Title: California Residential Housing Price Prediction

Name: Yizhen Wang

Affiliation: DSI- MSDS Program

GitHub Repository: https://github.com/yizhen1115/CA_House_Prices.git

1. Introduction

The housing market in California represents an important element of the state's economy, significantly influencing decisions for homeowners, buyers, investors, and policymakers. Accurate prediction of housing prices is not only essential for individuals making substantial financial commitments but also critical for maintaining economic stability. Previous research on the real estate market using machine learning can be broadly categorized into two approaches: house price index trend forecasting and house price valuation (Phan, T. D., 2019). This study focuses on predicting the final close price valuation in California.

The dataset utilized in this study is sourced from the California Regional Multiple Listing Service (CRMLS), a comprehensive repository of single-family home listings contributed by brokers, agents, and sellers. This dataset consists of 50,174 records and 27 features. However, the dataset presents significant challenges, with over 17 features containing missing values. Two features exhibit missing rates exceeding 30%: "Flooring" (39.8%) and "AssociationFeeFrequency" (79.7%).

Previous research on the real estate market using machine learning approaches includes tree-based models such as XGBoost and Random Forest, which excel at capturing non-linear relationships (Truong, Nguyen, Dang, & Mei, 2020). However, house prices often have a linear relationship with the features, such as the area of the house, the number of bedrooms, or the square footage of a home. Thus, linear regression models also are widely used to solve this problem. However, multicollinearity remains a challenge in housing price prediction, prompting researchers to adopt Ridge and Lasso Regression models for regularization. For instance, Sharma et al. (2024) demonstrated that Lasso Regression outperformed Ridge. Furthermore, studies comparing traditional regression models and K-Nearest Neighbors (KNN) have highlighted KNN's superior performance in housing price prediction (Kanadiya & Chawan, 2024). These findings guide our exploration of five primary models—KNN, XGBoost, Random Forest, Lasso Regression, and Ridge Regression.

2. Exploratory Data Analysis

To address the significant number of missing values in the dataset, I treated missing categorical features as 'unknown' and retained missing numerical features. Figure 1 presents the distribution of final close prices on a logarithmic scale to visualize the wide range of values better. The majority of properties cluster between $100,000 and $10,000,000, forming a bell-shaped distribution. A small fraction of properties, likely representing luxury estates or low-cost housing, fall outside this range, either below $10,000 or above $100,000,000.

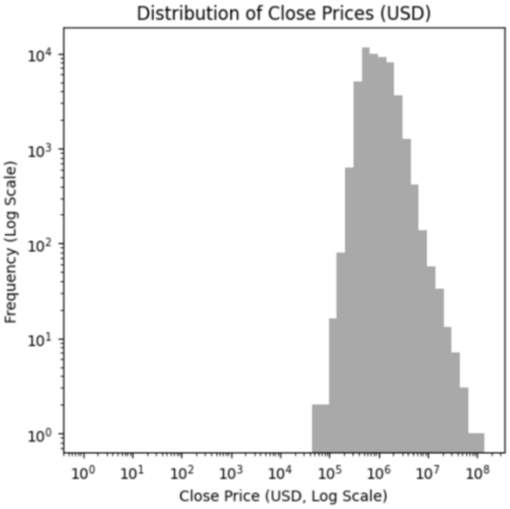

Figure 1 Distribution of Close Prices (USD)

Figure 2 visualizes the distribution of housing construction by year, with the vertical axis on a logarithmic scale. The red lines mark significant economic and societal events: the 2008 financial crisis and the 2019 COVID-19 pandemic. Following the 2008 financial crisis and the 2019 pandemic, there has been a clear decline in housing construction, highlighting the profound impact of events on the real estate market. Housing construction patterns closely mirror economic and societal changes.

