Regression-Based Approach for Accurate House Price Forecasting

Md Fahimul Kabir Chowdhury[1], Jayed Mohammad Barek[2]
Department of Computer Science and Engineering
University of North Texas, Denton, TX, USA
{mdfahimulkabirchowdhury, jayedmohammadbarek}@my.unt.edu

April 24, 2025

Abstract

This project focuses on predicting median housing prices in California using machine learning techniques. By leveraging the California Housing dataset, we explored key features such as income levels, room counts, population density, and geographic coordinates to model real estate values. Our approach included thorough data preprocessing, feature engineering, and experimentation with multiple regression algorithms. We began with Linear Regression as a baseline model and then implemented a Random Forest Regressor to capture more complex patterns in the data. Further improvements were achieved through hyperparameter tuning using GridSearchCV. Evaluation metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2) were used to assess performance. The final tuned Random Forest model delivered strong predictive results, achieving an R^2 score above 0.80, demonstrating its effectiveness in modeling housing prices. Overall, the project highlights the importance of feature design, model selection, and tuning in building reliable prediction systems for real-world data.

1 Introduction

Predicting housing prices accurately is an important task in areas like real estate, city planning, and finance. Traditional methods, which often rely on simple formulas or expert rules, may not fully capture the complicated patterns in housing data. In recent years, machine learning has provided more powerful tools that can handle this complexity and improve prediction accuracy. The main goal of this project is to build a machine learning model that can estimate housing prices in California based on data from the U.S. Census. We use features such as median income, average number of rooms, age of houses, and geographic coordinates (like latitude and longitude). The goal is to create a model that not only predicts prices well but can also help support smart decisions in real estate and public planning.

This project is useful for many groups, including real estate agents, investors, urban planners, and government officials. Accurate price predictions can reveal trends in the housing market, help assess housing affordability, and guide investment and planning decisions, especially in high-demand areas like California. Recent research has shown that machine learning models such as Gradient Boosting, XGBoost, and Deep Neural Networks are more effective than traditional methods when it comes to predicting house prices. For instance, [1] showed that XGBoost works well on large, complex datasets. Other studies, like [2], found that using location-based features (such as coordinates) can greatly improve prediction accuracy. Real-world platforms like Zillow also use machine learning to estimate home values, showing how useful these models are in industry. Additionally, recent research by [3] emphasizes the importance of making models transparent and fair-especially when they are used in sensitive areas like housing and finance. A more recent study by [4] explored different machine learning models for predicting home prices in Seattle. The study showed that combining multiple models into one (a technique called Stacking) gave better results than using a single model.

In this project, we follow a similar approach shown in (Figure 1). Using the California Housing dataset, we go through the full machine learning pipeline-data cleaning, feature engineering, model building, and evaluation. Our aim is to build a model that is both accurate and easy to understand, and that can help people make better decisions in the housing market.

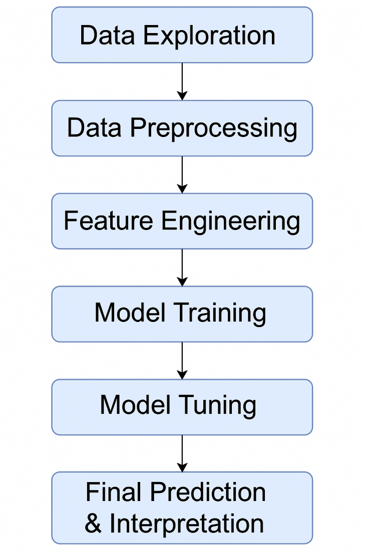

Figure 1: Outflow of Research

2 Data Exploration and Preprocessing

We began our project by exploring the structure and contents of the California Housing dataset. The dataset contains 20,640 entries and 9 numerical features, including variables such as median income, average number of rooms, population, and geographic location. Our first step was to check for missing values, data types, and any obvious inconsistencies. Fortunately, the dataset was clean, with no missing values or data formatting issues.

Dataset Overview

The California Housing dataset contains detailed housing information for various block groups across California. A summary of the dataset is as follows:

- **Total Entries:** 20,640 rows
- **Total Features:** 9 (including the target variable `MedHouseVal`)
- **Data Type:** Numeric (`float64`) for all features
- **Missing Values:** None identified

Table 1 presents the first five rows of the dataset, offering a snapshot of the feature values.

MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedHouseVal
8.3252	41.0	6.9841	1.0238	322.0	2.5556	37.88	-122.23	4.526
8.3014	21.0	6.2381	0.9719	2401.0	2.1098	37.86	-122.22	3.585
7.2574	52.0	8.2881	1.0734	496.0	2.8023	37.85	-122.24	3.521
5.6431	52.0	5.8174	1.0731	558.0	2.5479	37.85	-122.25	3.413
3.8462	52.0	6.2819	1.0811	565.0	2.1815	37.85	-122.25	3.422

Table 1: First five rows of the California Housing dataset.

Descriptive Statistics

The dataset contains significant variability across features. Table 2 provides a statistical summary of each numerical variable, including count, mean, standard deviation, and percentile values.

max width=

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedHouseVal
Count	20640	20640	20640	20640	20640	20640	20640	20640	20640
Mean	3.8707	28.64	5.4290	1.0967	1425.5	3.0707	35.63	-119.57	2.0686
Std	1.8998	12.59	2.4742	0.4739	1132.5	10.3861	2.1360	2.0035	1.1540
Min	0.4999	1.0	0.8462	0.3333	3	0.6923	32.54	-124.35	0.1500
25%	2.5634	18.0	4.4407	1.0061	787	2.4297	33.93	-121.80	1.1960
50%	3.5348	29.0	5.2291	1.0488	1166	2.8181	34.26	-118.49	1.7970
75%	4.7433	37.0	6.0524	1.0995	1725	3.2823	37.71	-118.01	2.6473
Max	15.0001	52.0	141.91	34.07	35682	1243.33	41.95	-114.31	5.0000

Table 2: Descriptive statistics of the California Housing dataset.

Insights from Data Exploration

- **Median Income (MedInc):** Ranges from 0.4999 to 15.0001, showing considerable income diversity across neighborhoods.
- **Population:** Varies from 3 to 35,682, indicating both urban and rural regions are represented.
- **Median House Value (MedHouseVal):** Capped at 5.0, suggesting a potential ceiling effect in housing value reporting.

Visual Exploration

To further understand the relationships and distributions within the data, we conducted several visual analyses:

- **Correlation Heatmap:** Showed a strong positive correlation between `MedInc` and `MedHouseVal`, while most other features had weaker or no linear correlations.
- **Geographic Scatterplot:** A plot of latitude and longitude against house values revealed that high-value homes are generally concentrated along the coastal regions of California.
- **Histograms:** Highlighted skewness in several features such as `MedInc`, `AveRooms`, and `Population`, helping guide our preprocessing decisions like scaling and feature transformation.

Overall, this phase gave us a solid understanding of the dataset's structure, variability, and the patterns likely to influence housing prices. It laid the groundwork for informed preprocessing and model development in the later stages of the project.

Data Preprocessing

In this stage, we prepared the dataset for modeling by engineering new features, splitting the data, and applying standardization to bring all features onto a consistent scale. These steps were critical in ensuring the model could interpret the data accurately and avoid bias caused by feature magnitude differences.

Data Splitting and Scaling

- The dataset was split into a training set (80%) and a test set (20%) to evaluate generalization.
- We applied feature scaling using `StandardScaler` to standardize the input values, ensuring that no single feature dominated due to its magnitude.

Correlation Heatmap

To explore relationships between variables, we generated a correlation matrix heatmap (Figure 2). The heatmap reveals a strong positive correlation between `MedInc` (median income) and the target variable `MedHouseVal`, suggesting income is a key driver of housing prices. Other features, such as `Latitude` and `AveRooms`, showed moderate correlations.

