← Back to Home

Final Report

Table of Contents

Introduction

Predicting house prices is a widely discussed problem in the fields of real estate economics, urban planning and government policy control. This project aims to predict house prices in the top 50 US cities using machine learning techniques. By leveraging a dataset that includes factors like zip code population, median household income, and property features, we intend to build an accurate housing price estimation model. Accurate house price prediction is crucial in today's market due to its implications in multiple sectors, including household decision-making, urban development, and economic policy-making. This report summarizes the progress made up to the final checkpoint for our project on predicting house prices using machine learning models. We have focused on data preprocessing and implementing initial models to establish a baseline for future improvements.

Problem Definition

The primary problem is predicting house prices based on various demographic and property-related features. Given the complex factors influencing the real estate market, developing an accurate prediction model can assist stakeholders in making informed decisions. This project aims to create a robust machine learning model to provide these insights.

Dataset Overview

Our raw dataset contains 14 columns and 39,981 entries, with a mix of continuous, categorical, and location-based variables.

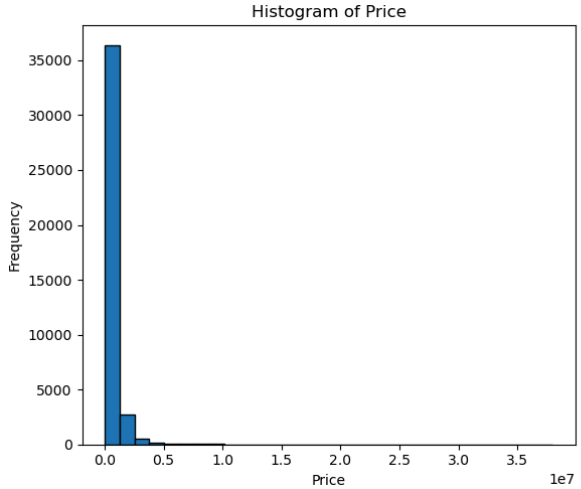
Continuous Variables

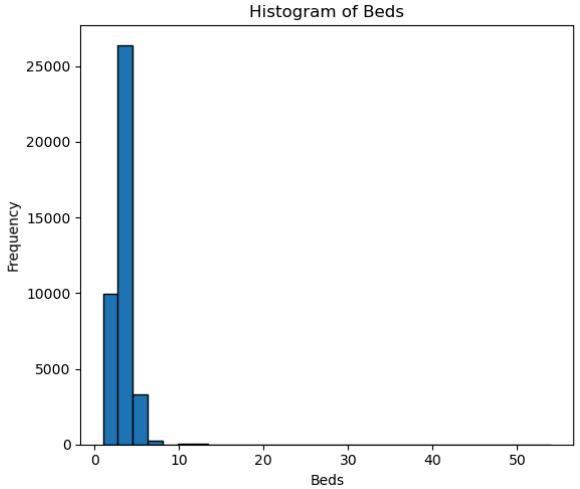
The continuous variables we analyzed include:

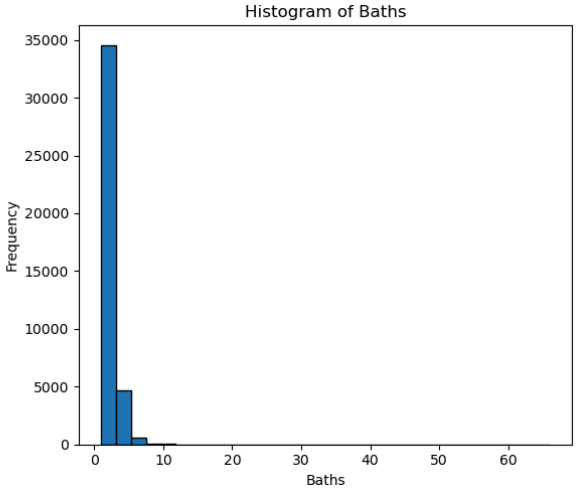
- **Price**
- **Beds**
- **Baths**
- **Living Space**
- **Zip Code Population**
- **Zip Code Density**
- **Median Household Income**
- **Latitude**
- **Longitude**

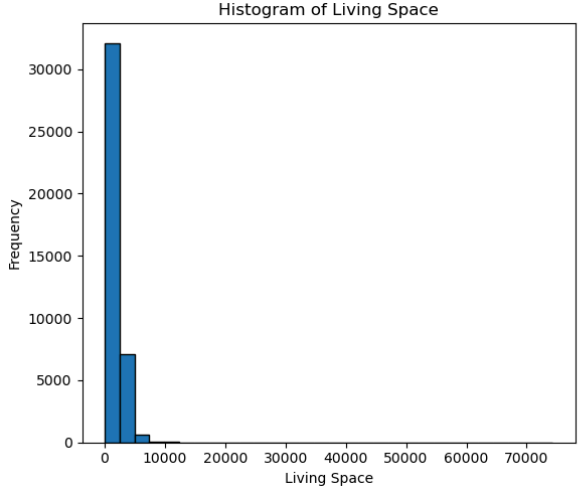
The boxplot and histogram will show the distribution and outliers of the continuous variables.

