California Housing Price Prediction

A Comparative Analysis of Machine Learning Regression Algorithms

Machine Learning Final Project

Liam Mays

ITCS 3156

UNC Charlotte

1. Introduction

1.1 Problem Statement

This project addresses the challenge of predicting median housing prices in California districts based on various demographic and geographic features. Housing price prediction is a fundamental problem in real estate economics and urban planning, with significant implications for investment decisions, policy-making, and individual homebuyers. The goal is to develop accurate machine learning models that can predict housing values based on observable characteristics of residential areas.

1.2 Motivation and Challenges

Housing affordability is a critical issue affecting millions of people, particularly in high-cost regions like California. Understanding the factors that drive housing prices can help various stakeholders make informed decisions:

- Homebuyers: Identify fair market values and underpriced properties
- Investors: Make data-driven investment decisions
- Policymakers: Develop effective housing and zoning policies
- Urban planners: Understand the impact of infrastructure on property values

The key challenges in this prediction task include: (1) non-linear relationships between features and housing prices, (2) geographic clustering effects where nearby properties have similar values, (3) handling varying scales across different features, and (4) the presence of capped values in the target variable at $500,000.

1.3 Approach Summary

To address these challenges, we employ a comparative analysis of three regression algorithms representing different modeling paradigms: Linear Regression as a baseline parametric model, Random Forest as an ensemble bagging method, and Gradient Boosting as an ensemble boosting method. This multi-model approach allows us to understand which algorithmic paradigm best captures the underlying patterns in housing data and provides insights into the relative importance of different features.

2. Data

2.1 Dataset Introduction

The California Housing dataset is derived from the 1990 U.S. Census and contains information about housing in California districts aggregated at the block group level (Pace and Barry 1997). This dataset is widely used in machine learning education and research due to its appropriate size and complexity.

```
Dataset Characteristics:
```

- Source: StatLib repository (1990 U.S. Census)
```
- Number of samples: 20,640
- Number of features: 8
```
- Target variable: Median House Value ($100,000s)
```
- Missing values: None
```

The eight input features describe characteristics of California census block groups:

```
MedInc : Median income in block group
HouseAge : Median house age in block group
AveRooms : Average number of rooms per household
AveBedrms : Average number of bedrooms per household
Population : Block group population
AveOccup : Average number of household members
Latitude : Block group latitude
Longitude : Block group longitude
```

2.2 Data Visualization and Analysis

Figure 1 shows the distribution of the target variable (median house value). The distribution is right-skewed with a notable spike at $500,000, which represents capped values in the original data. The mean house value is approximately $206,856 and the median is $179,700.

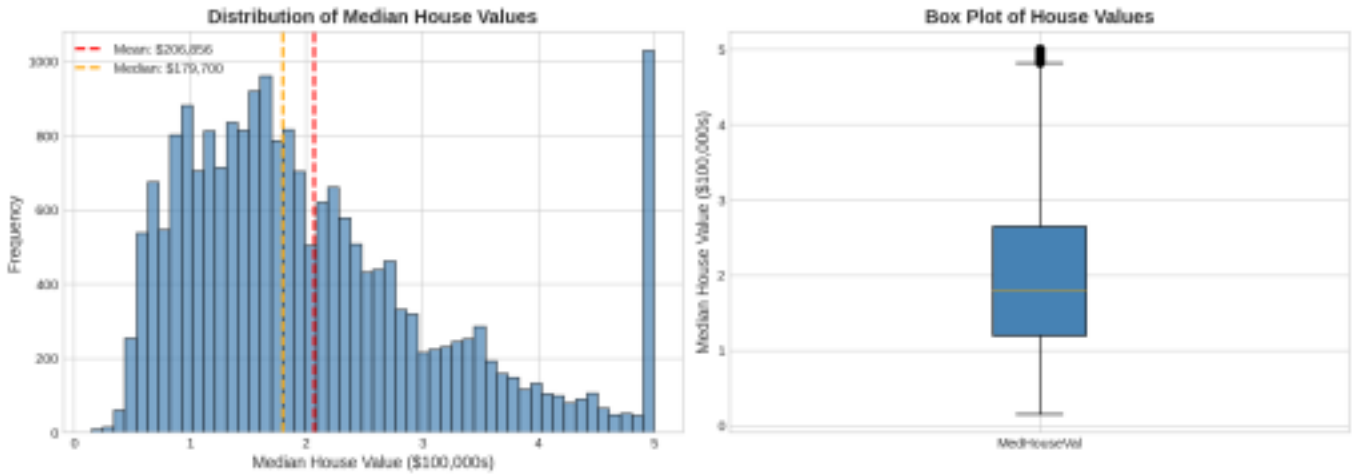

Figure 1: Distribution of Median House Values in California

Figure 2 displays the distributions of all eight features. Notable observations include: MedInc (median income) shows a right-skewed distribution; HouseAge has a relatively uniform distribution with a peak around 50 years; Population and AveOccup contain significant outliers representing densely populated areas.