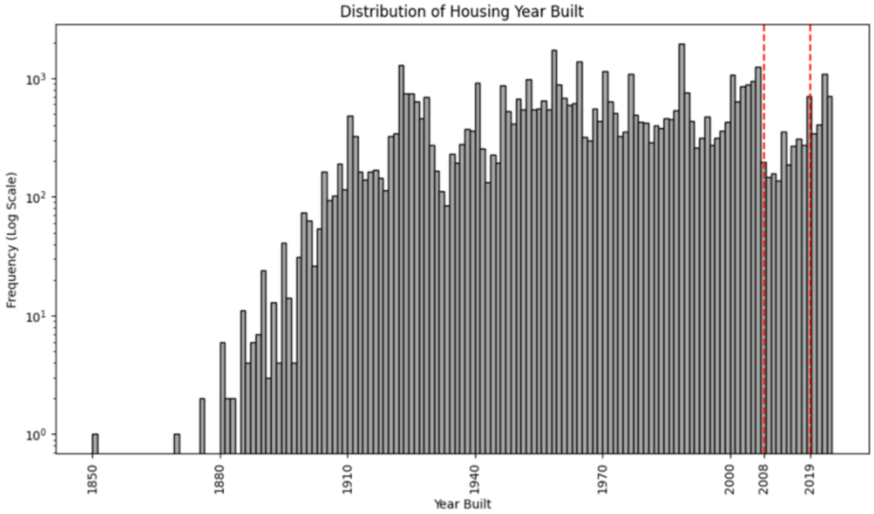

Figure 2 Distribution of Housing Year Built

Figure 3 illustrates the distribution of construction years for buildings across various cities, highlighting notable differences in housing characteristics. Cities such as Victorville, Murrieta, and Menifee exhibit more recent construction trends, reflecting modern development patterns and a relatively younger housing stock. In contrast, Los Angeles and Oakland display older construction patterns, with significant variability and a substantial number of outliers, indicating the presence of a considerable proportion of older structures.

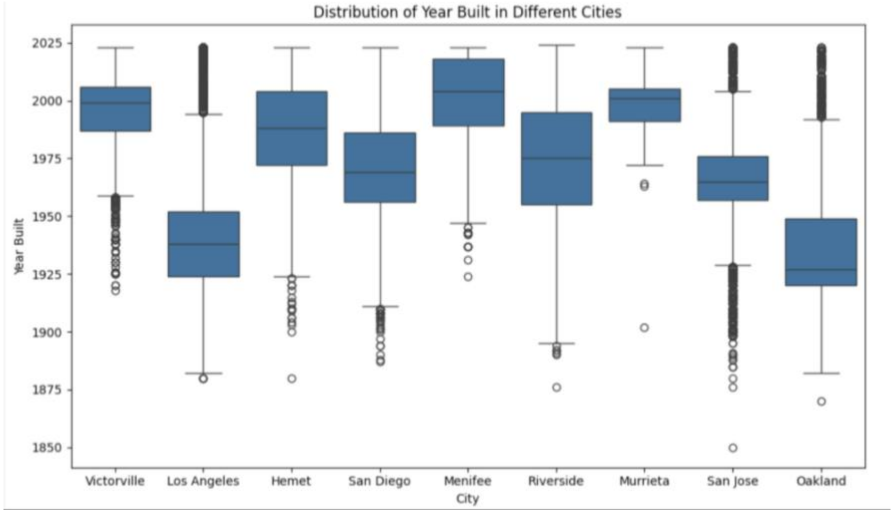

Figure 3 Distribution of Year Built in Different Cities

Figure 4 illustrates the distribution of close prices across various cities, with the vertical axis on a logarithmic scale. The red dashed line represents the mean close price, while the orange solid line indicates the median close price. High-value outliers are present across all cities, particularly in Los Angeles and San Diego, likely corresponding to luxury properties. Low-value outliers are more pronounced in Victorville and Hemet, likely reflecting affordable or suburban housing markets.