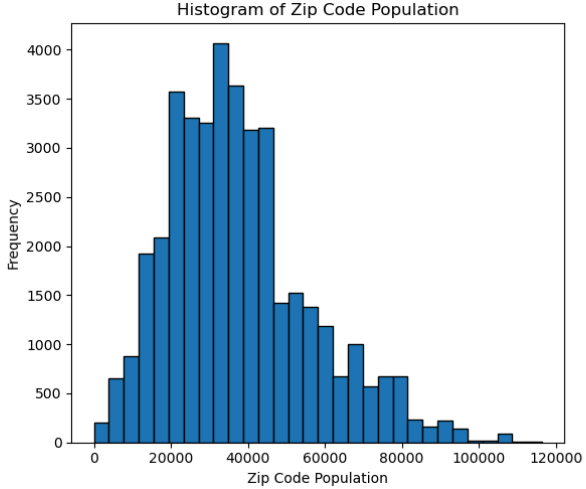
Distribution of Continuous Variables

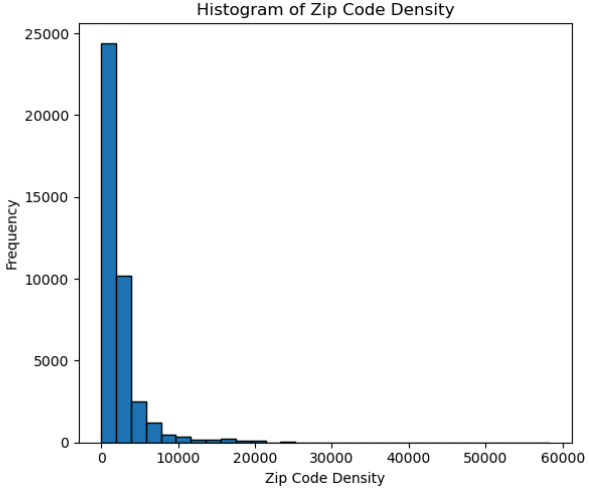

Histogram of Price
Frequency
Price
1e7

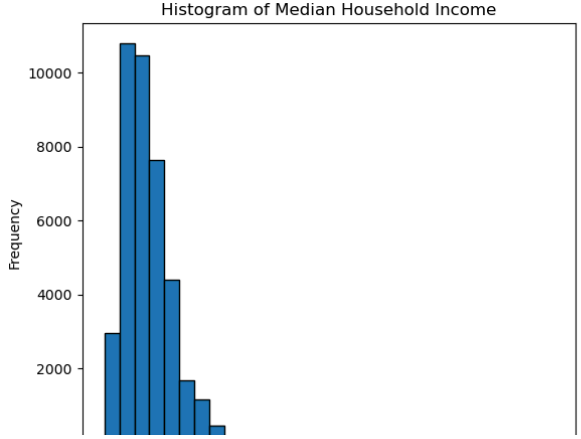

Histogram of Beds
Frequency
Beds

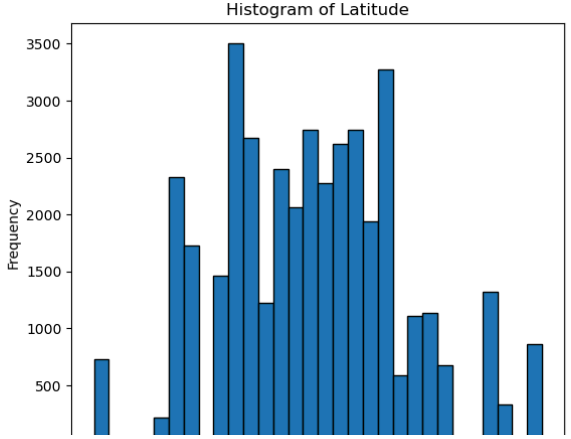

Histogram of Baths
Frequency
Baths

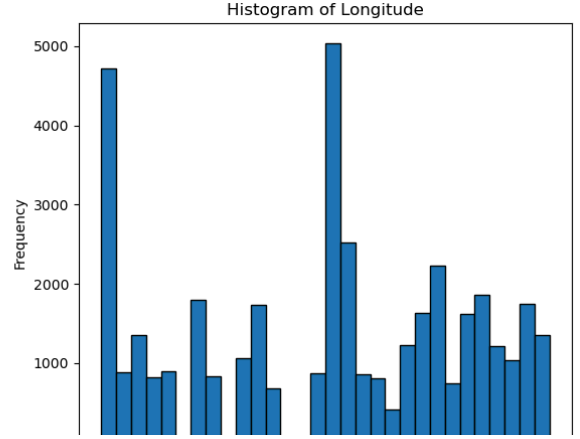

Histogram of Living Space
Frequency
Living Space

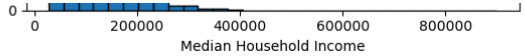

Histogram of Zip Code Population
Frequency
Zip Code Population

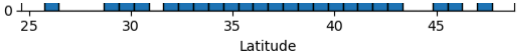

Histogram of Zip Code Density
Frequency
Zip Code Density

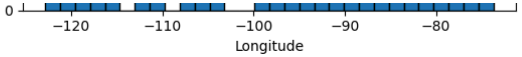

Histogram of Median Household Income
Frequency


Histogram of Latitude
Frequency

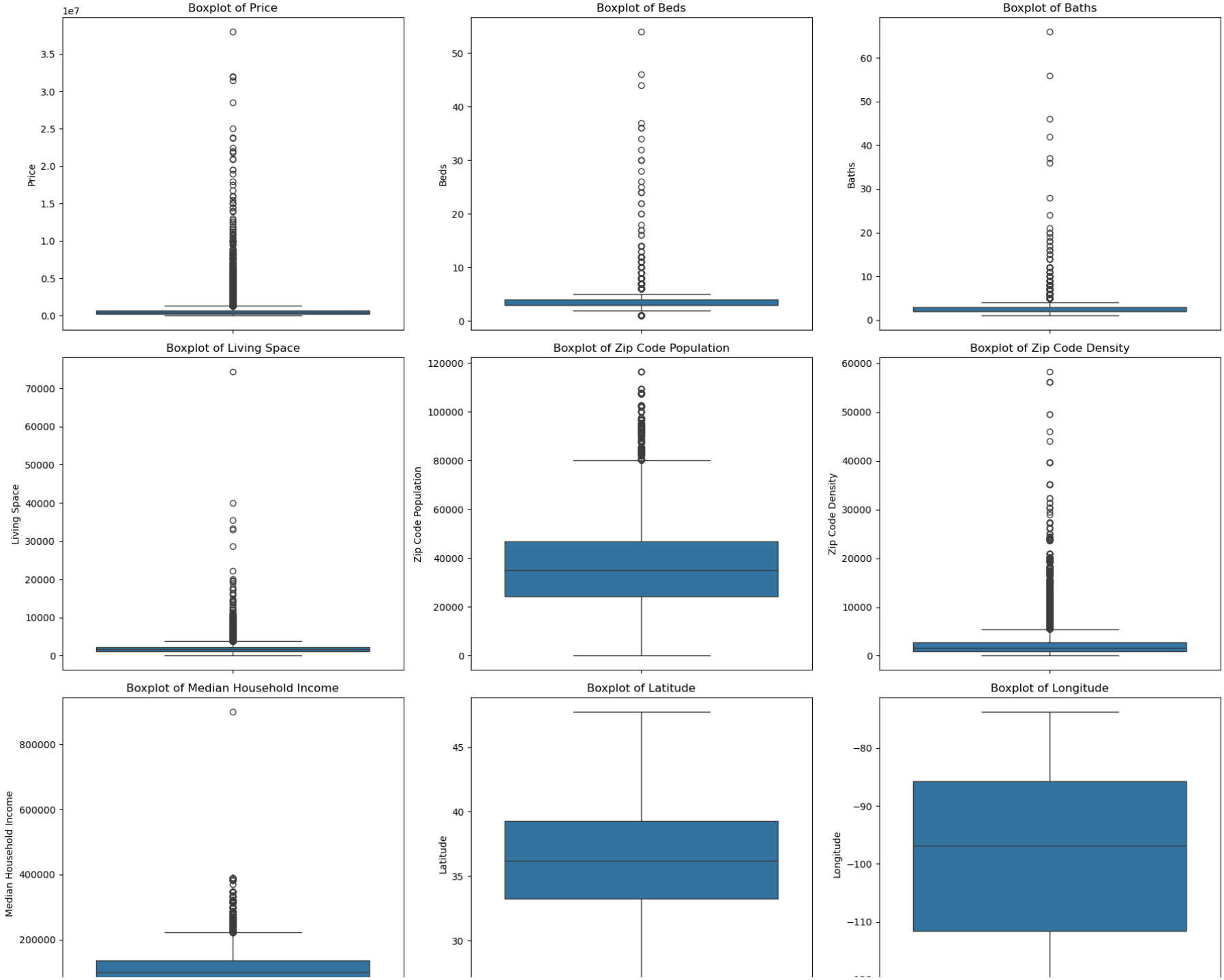

Histogram of Longitude
Frequency

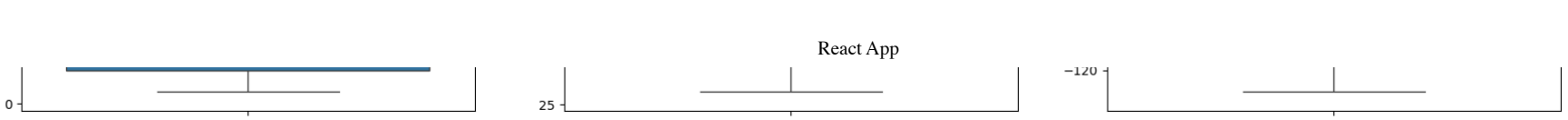






Outliers of Continuous Variables

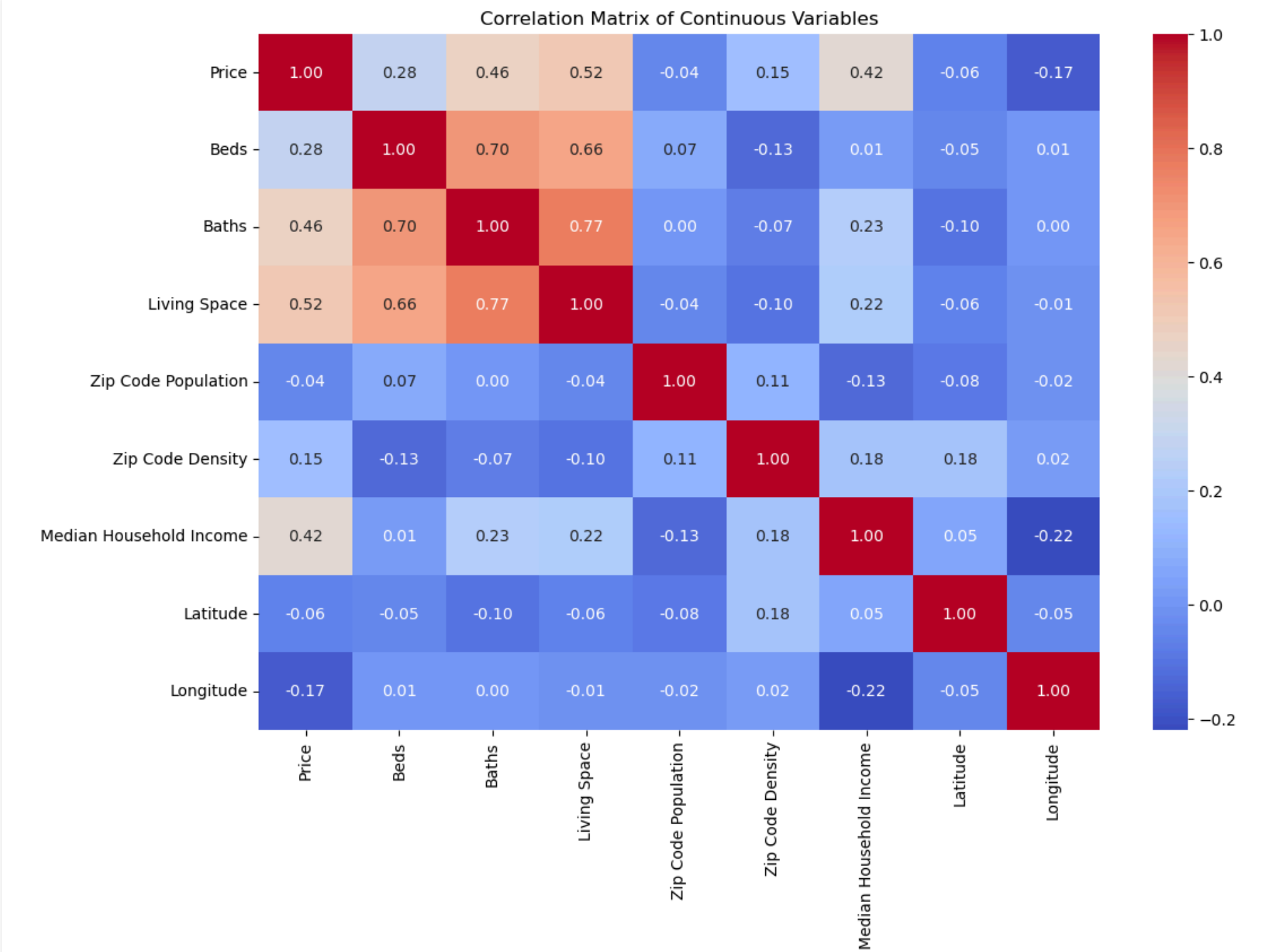

Boxplot of Price
1e7
Price
Boxplot of Beds
Beds
Boxplot of Baths
Baths
Boxplot of Living Space
Living Space
Boxplot of Zip Code Population
Zip Code Population
Boxplot of Zip Code Density
Zip Code Density
Boxplot of Median Household Income
Median Household Income
Boxplot of Latitude
Latitude
Boxplot of Longitude
Longitude



As we can tell from the plots above, the distributions of variables are commonly right skewed with several outliers located above boundary. In order to eliminate the influence of unbalanced data, we will take the logarithm of some variables in the preprocessing part.

Next, in order to intuitively explore different variables' relationship with price, the correlation matrix is shown as below.

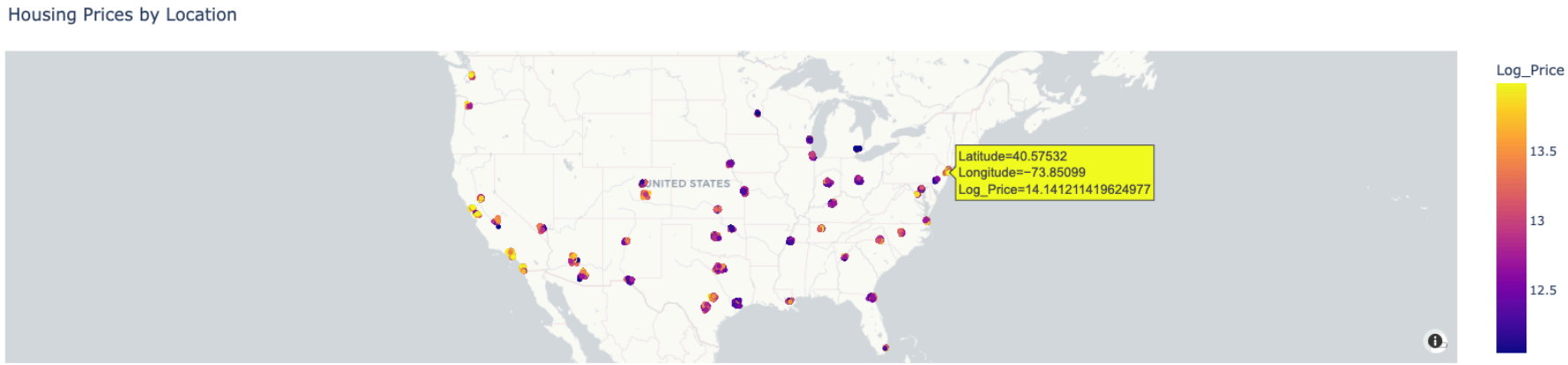
Correlation Matrix of Continuous Variables


Correlation Matrix of Continuous Variables
Price
Beds
Baths
Living Space
Zip Code Population
Zip Code Density
Median Household Income
Latitude
Longitude
1.00 0.28 0.46 0.52 -0.04 0.15 0.42 -0.06 -0.17
0.28 1.00 0.70 0.66 0.07 -0.13 0.01 -0.05 0.01
0.46 0.70 1.00 0.77 0.00 -0.07 0.23 -0.10 0.00
0.52 0.66 0.77 1.00 -0.04 -0.10 0.22 -0.06 -0.01
-0.04 0.07 0.00 -0.04 1.00 0.11 -0.13 -0.08 -0.02
0.15 -0.13 -0.07 -0.10 0.11 1.00 0.18 0.18 0.02
0.42 0.01 0.23 0.22 -0.13 0.18 1.00 0.05 -0.22
-0.06 -0.05 -0.10 -0.06 -0.08 0.18 0.05 1.00 -0.05
-0.17 0.01 0.00 -0.01 -0.02 0.02 -0.22 -0.05 1.00
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

Naively speaking, Living space, Baths and Household Income have the strongest correlations with price. We will examine this more rigorously in next parts.

Location Variables

Our dataset also contains rich information about the location of properties. According to the literature, location is always a determining factor when considering house price.