Figure 2: Distribution of All Features

The correlation heatmap (Figure 3) reveals important relationships between features and the target variable. Median income (MedInc) shows the strongest positive correlation with house values (r = 0.69). Geographic features (Latitude, Longitude) also show significant correlations, reflecting the coastal premium in California housing markets.

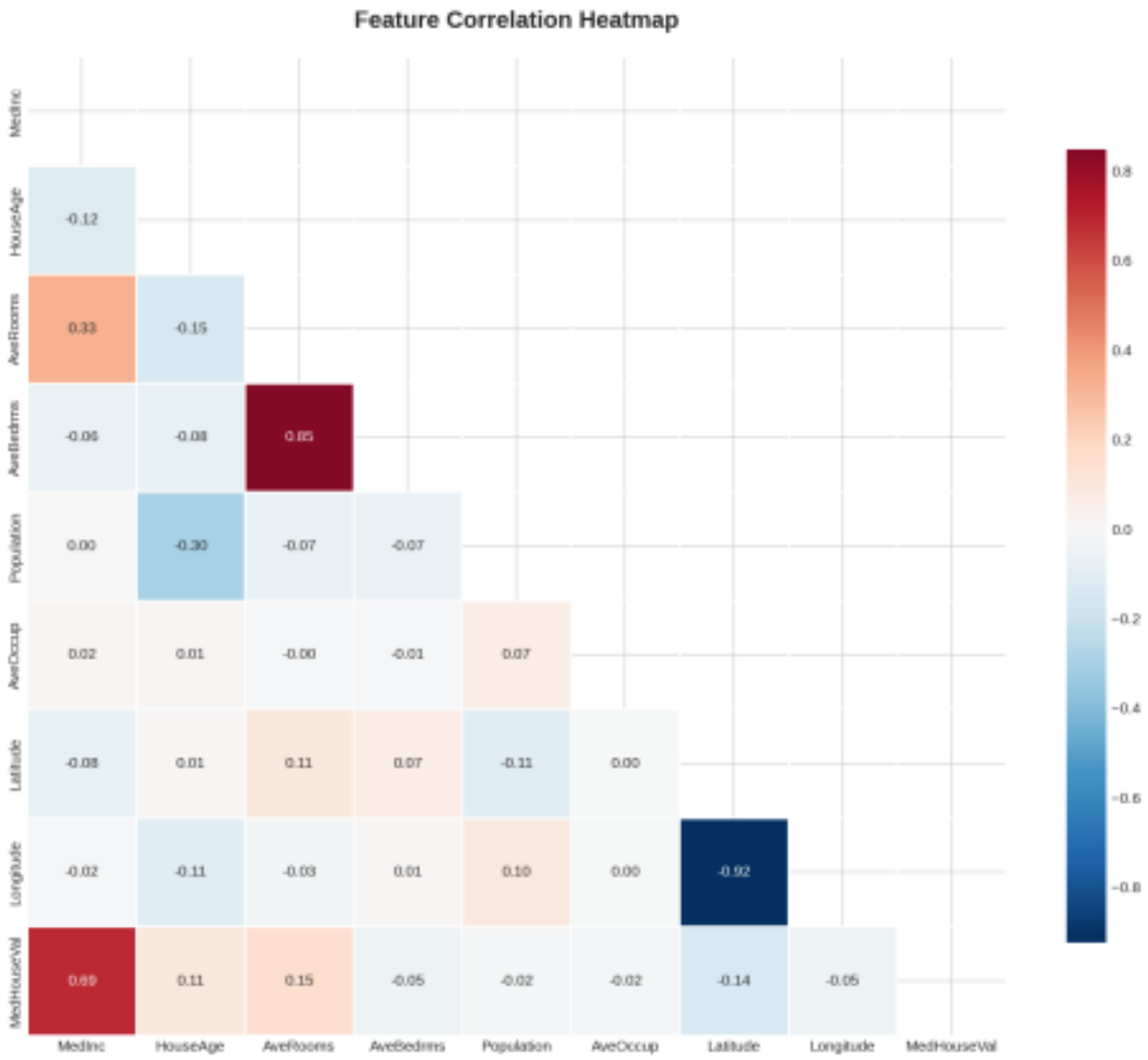

Figure 3: Feature Correlation Heatmap

Figure 4 shows scatter plots of key features against house values. The relationship between median income and house value is clearly positive and approximately linear. Other features show more complex, non-linear relationships that may be better captured by ensemble methods.