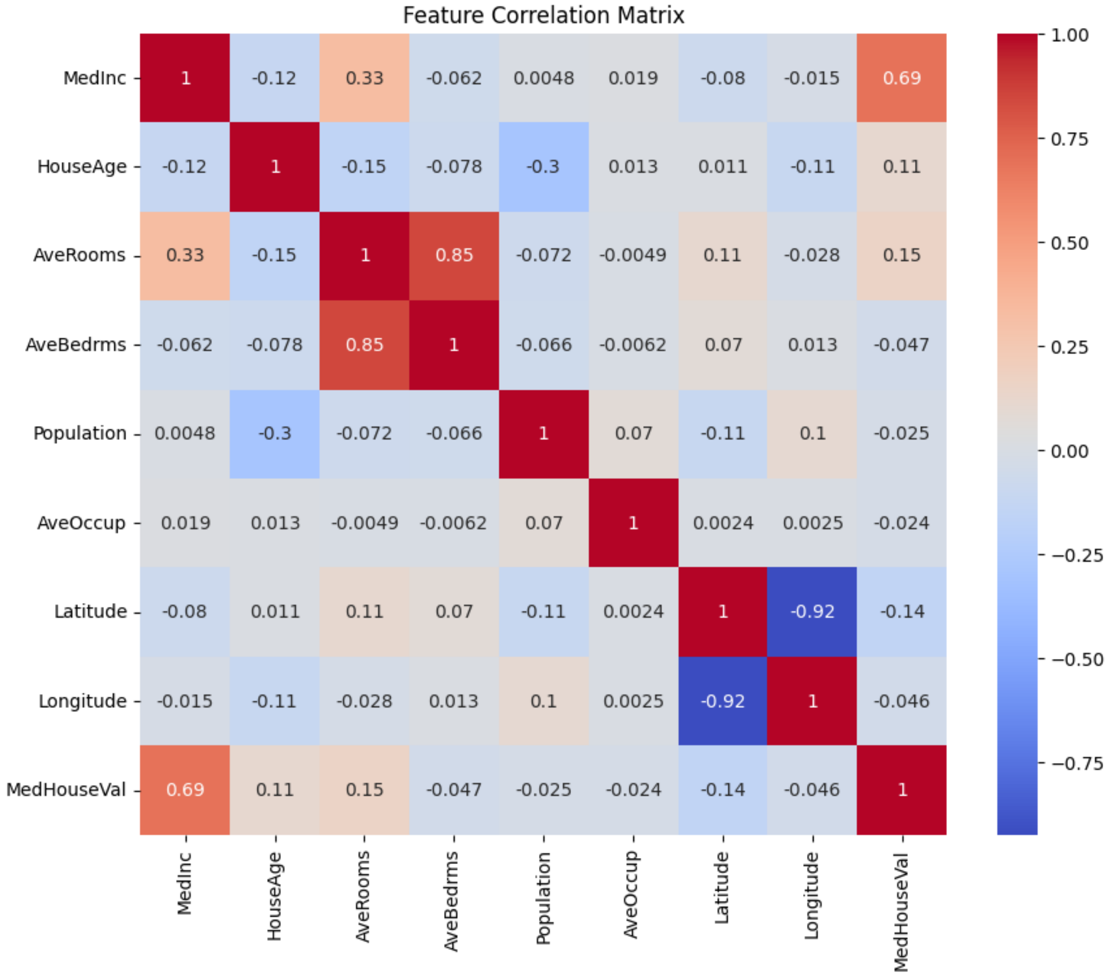

Figure 2: Feature Correlation Matrix showing relationships between predictors and the target variable. Strong positive and negative correlations are visually emphasized.

Geographic Scatterplot

A color-coded scatterplot using geographic coordinates (Figure 3) highlights regional price variations across California. Higher house values were predominantly located in coastal areas, particularly around the San Francisco Bay Area and Los Angeles. This visualization emphasized the importance of geographic features such as latitude and longitude in price prediction.

Histograms

Histograms were plotted for all features to examine their distributions (Figure 4). Some features, such as `MedInc`, `AveRooms`, and `Population`, exhibited skewness. These insights informed our decision to apply scaling and consider feature transformations if necessary for more advanced modeling.