According to our data visualization, the east and west coast have higher house prices than the central region, especially in New York City and San Francisco. This is also consistent with the reality and provides support for the rationality of our data.

Data Preprocessing

We performed extensive data preprocessing to prepare the dataset for modeling:

- **Loaded Raw Data:** Features include Zip Code, Price, Beds, Baths, Living Space, City, State, Zip Code Population, Zip Code Density, County, Median Household Income, Latitude, and Longitude.
- **Imputed Missing Values:** Filled missing numerical values with the mean.
- **Dropped Irrelevant Columns:** Removed columns like 'Address' that do not contribute to prediction.

- **Encoded Categorical Variables:** Label-encoded 'City', 'State', and 'County'.
- **Feature Transformation:** Applied log transformations to skewed features ('Living Space', 'Zip Code Population', 'Zip Code Density', 'Median Household Income').
- **Feature Engineering:** Created interaction terms like 'Beds_Baths' and polynomial features up to degree 2.
- **Feature Scaling:** Standardized numerical features using StandardScaler.
- **Saved Preprocessed Data:** Stored as `preprocessed_data.csv` for model training.
- **Created Sample for Frontend:** Generated `preprocessed_data_sample.csv` with 200 rows for the frontend.

Model Implementation

We implemented and evaluated machine learning models to predict house prices.

a. Linear Regression

We use linear regression model as a baseline.

$$Price = \theta_0 + \theta_1 \cdot State + \theta_2 \cdot Beds + \theta_3 \cdot Baths + \theta_4 \cdot Log\_Living\_Space + \theta_5 \cdot Longitude + \theta_6 \cdot Log\_Zip\_C$$

**Among the variables we used, State is the categorical variable which we encoded it into one-hot.*

b. Random Forest Regression

The Random Forest model was implemented both from scratch and using scikit-learn's library for comparison.

- **Custom Implementation:**
  - Created a `DecisionTreeNode` class to represent nodes in the decision tree.
  - Implemented decision tree logic using Mean Squared Error (MSE) as the splitting criterion.
  - Built a Random Forest Regressor that aggregates predictions from multiple decision trees.
  - Calculated feature importances by accumulating impurity decreases across all trees.

- **scikit-learn Implementation:**
  - Utilized `RandomForestRegressor` from scikit-learn.
  - Performed hyperparameter tuning using `RandomizedSearchCV` and `GridSearchCV`.
  - Leveraged scikit-learn's optimized algorithms for faster training and better performance.

c. Gradient Boosting Regression

We implemented Gradient Boosting Regression models to improve prediction performance.

