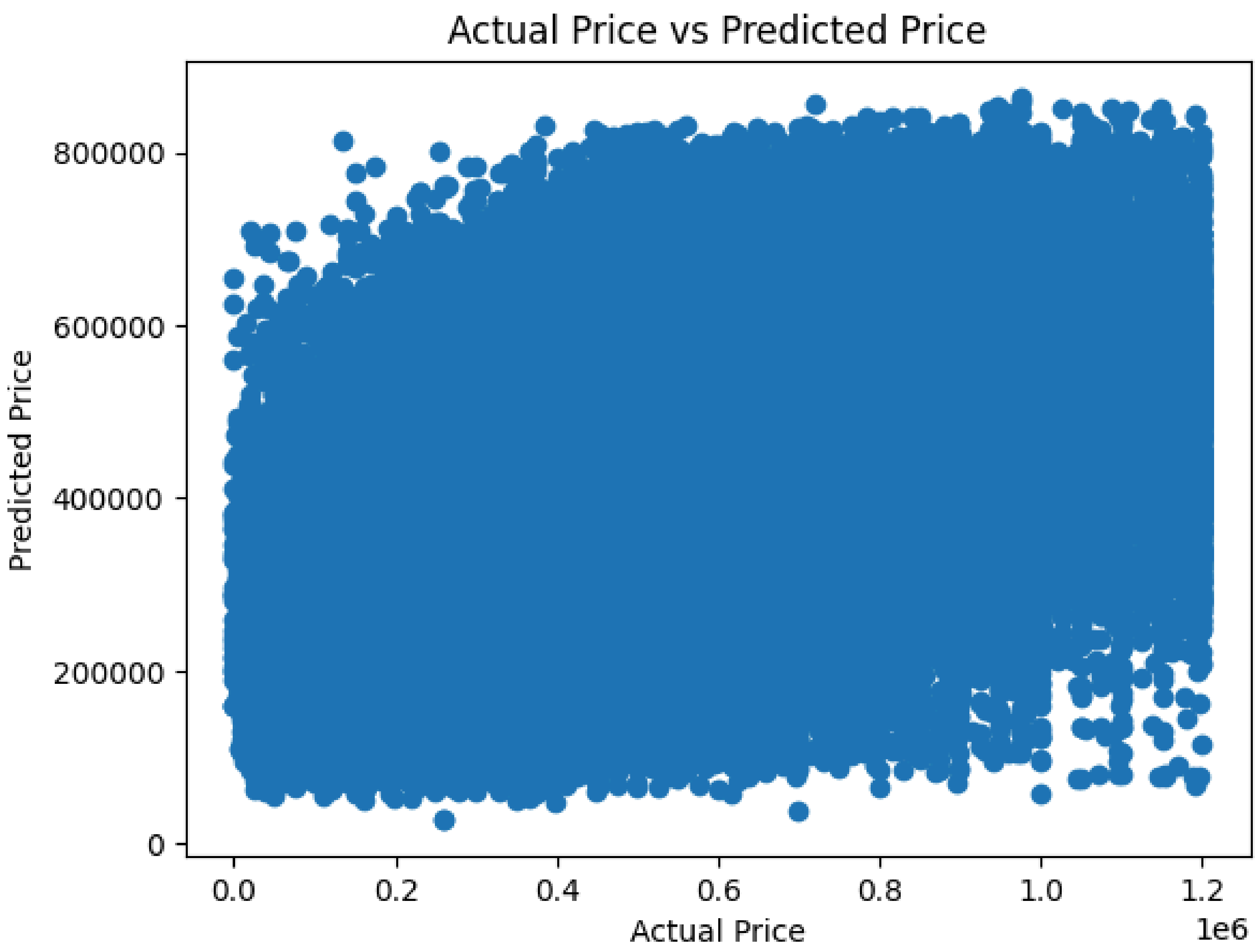




Multivariate Linear Regression

"Linear regression analysis is a set of statistical procedures designed to examine relationships between one or more independent variables (IV) and one dependent (i.e., outcome) variable (DV)"(Randolph & Myers, 2013).