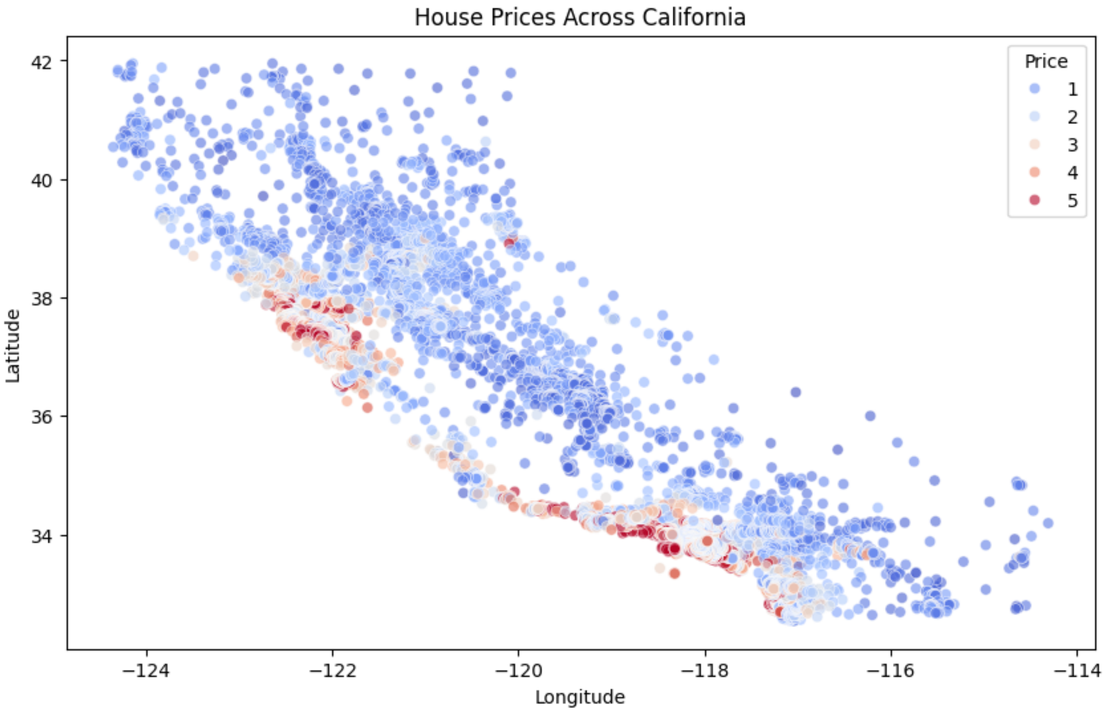

Figure 3: Geographic scatterplot of house prices across California. High-value areas appear concentrated near the coast.

Output Interpretation

- The correlation matrix revealed that median income is the most influential feature in predicting housing prices.
- The geographic scatterplot confirmed strong spatial patterns in housing value distribution, especially favoring coastal regions.
- Histograms highlighted feature skewness and variability, reinforcing the need for preprocessing and scaling.
- Engineered features such as `RoomsPerHousehold` and `PopulationPerHousehold` added meaningful context to the original dataset and helped improve model learning.
- Feature scaling ensured consistent input ranges across features, which is especially important for algorithms sensitive to input scale.

Together, these steps ensured that our data was well-prepared for training machine learning models and that we had a deeper understanding of the relationships and patterns present in the dataset.

3 Feature Engineering

In our project, we started with the original features provided by the California Housing dataset, such as average rooms, bedrooms, population, and occupancy per household. While these features were informative, we believed that creating a few additional features could help the model capture deeper patterns in the data. One of the new features we added was RoomsPerHousehold, which was calculated by dividing the average number of rooms by the average number of occupants. This gave us a better sense of space available per household and helped the model understand the living conditions in each area more clearly.