- **Custom Implementation:**
  - Developed a Gradient Boosting Regressor from scratch using decision trees as weak learners.
  - Implemented gradient boosting logic to minimize the loss function iteratively.
  - Set parameters: `n_estimators=100`, `learning_rate=0.1`, `max_depth=3`.
- **scikit-learn Implementation:**
  - Used `GradientBoostingRegressor` from scikit-learn.
  - Conducted hyperparameter tuning using `GridSearchCV` to optimize model performance.
  - Explored parameters like `n_estimators`, `learning_rate`, `max_depth`, etc.

d. K-Means Clustering

We also implemented K-Means clustering model to identify distinct clusters among the dataset.

- **Scikit-learn's KMeans clustering model:**
  - The `init=random` method was used to initialize centroids by selecting random observations from the dataset.
  - `random_state` was set to 42.
  - Default parameters were used for KMeans except for `n_clusters`, which varied to determine the optimal number of clusters.
- **Elbow Method for Determining the Optimal Number of Clusters (K)**
  - The K-Means algorithm was executed with varying values of `n_clusters` ranging from 1 to 21.

- Get the sum of squared distances of samples to their closest cluster center (WCSS) from KMeans model attribute `inertia_`
- To determine the optimal number of clusters, **WCSS** was plotted against **k**

Issues Addressed

a. Feature Importances Being Zero

- **Problem:** Feature importances were all zeros in the custom implementation.
- **Cause:** Incorrect calculation of impurity decreases during tree traversal.
- **Solution:**
  - Corrected the impurity calculation by properly computing the variance at each node.
  - Modified the `DecisionTreeNode` class to store target values at nodes.
  - Updated the traversal method to accumulate impurity decreases weighted by the number of samples.

b. Predicted Prices Plateauing

- **Problem:** Model predictions plateaued for higher "Living Space" values.
- **Cause:** Model limitations due to data sparsity at high "Living Space" values.
- **Solution:**
  - Applied log transformations to reduce skewness in "Living Space" and other features.
  - Enhanced feature engineering to capture non-linear relationships.
  - Increased model complexity by adjusting hyperparameters like `max_depth` and `n_estimators`.

Results

a. Linear Regression Results

The Linear Regression model gives us r2_score of 0.3602, and RMSE of 830816.62. Next, we use Lasso and Ridge Regression for regularization.

Model	Best alpha	RMSE ($)	r2_score
Linear regression	/	830816.62	0.3602
Lasso regression	0.8947	830822.19	0.3602
Ridge regression	0.1368	830836.14	0.3602

Important features are:

Selected features by Lasso (Importance Descending Order):

```
['State_New York', 'State_Michigan', 'State_Washington',
       'State_Wisconsin', 'State_District of Columbia', 'State_Minnesota',
       'State_Pennsylvania', 'State_Ohio', 'State_Illinois', 'State_Indiana',
       'State_Oregon', 'State_Virginia', 'State_Maryland', 'State_Kentucky',
       'State_Tennessee', 'State_California', 'State_North Carolina', 'State_Nebraska',
       'State_Colorado', 'State_Missouri', 'Log_Median Household Income',
       'Log_Living Space', 'State_Georgia', 'State_Florida', 'State_Kansas', 'Baths',
       'State_Louisiana', 'State_Oklahoma', 'State_New Mexico', 'Log_Zip Code Density',
       'Log_Zip Code Population', 'State_Texas', 'Latitude', 'State_Nevada', 'Beds',
       'Longitude', 'Beds_Baths']
```

It in part matches our estimates in the data visualization part.

The linear regression model didn't take into account more detailed location information such as City and County. Its RMSE also does not meet our expectation. We used other advanced methods to further explore the data and better predict the house price.

b. Random Forest Results

The Random Forest models were evaluated using performance metrics such as MAE, RMSE, R² Score, and MAPE.

Custom Implementation Results

n_estimators	max_depth	MAE ($)	RMSE ($)	R² Score	MAPE (%)
50	10	191,097.33	629,436.76	0.6328	64.22
50	15	191,509.42	630,559.92	0.6315	63.62
50	20	178,628.96	622,411.25	0.6409	60.14
100	10	195,076.67	625,415.23	0.6374	64.17
100	15	184,683.71	618,416.55	0.6455	60.76
100	20	189,833.20	649,842.94	0.6086	62.72

Best Model:

- **n_estimators:** 100
- **max_depth:** 15
- **Performance Metrics:**
  - **MAE:** $184,683.71
  - **RMSE:** $618,416.55
  - **R² Score:** 0.6455
  - **MAPE:** 60.76%

scikit-learn Implementation Results

After hyperparameter tuning using `RandomizedSearchCV` and `GridSearchCV`, the following results were obtained:

Best Parameters Found:

```
{
  "bootstrap": false,
  "max_depth": 15,
  "max_features": "log2",
  "min_samples_leaf": 5,
  "min_samples_split": 2,
  "n_estimators": 178
}
```


Performance Metrics:

- **MAE:** $150,668.92
- **RMSE:** $537,077.09
- **R² Score:** 0.7326
- **MAPE:** 50.44%

Feature Importances:


```
{
  "features": [
    "Zip Code",
    "Beds",
    "Baths",
    "City",
    "State",
    "County",
    "Latitude",
    "Longitude",
    "Log_Living Space",
    "Log_Zip Code Population",
    "Log_Zip Code Density",
    "Log_Median Household Income",
    "Beds_Baths"
  ],
  "importances": [
    0.0672, 0.0320, 0.1448, 0.0212, 0.0432, 0.0263, 0.0660, 0.0535,
```

```
      0.2571, 0.0209, 0.0473, 0.1259, 0.0946
    ]
  }
```

Visualizations

Below are visualizations comparing the performance metrics and feature importances of the models:

Model Performance Comparison

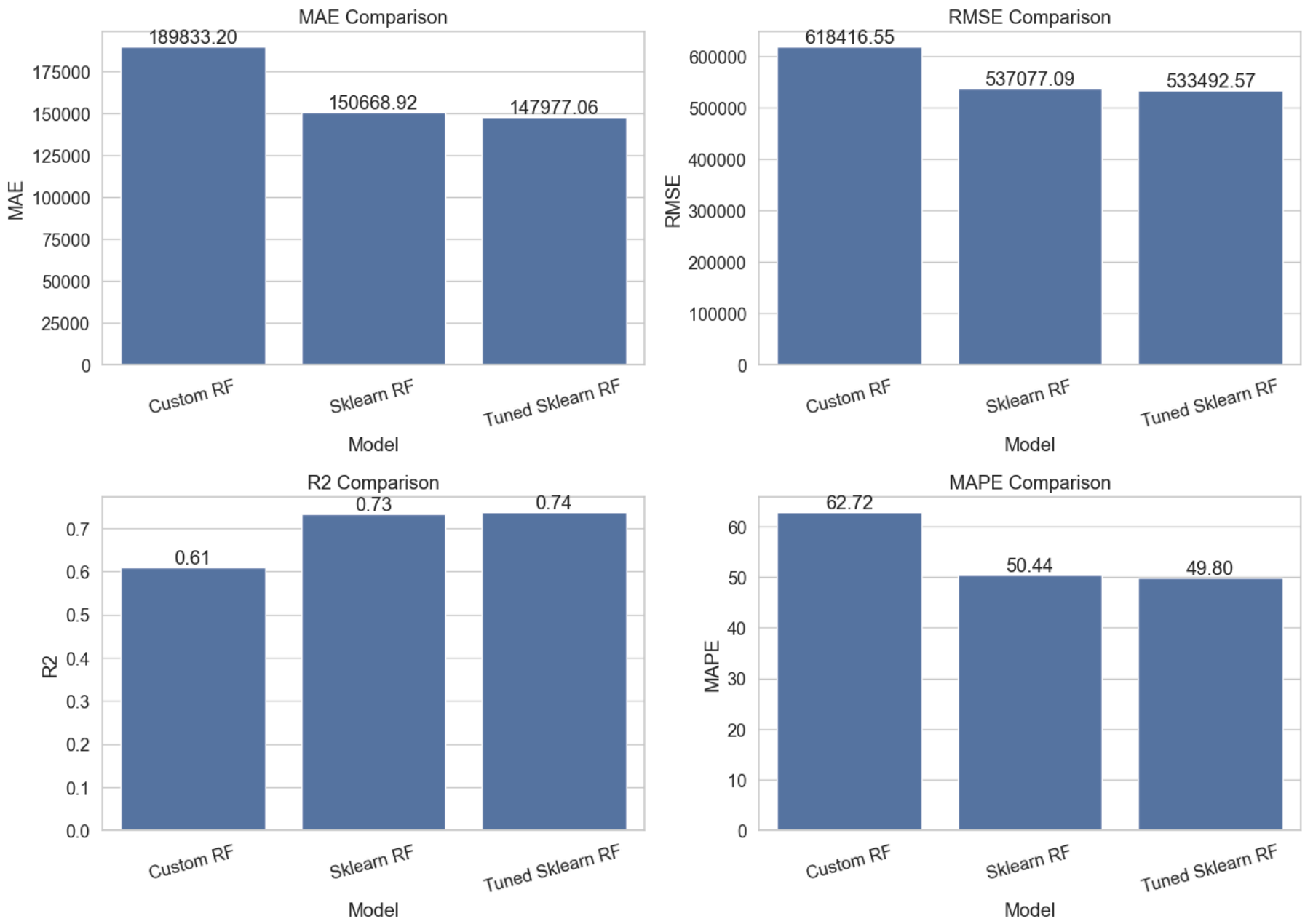


Figure 1: Comparison of MAE, RMSE, R² Score, and MAPE across different models.