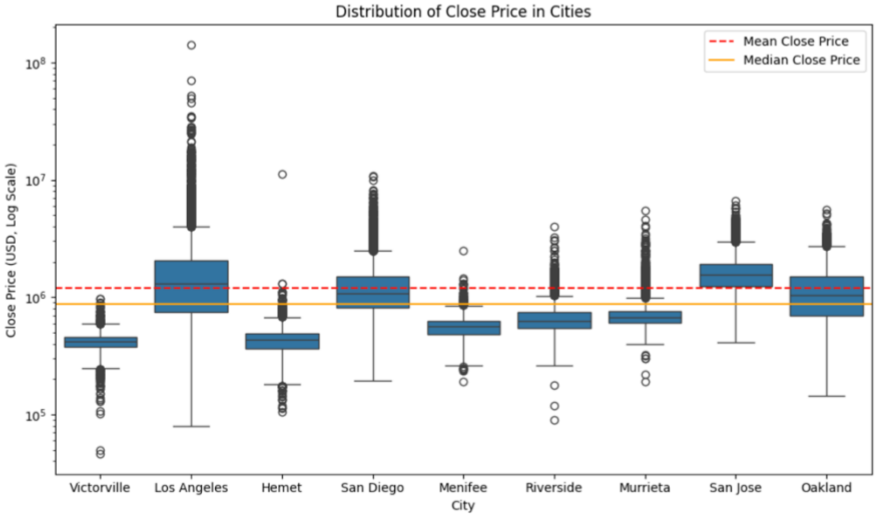

Figure 4 Distribution of Close Prices in Cities

3. Methods

3.1 Feature Selection

Following the exploratory data analysis, I conducted feature selection to enhance model performance. Columns containing only a single value (`StateOrProvince`, `PropertyType`, `PropertySubType`) or with unique values (`ListingKey`) were dropped. Additionally, features with a correlation coefficient greater than 0.9 (`ListPrice` and `Stories`) were removed to mitigate multicollinearity issues.

3.2 Feature Engineering

Recognizing the temporal dependency of property prices, I conducted feature engineering on the `CloseDate` feature. I extracted the `CloseYear` and `CloseMonth` components to explicitly model potential seasonal and yearly trends in property values.

3.3 Data Split

I conducted an initial split of the dataset into training and validation sets and a testing set, using an 80 percent to 20 percent ratio. To address the imbalance and heterogeneity in close prices across different cities, the split was stratified based on the `City` feature. This approach ensured that the proportion of each city in both subsets was representative of the original dataset

3.4 Preprocessing

Categorical features such as \`Flooring\`, \`ViewYN\`, \`Levels\`, \`PoolPrivateYN\`, \`AttachedGarageYN,\` \`BuyerAgencyCompensationType\`, \`City\`, \`FireplaceYN\`, \`NewConstructionYN\`, and \`PostalCode\` were encoded using one-hot encoding to convert them into numerical representations suitable for machine learning algorithms. For ordinal categorical features such as \`AssociationFeeFrequency\`, \`YearBuilt\`, \`CloseMonth\` and \`CloseYear\`, which exhibit a natural ordering, ordinal encoding was applied. Numerical features, including \`Latitude\`, \`Longitude\`, \`LivingArea\`, \`DaysOnMarket\`, \`LotSizeAcres\`, \`BathroomsTotalInteger\`, and \`BedroomsTotal\` were standardized using a standard scaler.

4. Machine Learning Pipeline

To address missing values in the numerical features \`LivingArea\`, \`LotSizeAcres\`, \`BathroomsTotalInteger\`, and \`YearBuilt\`, I employed multiple imputations using Random Forest as the imputation model. This process was repeated three times to ensure stability. For reliable model evaluation, I utilized a stratified K-fold cross-validation technique, stratifying on the City feature to ensure that each fold was representative of the overall dataset regarding city distribution.

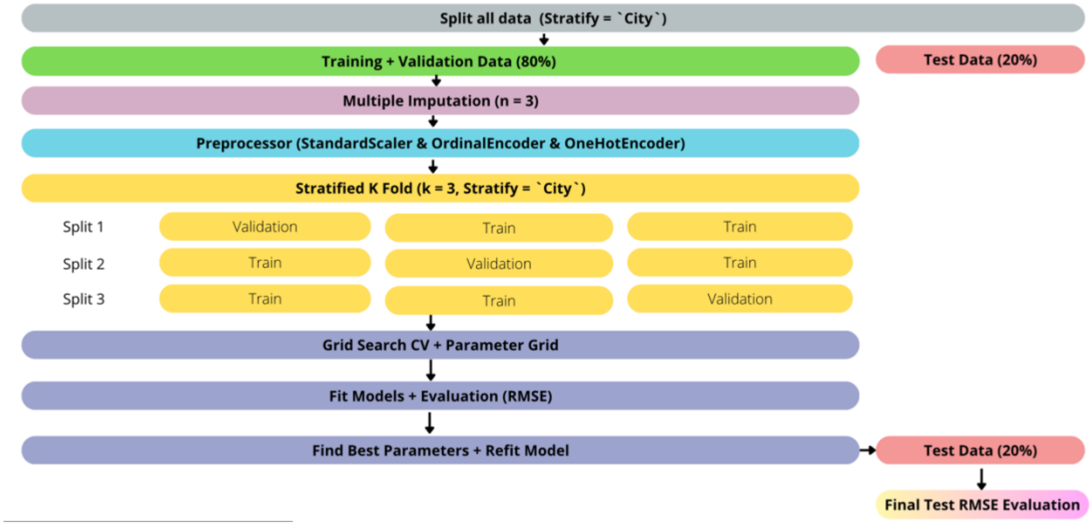
Split all data (Stratify = \`City\`)
Training + Validation Data (80%)
Test Data (20%)
Multiple Imputation (n = 3)
Preprocessor (StandardScaler & OrdinalEncoder & OneHotEncoder)
Stratified K Fold (k = 3, Stratify = \`City\`)
Split 1
Validation
Train
Train
Split 2
Train
Validation
Train
Split 3
Train
Train
Validation
Grid Search CV + Parameter Grid
Fit Models + Evaluation (RMSE)
Find Best Parameters + Refit Model
Test Data (20%)
Final Test RMSE Evaluation