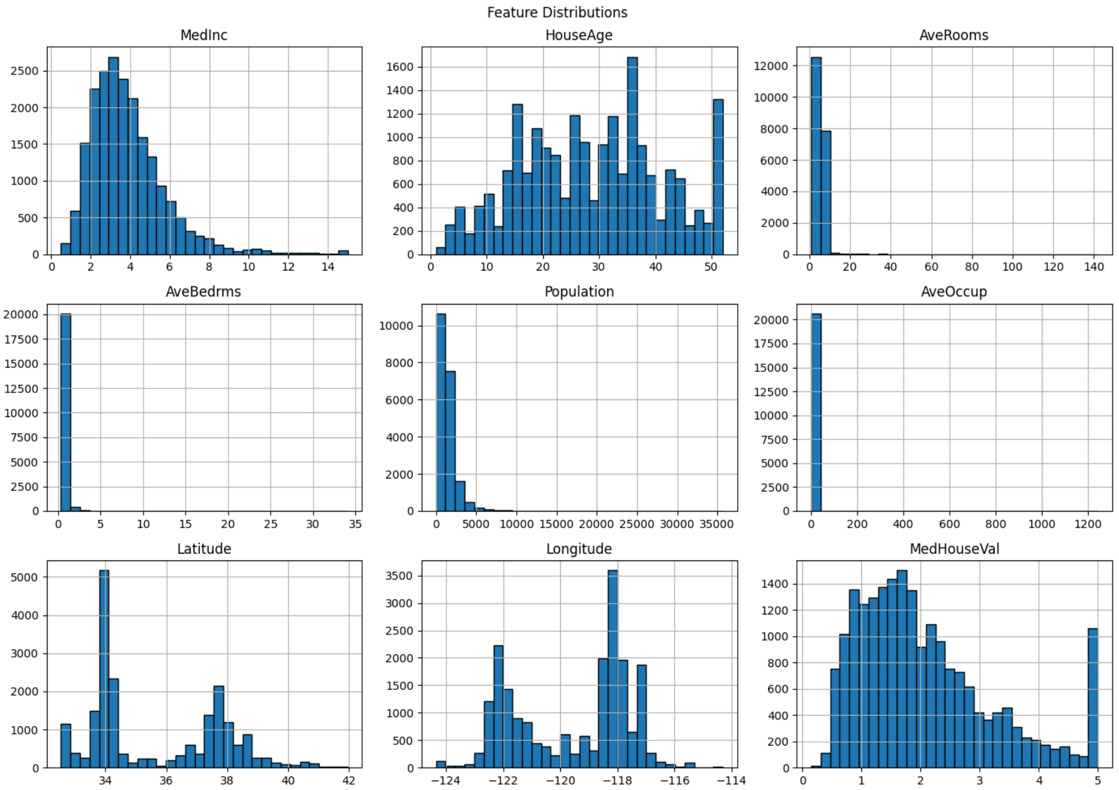

Figure 4: Histograms of key features showing skewed distributions and value ranges.

We also introduced BedroomsPerRoom, which showed the proportion of bedrooms compared to total rooms. This ratio helped us capture how homes are structured-whether they tend to be large, multi-room houses or smaller spaces with fewer bedrooms. Finally, we added PopulationPerHousehold, which gave us an idea of how crowded an area might be. Areas with a high number of people per household could reflect different housing demands or social patterns that impact home prices.

To enrich the dataset with more informative variables, we created the following engineered features:

- `RoomsPerHousehold` = `AveRooms` / `AveOccup`
  Reflects the average room availability per household, helping capture housing density.
- `BedroomsPerRoom` = `AveBedrms` / `AveRooms`
  Indicates how much of the housing area is dedicated to bedrooms, giving insight into house structure.
- `PopulationPerHousehold` = `Population` / `AveOccup`
  Measures population density within households, which can relate to housing affordability and size.

After adding these features, we noticed that the model's performance improved, especially with tree-based algorithms like Random Forest. These new features provided extra insights that the model couldn't easily extract from the original features alone. By engineering these variables, we were able to give the model more meaningful information to work with, leading to better predictions and stronger overall performance.

4 Model Building and Evaluation

We focused on building a model that could accurately predict housing prices based on the available features. To do that, we tried out a couple of different regression algorithms and compared their performance. Our main goal was to find a model that balances both accuracy and generalizability.

We started with **Linear Regression** as a baseline. It is a straightforward model that gives us a good starting point to see how well a simple approach can perform. However, housing prices often depend on non-linear relationships and complex patterns, so we also brought in a **Random Forest Regressor**, which is an ensemble learning method that tends to perform well with structured data like this. Once we trained both models on the scaled training data, we evaluated them using several standard metrics:

- **Mean Squared Error (MSE)** to measure the average squared difference between predicted and actual prices,
- **Mean Absolute Error (MAE)** for more interpretable average errors, and
- **R-squared ($\mathbf{R^2}$)** to understand how well the model explains the variance in housing prices.

Not surprisingly, the Random Forest model performed better than Linear Regression. It was able to capture more of the hidden relationships in the data and produced more accurate predictions. After confirming its initial performance, we chose Random Forest as our main model and moved forward with tuning it to make it even better.

Overall, this phase helped us understand which algorithms worked best with our data and how we could refine them to achieve reliable results. By comparing models with consistent evaluation metrics, we were able to confidently select the one that gave us the strongest performance.

5 Results and Analysis

After evaluating different models, we observed noticeable differences in their predictive performance. The baseline Linear Regression model offered a decent starting point but was limited in capturing complex, non-linear relationships in the housing data. In contrast, the Random Forest Regressor significantly outperformed the baseline in all evaluation metrics.

To further refine the performance, we applied hyperparameter tuning using GridSearchCV. The tuning process tested 36 combinations of hyperparameters using 3-fold cross-validation. The best model had the following configuration:

```
{'n_estimators': 200, 'max_depth': 20, 'min_samples_split': 2, 'min_samples_leaf': 2}
```

Although the improvement after tuning was modest, it confirmed that the original model was close to optimal. The tuned version slightly reduced both MSE and MAE, while achieving the highest R^2 score overall.