Feature Importances

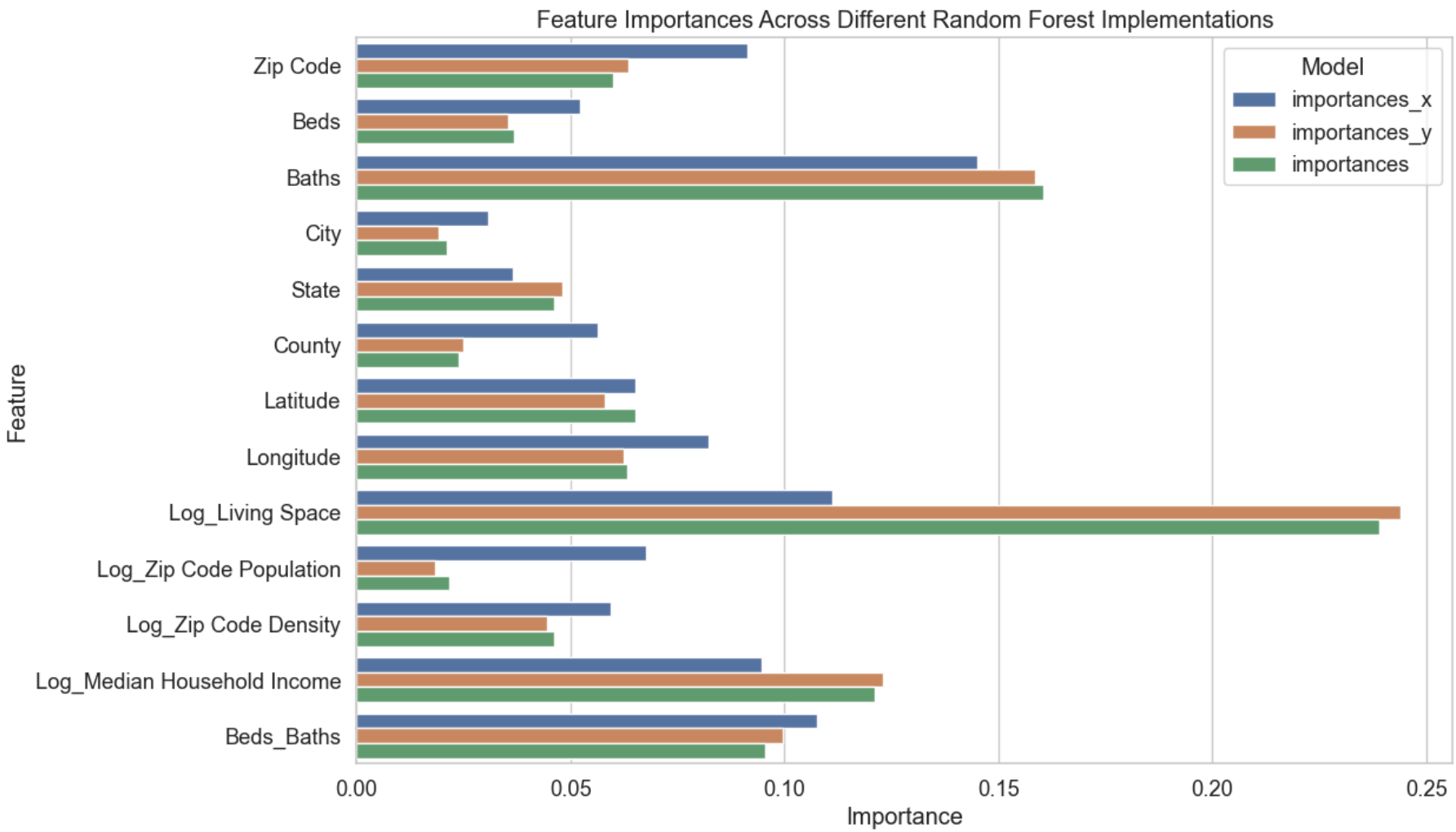


Figure 2: Feature importances as determined by the Random Forest models.

c. Gradient Boosting Results

The Gradient Boosting models showed improved performance over Random Forest models.

Custom Implementation Results

Parameters	MAE ($)	RMSE ($)	R² Score	MAPE (%)
n_estimators=100, learning_rate=0.1, max_depth=3	180,611.69	537,090.16	0.7326	55.48

scikit-learn Implementation Results

After hyperparameter tuning using `GridSearchCV`, the following results were obtained:

Best Parameters Found:

```
{
  "learning_rate": 0.0659,
  "max_depth": 7,
  "max_features": null,
  "min_samples_leaf": 1,
  "min_samples_split": 4,
  "n_estimators": 177,
  "subsample": 0.7491
}
```

Performance Metrics:

- **MAE:** $147,977.62
- **RMSE:** $533,492.22
- **R² Score:** 0.7362
- **MAPE:** 50.21%

Feature Importances:

```
{
  "features": [
    "Zip Code",
    "Beds",
    "Baths",
    "City",
    "State",
    "County",
    "Latitude",
    "Longitude",
    "Log_Living Space",
    "Log_Zip Code Population",
    "Log_Zip Code Density",
    "Log_Median Household Income",
    "Beds_Baths"
  ],
  "importances": [
    0.0564, 0.0325, 0.0853, 0.0202, 0.0440, 0.0244, 0.0740, 0.0495,
    0.2976, 0.0246, 0.0425, 0.1540, 0.0945
  ]
}
```


Visualizations

Below are visualizations comparing the performance metrics and feature importances of the models:

Model Performance Comparison

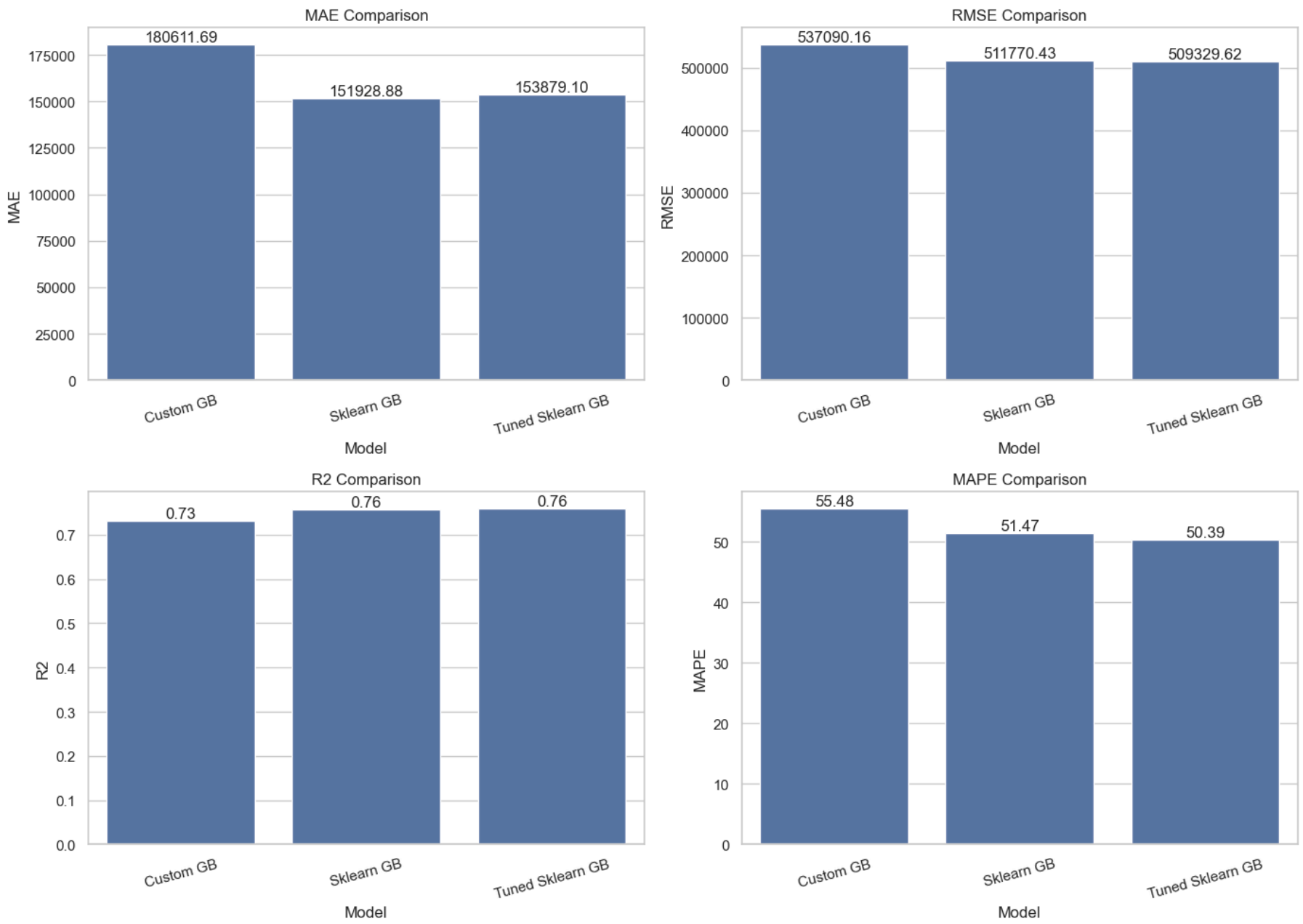


Figure 1: Comparison of MAE, RMSE, R² Score, and MAPE across different models.

Feature Importances

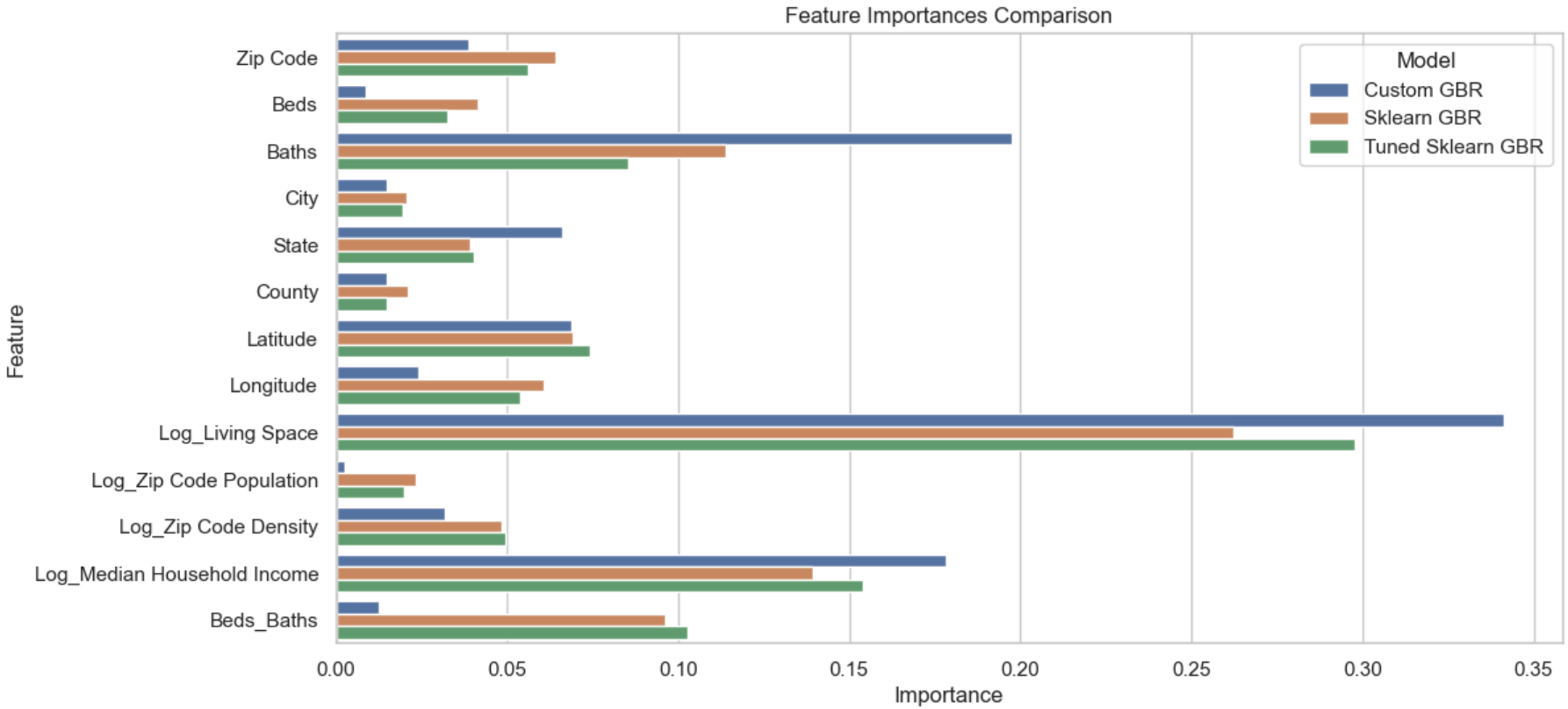


Figure 2: Feature importances as determined by the Gradient Boosting models.

d. Comparison of Models

Implementation	RMSE ($)	R^2 Score	MAE ($)	MAPE (%)
Custom Random Forest	618,416.55	0.6455	184,683.71	60.76
scikit-learn Random Forest	537,077.09	0.7326	150,668.92	50.44
Custom Gradient Boosting	537,090.16	0.7326	180,611.69	55.48

Implementation	RMSE ($)	R² Score	MAE ($)	MAPE (%)
scikit-learn Gradient Boosting	533,492.22	0.7362	147,977.62	50.21

Insights:

- **Improved Performance:** Gradient Boosting models generally performed better than Random Forest models.
- **Best RMSE and MAE:** The hyperparameter-tuned scikit-learn Gradient Boosting model achieved the lowest RMSE and MAE.
- **Consistent Top Feature:** "Log_Living Space" remains the most important feature across all models.
- **Effectiveness of Hyperparameter Tuning:** Tuning significantly improved model performance.

d. K-Means Clustering Results

Elbow Method Result

Based on the **Within-Cluster Sum of Squares (WCSS)** vs. **k** plot shown below, the "elbow point" is approximately at $k = 5$. This suggests that the optimal number of clusters for this dataset is around 5.