Figure 5 Machine Learning Pipeline Overview

During the model training phase, five machine learning algorithms were evaluated: KNN, Lasso, Ridge, Random Forest, and XGBoost. Hyperparameter tuning for each model was performed using GridSearchCV to identify the optimal parameters. Table 1 outlines the parameter ranges explored for each algorithm, with the best-performing parameters highlighted in bold.

Model	Parameter	Range

Lasso	model__alpha	[0.01, 0.1, 1.0, 10.0, **100.0**]
Ridge	model__alpha	[0.01, 0.1, **1.0**, 10.0, 100.0]
KNN	model__n_neighbors model__weights	[1, **5**, 10, 100] ['uniform', **'distance'**]
XGBoost	model__reg_alpha model__reg_lambda model__max_depth	[0e0, 1e-2, 1e-1, 1e0, 1e1, **1e2**] [0e0, 1e-2, **1e-1**, 1e0, 1e1, 1e2] [1,3,**10**,30,100]
Random Forest	model__max_depth model__max_features	[1, 3, 10, **20**] [0.25, **0.5**, 0.75, 1.0]

Table 1 Hyperparameter Ranges and Optimal Values for Each Model

To evaluate the performance of our models, I adopted the root mean squared error (RMSE) as the primary metric. RMSE was chosen primarily because its unit matches that of the target variable making the results more interpretable. Additionally, RMSE is particularly effective in penalizing larger errors, which aligns with the objective of the study: minimizing significant deviations in price predictions that could have substantial implications in practical applications. The final evaluation was conducted on the test set, where the best model from each imputation strategy was applied, and the RMSE was computed.

5. Results

5.1 Test RMSE Results

Figure 6 presents the mean test RMSE for five machine learning algorithms. The red dashed line represents the baseline RMSE, calculated as the mean of the test set target values, serving as a benchmark for model performance. KNN exhibits the highest mean RMSE among the algorithms, reflecting the weakest performance in this context. In contrast, XGBoost and Random Forest achieve the lowest and second-lowest RMSE values, respectively, demonstrating superior predictive performance for this dataset.

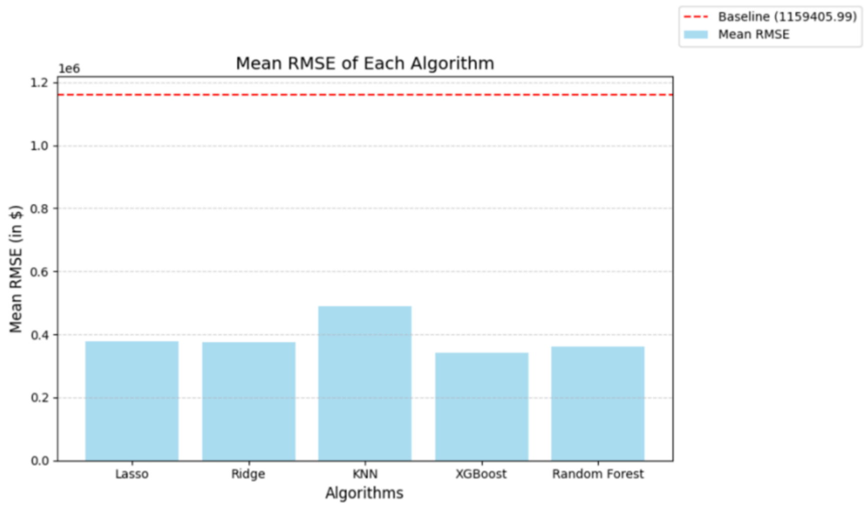

Figure 6 Mean RMSE of Each Algorithm

Figure 7 presents the standard deviation of RMSE for each machine-learning algorithm. Among the two models with the lowest mean RMSE, XGBoost is the best performer. Although KNN demonstrates remarkable stability with the lowest variability in RMSE, its mean RMSE is the highest, indicating weaker overall accuracy. Balancing both accuracy and consistency, XGBoost emerges as the optimal choice. Consequently, XGBoost was selected as the best model, with the following hyperparameters: reg_alpha = 100.0, reg_lambda = 0.1, and max_depth = 10.