R-Squared Result:
~0.298 on training data
~0.301 on testing data

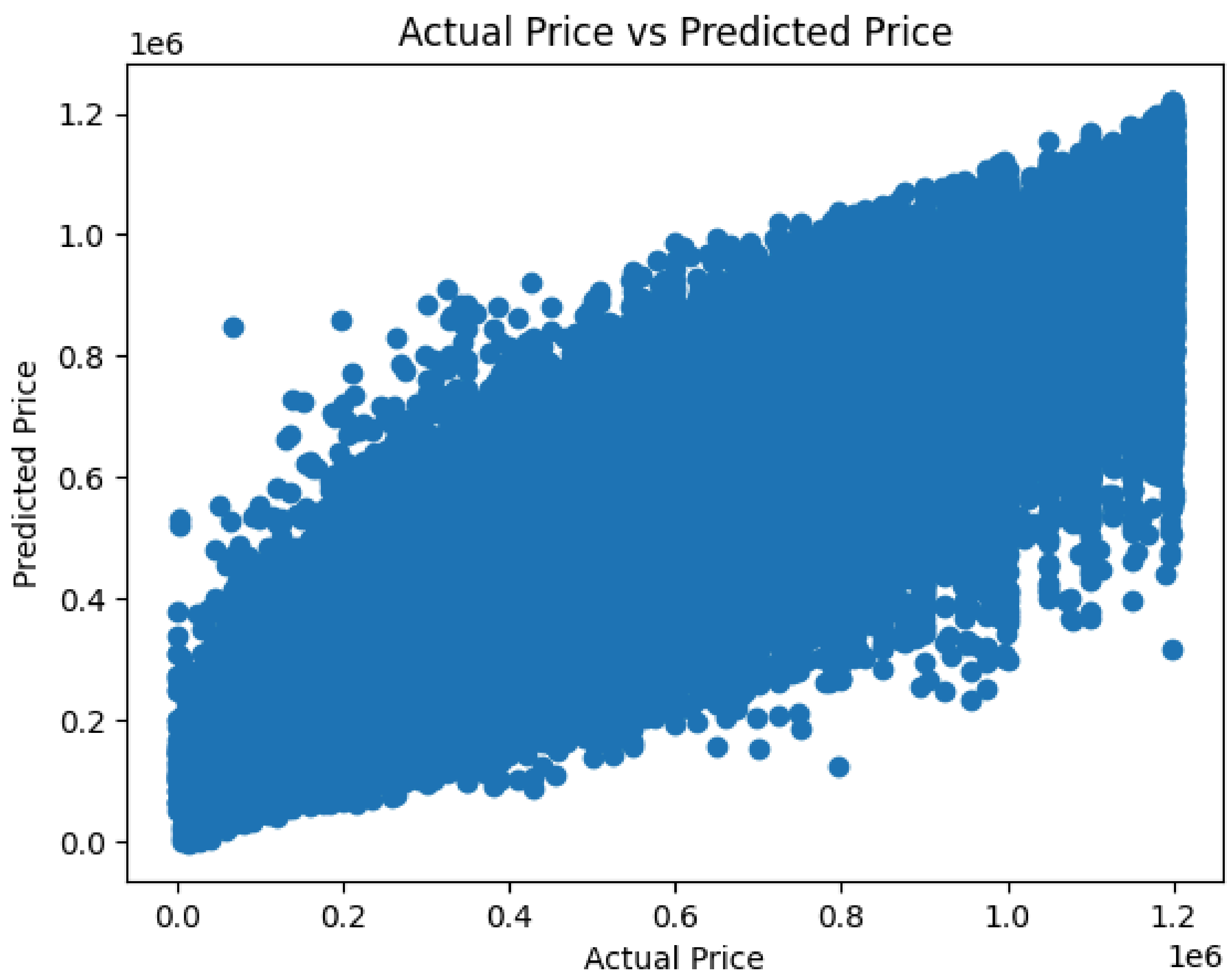


Randolph, K. A., & Myers, L. L. (2013). Bivariate and Multivariate Linear Regression Analysis. Basic Statistics in Multivariate Analysis, 109–132. https://doi.org/10.1093/acprof:oso/9780199764044.003.0005

Extreme Gradient Boosting (XGBRegressor)

“XGBoost stands for Extreme Gradient Boosting, which applies a Gradient Boosting technique based on decision trees. It constructs short, basic decision trees iteratively”(Subasi et al., 2022).

R-Squared Result:
~0.89 on training data
~0.74 on testing data



Subasi, A., Panigrahi, S. S., Patil, B. S., Canbaz, M. A., & Klén, R. (2022). Advanced pattern recognition tools for disease diagnosis. 5G IoT and Edge Computing for Smart Healthcare, 195–229. https://doi.org/10.1016/b978-0-323-90548-0.00011-5

Training the XGBRegressor Model

Parameters adjusted to fit the model:

- n-estimators
- training/testing set size
- tree depth
- learning rate

Started with:

- 50 estimators
- training set 60% of dataset
- max depth of 6
- learning rate of 0.1

Results:

- R-squared ~0.69 on training data
- R-squared of ~0.69 on testing data

Ended with:

- 5000 estimators
- training set 60% of dataset
- max depth of 8
- learning rate of 0.1

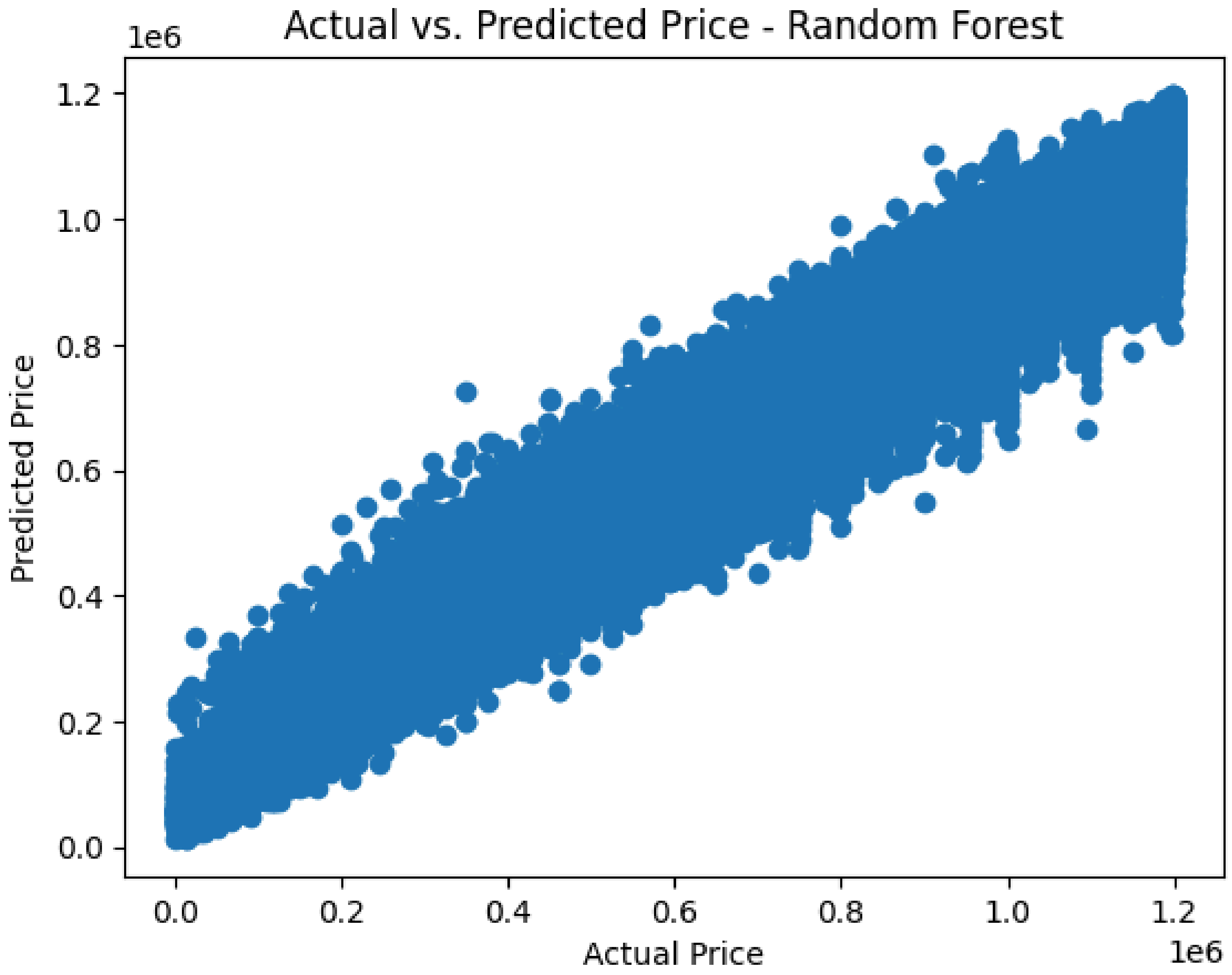
Results:

- R-squared ~0.89 on training data
- R-squared ~0.74 on testing data

Random Forest Regression

“A random forest is a meta estimator that fits a number of decision tree regressors on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting”(Sklearn.ensemble.randomforestregressor, n.d.).