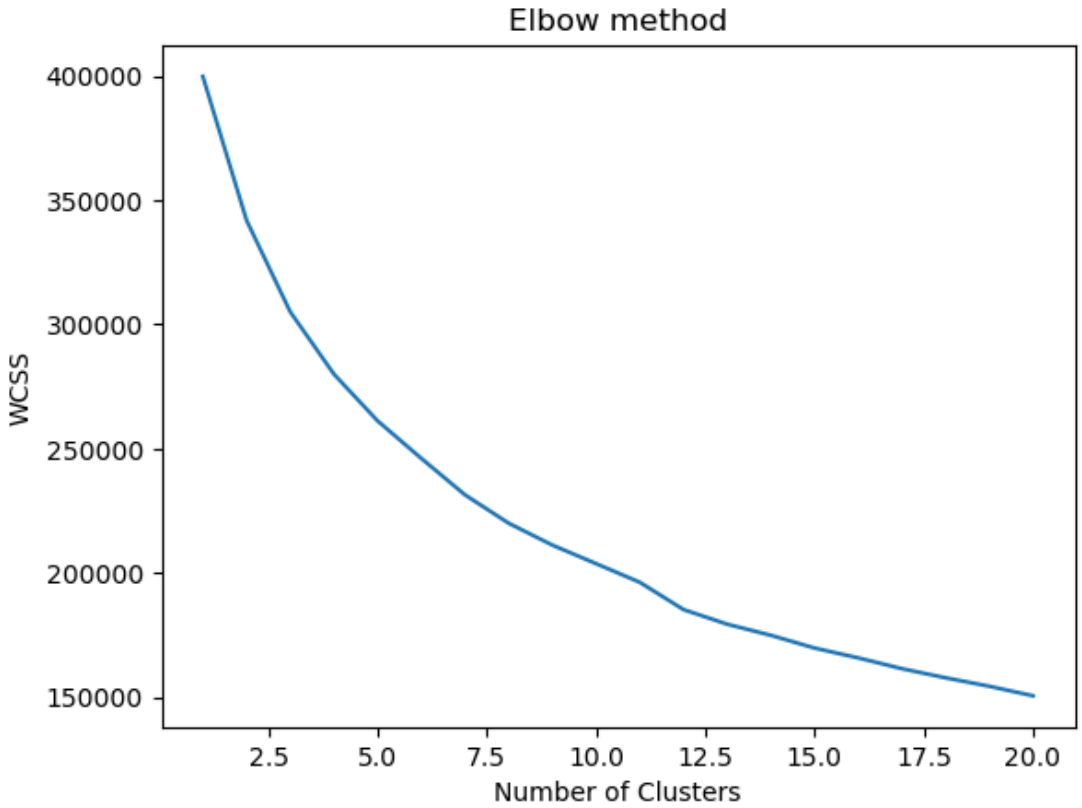


Figure 1: Elbow Method, WCSS vs. k

KMeans Clustering Results

In this analysis, we set the number of clusters to **5** based on the elbow method and performed K-Means clustering using the **scikit-learn** library. The algorithm identified five distinct clusters, segmenting the dataset effectively. The table below provides a summary of the statistics of data points in each identified cluster:

Cluster Label	Number of Points	Avg Price	Avg Beds	Avg Baths	Avg Living Space	Avg Median Household Income	Avg Zip Code Density
0	10,340	335,146	2.661	1.795	7.120	11.388	7.798

Cluster Label	Number of Points	Avg Price	Avg Beds	Avg Baths	Avg Living Space	Avg Median Household Income	Avg Zip Code Density
1	9,004	686,715	2.749	2.076	7.232	11.638	7.647
2	524	5,976,990	6.576	7.261	8.632	12.048	7.832
3	7,800	963,185	4.244	3.730	8.011	11.778	6.873
4	12,313	374,066	3.086	2.311	7.388	11.405	6.881

Table 1: Statistics of Clustered Data

From the table above, we summarize 5 potential housing categories as below:

- **Cluster 0**:
  - The second-largest cluster with **10,340** data points
  - Lowest price, fewest beds and baths, lowest living space
  - Lowest household income, and high zip code density.
- **Cluster 1**:
  - The third-largest cluster with **9,004** points
  - Medium price, medium beds and baths, medium living space
  - Medium household income, and medium zip code density.
- **Cluster 2**:
  - The smallest cluster, with only **524** points
  - Highest price, most beds and baths, highest living space
  - Highest household income, and highest zip code density.
- **Cluster 3**:
  - The second-smallest cluster with **7,800** points
  - Second highest price, second most beds and baths, second highest living space
  - Second highest household income, and lowest zip code density.

- **Cluster 4**:
  - The largest cluster with **12,313** points
  - Second lowest price, medium beds and baths, medium living space
  - Medium household income, and second lowest zip code density.

The KDE plots visualize the distribution of key features for each cluster identified by the K-Means algorithm. The features visualized include **Price**, **Beds**, **Baths**, **Log Living Space**, **Log Median Household Income**, and **Log Zip Code Density**. From the KDE plots we get clusters are well-separated on **Price**, **Beds**, **Baths**, **Living Space**, and **Median Household Income**, while **Zip Code Density** is more overlapped. It means that 5 well-separated features are more important than **Zip Code Density** in clustering. It also proves that our clustering model is effective.

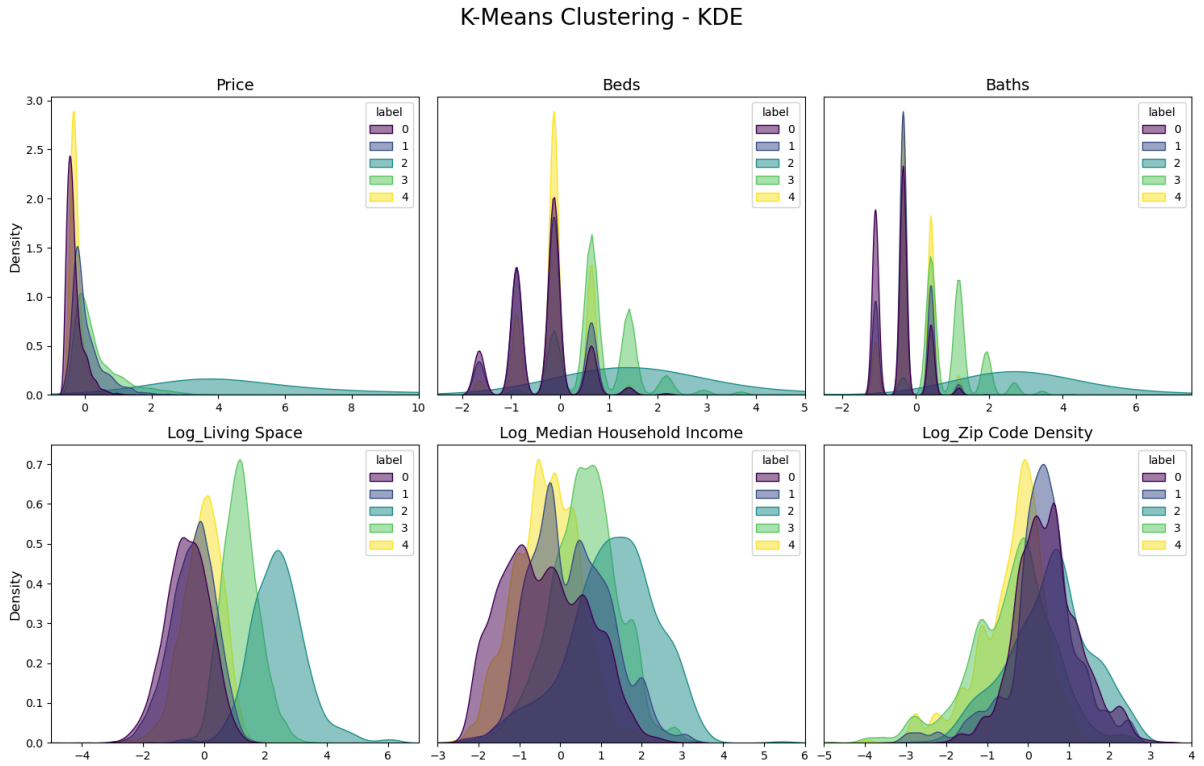


Figure 2: KDE plots after K-Means clustering

The pairplot below illustrates the relationships between key features, including **Price**, **Beds**, **Baths**, **Log Living Space**, **Log Median Household Income**, and **Log Zip Code Density**. It shows strong positive correlations such as **Price** vs. **Living Space**, **Baths** vs. **Living Space**, and **Beds** vs. **Baths**. This plot also demonstrates clusters are well seperated on all 5 key features except for **Zip Code Density**.