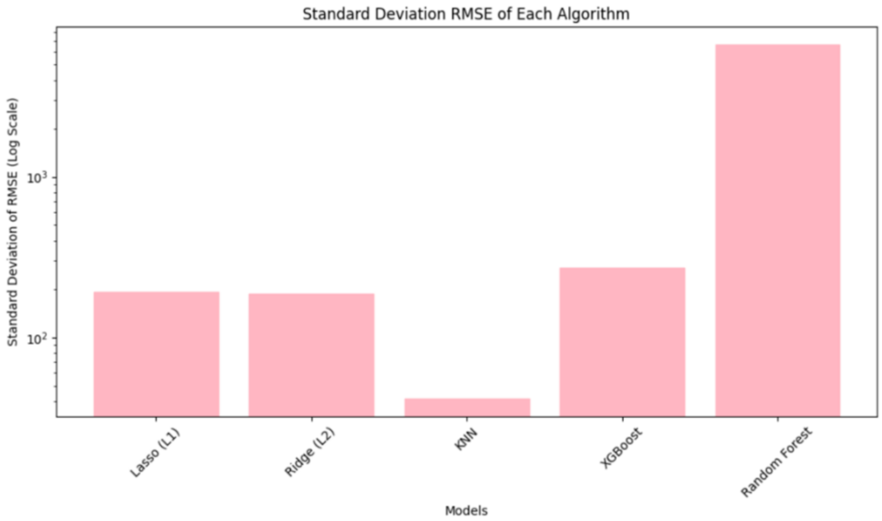

Figure 7 Standard Deviation of RMSE For Each Algorithm

5.2 Global Feature Importance

Having identified XGBoost as the optimal model due to its superior balance of accuracy and consistency, this section proceeds to analyze the global feature importance.

5.2.1 Permutation Importance

Figure 8 highlights the top five important features influencing the performance of the optimal XGBoost model, as determined by permutation importance. Among these features, location-related attributes dominate underscoring the significance of geographical factors in predicting housing prices. Additionally, property size attributes, including Living Area and Lot Size Acres, further emphasize the importance of the physical characteristics of a property in determining its market value. These findings align with the broader understanding of real estate markets, where both location and property size play pivotal roles.

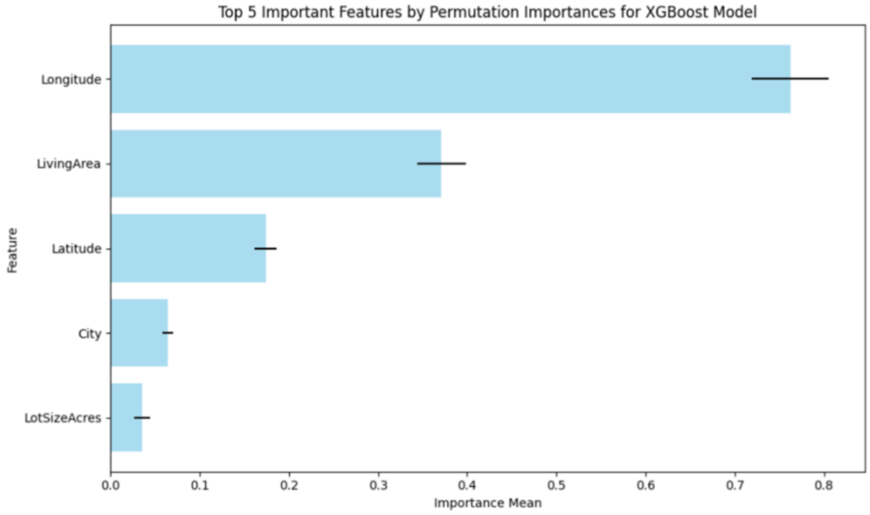

Figure 8 Top 5 important Features by Permutation Importances for XGBoost

5.2.2 Feature Importance by XGBoost

Figure 9 ranks the top five features by their weight values, indicating the frequency with which each feature is used in splits across all trees in the XGBoost model. Among these, the Living Area is the most frequently utilized feature, further reinforcing the critical role of property size. The prominence of Lot Size Acres, Latitude, and Longitude also highlights the importance of geographic location, making these two factors—property size and location—the primary determinants in the model's decision-making process. Additionally, Days on the Market appears as a significant feature, suggesting that the length of time a property remains on the market also plays a role in shaping price predictions.