R-Squared Results:
~0.98 on training data
~0.86 on testing data



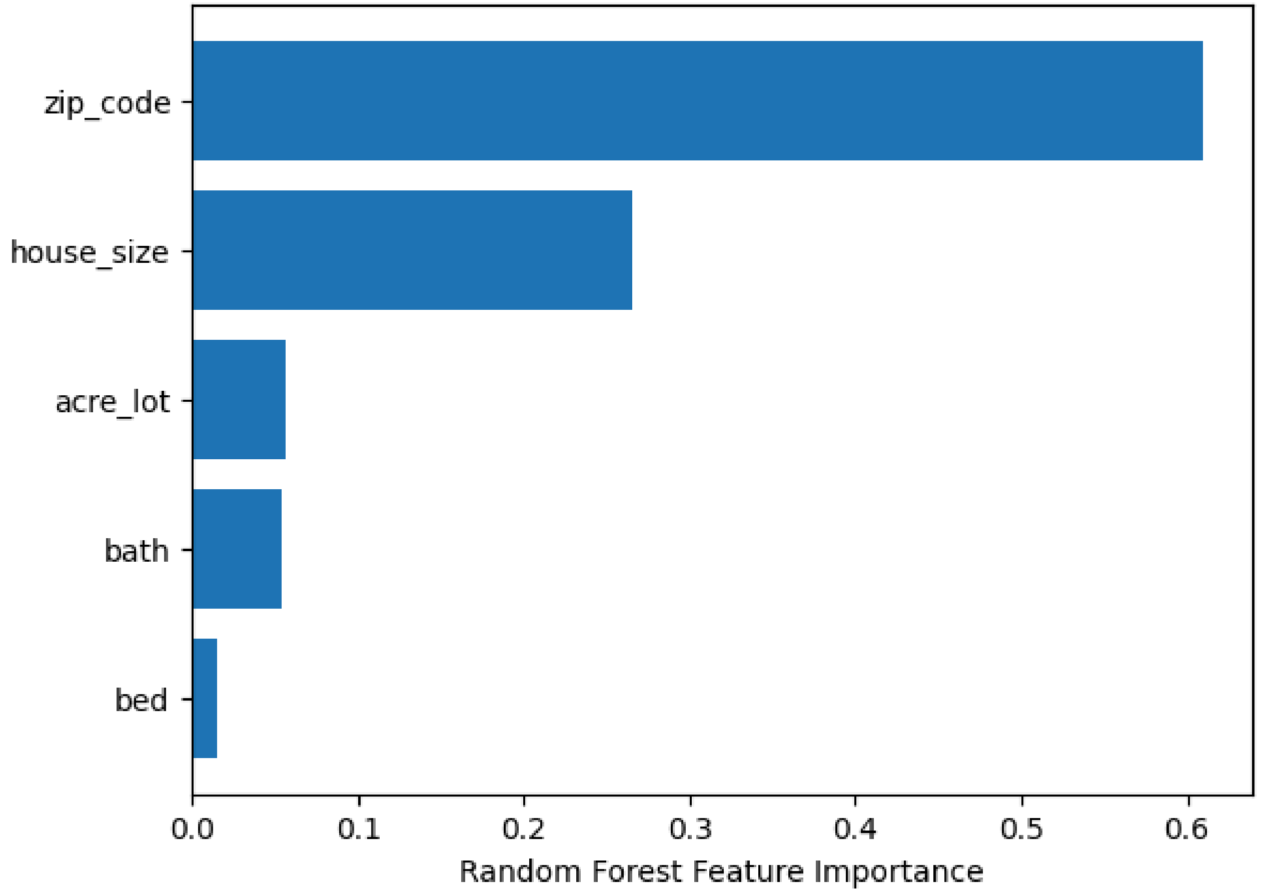
Sklearn.ensemble.randomforestregressor. scikit. (n.d.). https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

Training the Random Forest Model

Parameters adjusted to fit the model:

- n-estimators
- training/testing set size

Feature Importances:



Started with:

- 50 estimators
- training set 90% of dataset

Results:

- R-squared ~0.978 on training data
- R-squared ~0.855 on testing data

Ended with:

- 200 estimators
- training set 90% of dataset

Results:

- R-squared ~0.979 on training data
- R-squared ~0.856 on testing data

RESULTS

BEST

RANDOM FOREST

R-squared value was high, fit the data very well

WORST

LINEAR REGRESSION

R-squared value was poor, did not fit the data well

LIMITATIONS

Since extreme outliers were filtered out from our dataset, the resulting models are less robust. The models will not be able to accurately predict the price of very large and/or expensive homes.

REAL WORLD APPLICATION

- The algorithm could be used to predict single-family home prices based on the various features.

- Real estate agencies can use the algorithm to provide more accurate home prices to clients and helping sellers set competitive prices.

- City planners can use the algorithm to analyze housing markets and trends, helping them make informed decisions about zoning regulations, and urban revitalization planning.

QUESTIONS?