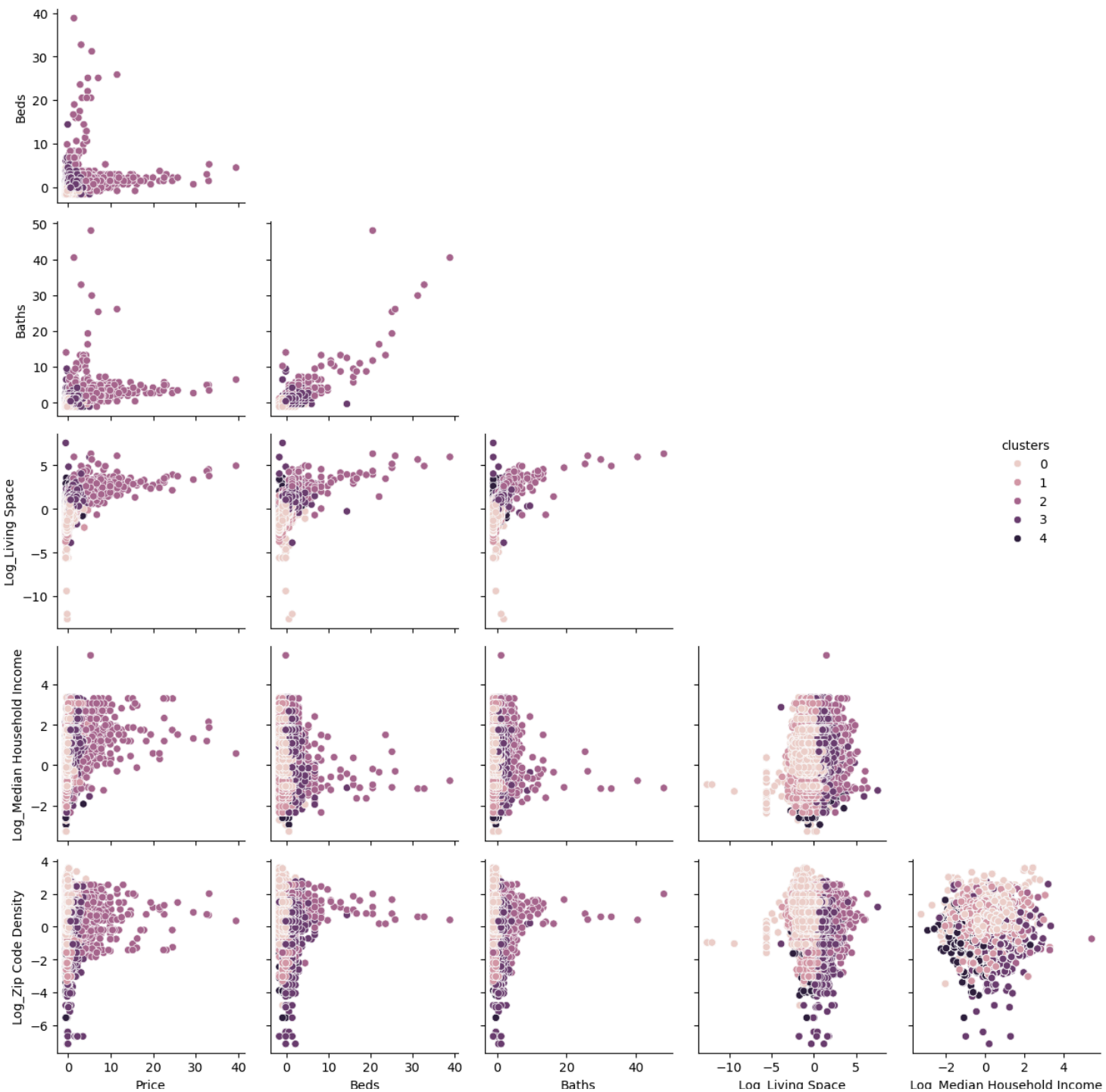

Beds
Baths
Log_Living Space
Log_Median Household Income
Log_Zip Code Density
Price
Beds
Baths
Log_Living Space
Log_Median Household Income
clusters
0
1
2
3
4

Figure 3: Pairplot after K-Means clustering

Clustering Discussion

The clustering results offer valuable insights on strategic decisions in the housing market.

- **1. Predicting Housing Market Demand in Specific Regions**
  - Cluster 0 and Cluster 4, two of the largest clusters, indicate significant demand in their segments. Cluster 0 appeals to budget-conscious buyers seeking affordable housing, while Cluster 4 targets the middle market.
  - Cluster 2's size is relatively small, but represents the luxury housing market.
- **2. Identifying Target Demographics for Real Estate Developers**
  - Cluster 0 and Cluster 4 are ideal for developers focusing on low price housing.
  - Cluster 1 is good for providing medium-priced housing for middle class family.
  - Cluster 3 targets upper-middle-income families looking for larger living spaces and low population density areas.
  - Cluster 2 attracts high-income individuals. Developers can focus on offering premium amenities and exclusive locations.
- **3. Urban Planning**
  - High-density but low-income areas like Cluster 0 may need to enhance affordable public infrastructure to accommodate the population, such as public transportation, schools, and healthcare, which can improve their living conditions.
  - Low-density areas in Cluster 3 might focus on improving connectivity to necessary amenities such as hospitals, commercial hubs and so on.

Evaluation Against Expected Results

In our initial proposal, we set ambitious goals for our model's performance metrics:

- **Mean Absolute Error (MAE):** less than $30,000
- **Root Mean Squared Error (RMSE):** less than $35,000

- **R^2 Score:** above 0.85

Our actual results are significantly below these targets. The best-performing model achieved an MAE of $147,977.62, RMSE of $533,492.22, and R^2 Score of 0.7362.

This discrepancy is attributed to overconfidence and setting unrealistic goals without thorough data exploration. We underestimated the complexity of the housing data and the challenges in modeling such relationships.

Moving forward, we will recalibrate our expectations based on empirical evidence and focus on incremental improvements.

Next Steps

- **Implement Additional Models:** Explore algorithms like XGBoost, LightGBM, and Neural Networks.
- **Advanced Hyperparameter Tuning:** Use more sophisticated techniques and larger parameter grids.
- **Feature Engineering:** Investigate new features and interactions, and consider dimensionality reduction.
- **Cross-Validation:** Employ k-fold cross-validation for more robust performance evaluation.
- **Data Augmentation:** Collect more data or generate synthetic data to address sparsity issues.
- **Model Ensemble:** Combine predictions from multiple models to enhance performance.
- **Regularization Techniques:** Apply methods to prevent overfitting and improve generalization.
- **User Feedback:** Integrate mechanisms to gather feedback and continuously refine the model.

Summary:

Since the Linear Regression model lacked detailed location information and generated an RMSE of $830,816.62—far exceeding our goal of $35,000—we shifted our focus to the outputs of Random Forest and Gradient Boosting models. The best-performing Gradient Boosting model achieved an R^2 score of 0.7362 and highlighted key predictive features such as living space, household income, and location variables. While this model significantly improved our predictive accuracy, it still did not meet our initial targets. The findings suggest that incorporating more detailed data and advanced modeling techniques could

further enhance the model's performance, benefiting stakeholders looking to make informed decisions in the real estate market.

Gantt Chart

The updated project timeline is available here: Gantt Chart.

Contribution Table

Name	Final Contributions
Yan, Ethan Yikai	• Conducted data preprocessing and feature engineering • Implemented Random Forest and Gradient Boosting models • Performed hyperparameter tuning • Evaluated model performance • Generated visualizations for model comparison • Integrated results into Final Report Page • Created Final Report Page • Drafted documentation and updated next steps

Name	Final Contributions
Li, Yue	• Performed Exploratory Data Analysis (EDA) on the dataset • Implemented Linear Regression Model • Performed Regularization Evaluation • Final report conclusion • Final report presentation slides • Presentation Video creation
Li, Summer	• Integrated Dataset Overview and Linear Regression into Final Report • Final report presentation slides • Created Gantt Chart
Zhang, Wenxi	• Final Report Summary Section • Final Report Presentation Slides
Kang, Zhen	• Implemented K-Means Clustering Model • Visualized Clustering Results and Analyzed Data in Each Cluster • Final Report Presentation Slides

References

1. F. Maloku, B. Maloku, and A. A. D. Kumar, "House price prediction using machine learning and artificial intelligence," *Journal of Artificial Intelligence & Cloud Computing*, vol. 3, no. 4, pp. 2–10, Aug. 2024. DOI: doi.org/10.47363/JAICC/2024(3)357.
2. T. Mao, "Real estate price prediction based on linear regression and machine learning scenarios," *BCP Business & Management EMFRM*, vol. 38, pp. 401–402, 2023. DOI: doi.org/10.54691/bcpbm.v38i.3720.
3. L. Yang, Y. Liang, Q. Zhu, and X. Chu, "Machine learning for inference: using gradient boosting decision tree to assess non-linear effects of bus rapid transit on house prices," *Journal of Urban Technology*, vol. 28, no. 3, pp. 273–284, 2021. DOI: doi.org/10.1080/19475683.2021.1906746.
4. C. Ding and G.-J. Knaap, “Property Values in Inner-City Neighborhoods: The Effects of Homeownership, Housing Investment, and Economic Development,” *Housing Policy Debate*, vol. 13, pp. 701–727, Jan. 2002. DOI: doi.org/10.1080/10511482.2002.9521462.
5. K. E. Case and R. J. Shiller, “The behavior of home buyers in boom and post-boom markets,” *New England Economic Review*, no. Nov, pp. 29–46, 1988.
6. G. Knaap, “The Determinants of Residential Property Values: Implications for Metropolitan Planning,” *Journal of Planning Literature*, vol. 12, no. 3, pp. 267–282, 1998. DOI: doi.org/10.1177/088541229801200301.