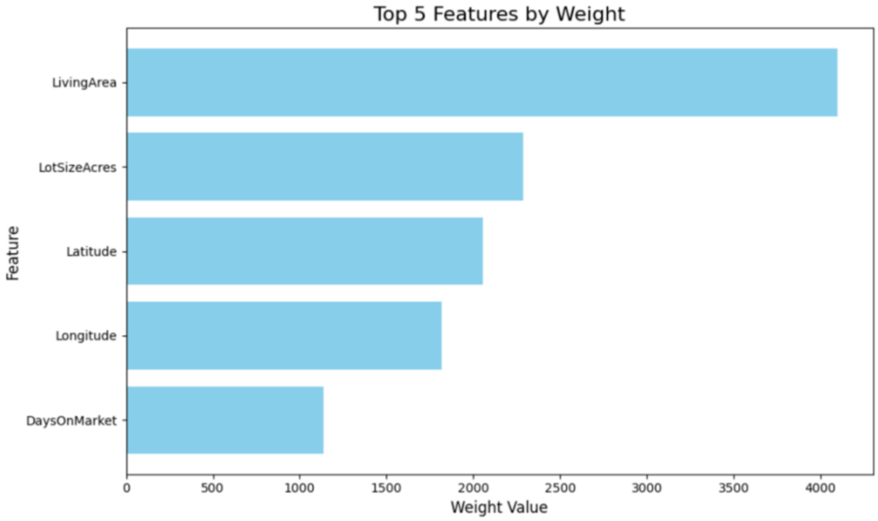

Figure 9 Top 5 Features by XGBoost – Weight

Figure 10 shifts the focus to the top five features ranked by their total gain values, which reflect the cumulative improvement in accuracy from splits involving each feature. While Living Area continues to dominate as the most impactful feature, Bathrooms Total Integer stands out as a key factor, representing another dimension of property size. Together, these results emphasize the primary influence of property size and geographic location while also revealing the added contributions of temporal and detailed structural attributes, such as Days on Market and Bathrooms Total Integer, in enhancing model accuracy.

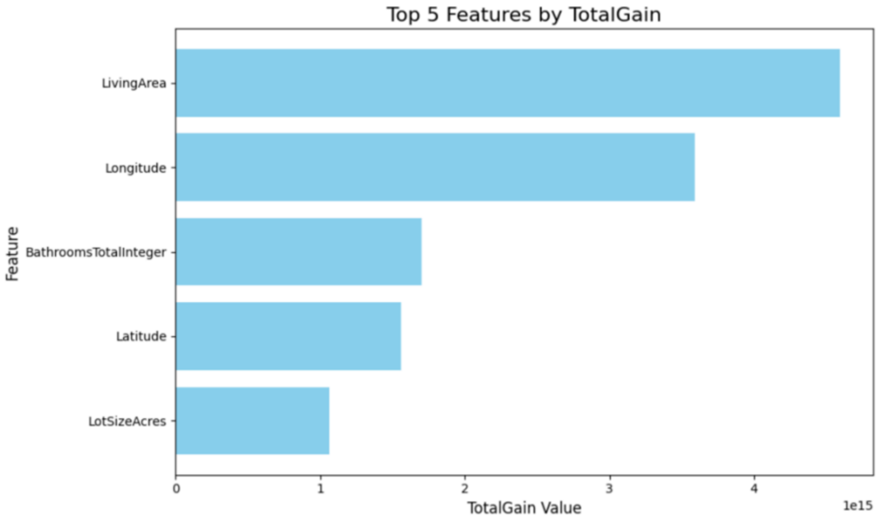

Figure 10 Top 5 Features by XGBoost - Total Gain Values

5.2.3 SHAP

Figure 11 displays the top five important features for the XGBoost model as determined by SHAP values, which measure each feature's contribution to the predictions. Property size plays a key role, emphasizing that larger properties tend to command higher prices. Additionally, `PostalCode_90049` and `PostalCode_90039` emerge as influential features, highlighting the significant impact of specific geographic regions on housing prices. However, over-reliance on zipcode-specific features risks reducing model robustness, as such models may struggle to generalize to unseen regions or different geographic contexts.