Model	**MSE**	**MAE**	$\mathbf{R^2}$
Linear Regression	0.4540	0.4874	0.6535
Random Forest (Untuned)	0.2561	0.3299	0.8046
Random Forest (Tuned)	**0.2549**	**0.3284**	**0.8054**

Table 3: Performance comparison of different regression models

From 3 and 5 shows that tree-based ensemble methods, especially Random Forests, are well-suited for this kind of structured tabular data. With tuning, we achieved a highly accurate model that generalizes well to unseen housing data.

6 Conclusion

In this project, we set out to predict housing prices in California using machine learning. Starting with the California Housing dataset, we explored the data, engineered meaningful new features, and built multiple regression models to estimate median house values. Through experimentation, we found that while Linear Regression provided a solid baseline, it could not capture the complexity of the relationships in the data. Random Forest, on the other hand, significantly improved performance by modeling non-linear patterns

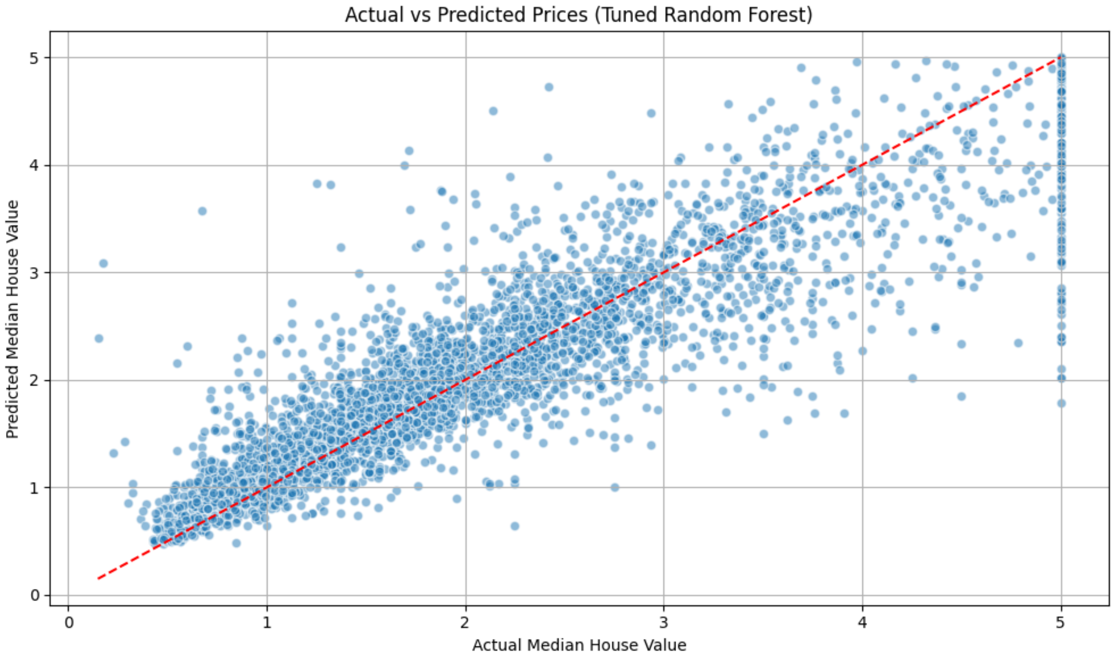

Figure 5: Actual VS Predicted price after model finetuning.

more effectively. We further tuned the Random Forest model using GridSearchCV, which resulted in a small but consistent improvement across all evaluation metrics. Our final model demonstrated strong accuracy and generalization ability, with an R^2 score of over 0.80. This shows that the features we used-along with the modeling choices-were effective in explaining and predicting housing prices. While the performance gain from hyperparameter tuning was modest, it confirmed the stability of the default model and helped reinforce our understanding of how different parameters affect performance. This project not only delivered an accurate predictive model but also gave us hands-on experience with practical machine learning workflows-from preprocessing and feature engineering to model tuning and evaluation.

References

[1] Tianqi Chen and Carlos Guestrin. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.

[2] Nils Kok, Paavo Monkkonen, and John M Quigley. Land use regulations and the value of land and housing: An intra-metropolitan analysis. *Journal of Urban Economics*, 81:136–148, 2014.

[3] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67, 2020.

[4] Jiapei Liao. House price prediction using machine learning: A case study in seattle, us. *ResearchGate*, 2023.