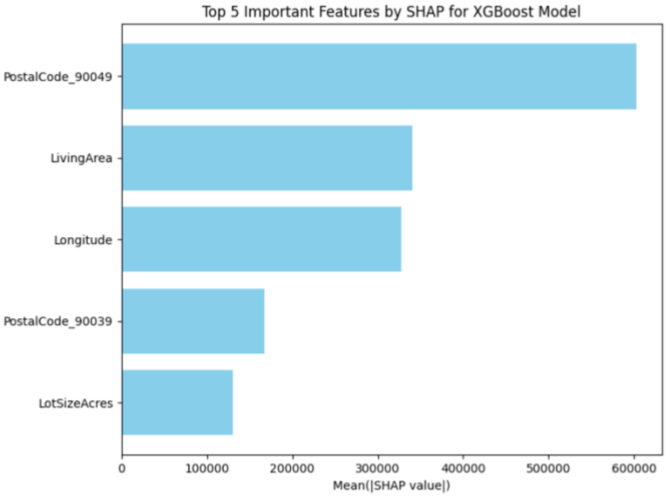

Figure 11 Top 5 Important Features by SHAP

5.3 Local Feature Importance

Figure 12 illustrates the contributions of various features to a specific prediction made by the XGBoost model, which results in a final predicted price of $3,281,659.22—higher than the baseline value. The largest positive contributors are `PostalCode_90049`, `LivingArea`, and `Longitude`, emphasizing the critical roles of geographic location and property size in driving the higher predicted price. In contrast, `LotSizeAcres` has a negative contribution of -0.11, suggesting that the property has a relatively smaller lot size compared to others, which slightly reduces the predicted value.

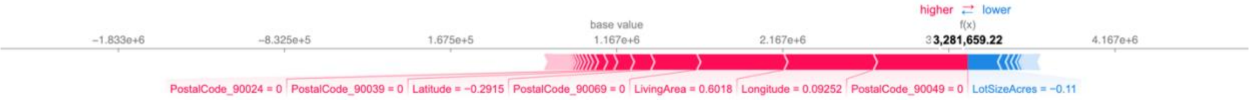

Figure 12 SHAP Feature Contributions for a Single Prediction

6. Outlook

The findings of this project for Random Forest and XGBoost models align with existing research, such as Truong, Nguyen, Dang, & Mei (2020). However, there is ample room for further enhancement. A key challenge faced during this project was the computational limitations, with the machine learning pipeline taking approximately six hours to complete. In the future, I plan to increase the k-fold cross-validation iterations from three to ten, which will improve the stability and robustness of the results. Furthermore, I aim to explore advanced modeling approaches, such as deep learning techniques, to better capture the complexities and nuances of the housing market. These improvements will not only refine predictive accuracy but also provide deeper insights into market dynamics.

References

California Regional Multiple Listing Service. (n.d.). *California Regional Multiple Listing Service*. Retrieved from https://go.crmls.org

RESO Data Dictionary Wiki. (n.d.). *Property Resource*. Retrieved from https://ddwiki.reso.org/display/DDW20/Property+Resource

Phan, T. D. (2019). Housing price prediction using machine learning algorithms: the case of Melbourne city, Australia. In P. K. Rhee, D. Howard, & M. R. Bashar (Eds.), *Proceedings International Conference on Machine Learning and Data Engineering: iCMLDE 2018* (pp. 35-42). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/iCMLDE.2018.00017

Quang Truong, Minh Nguyen, Hy Dang, Bo Mei, Housing Price Prediction via Improved Machine Learning Techniques, Procedia Computer Science, Volume 174, 2020, Pages 433-442, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.06.111.

M. Sharma, R. Chauhan, S. Devliyal and K. R. Chythanya, "House Price Prediction Using Linear and Lasso Regression," *2024 3rd International Conference for Innovation in Technology (INOCON)*, Bangalore, India, 2024, pp. 1-5, doi: 10.1109/INOCON60754.2024.10511592.

Kanadiya, P., & Chawan, P. M. (2024). A KNN-Linear Regression Fusion Approach for Improved Real Estate Price Estimation. *International Research Journal of Engineering and Technology (IRJET)*, *11*(8).