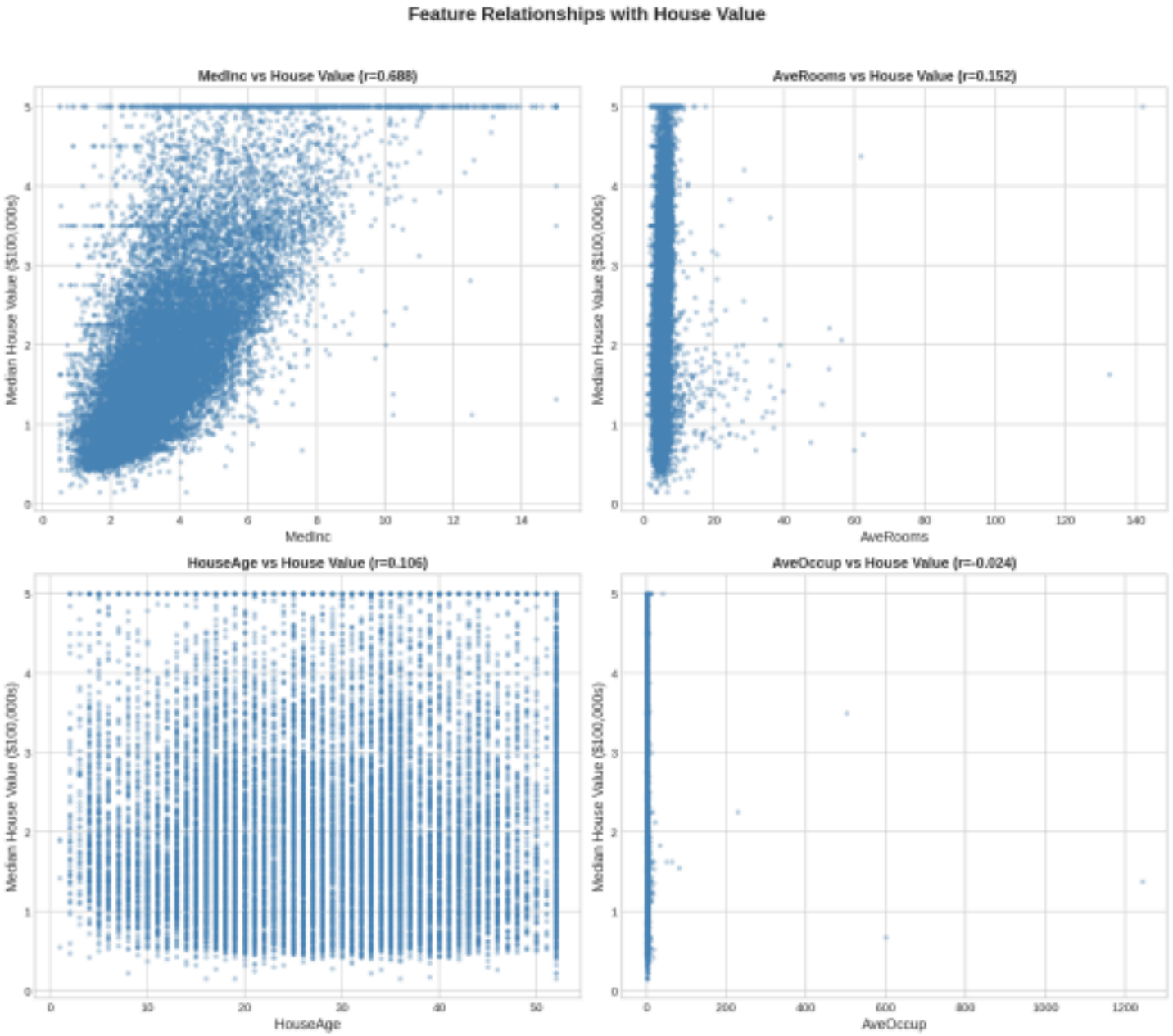

Figure 4: Feature Relationships with House Value

The geographic visualization (Figure 5) clearly shows the spatial patterns in housing prices. Higher values (yellow) are concentrated along the coast, particularly in the San Francisco Bay Area and Los Angeles metropolitan regions. Interior areas show generally lower values (purple).

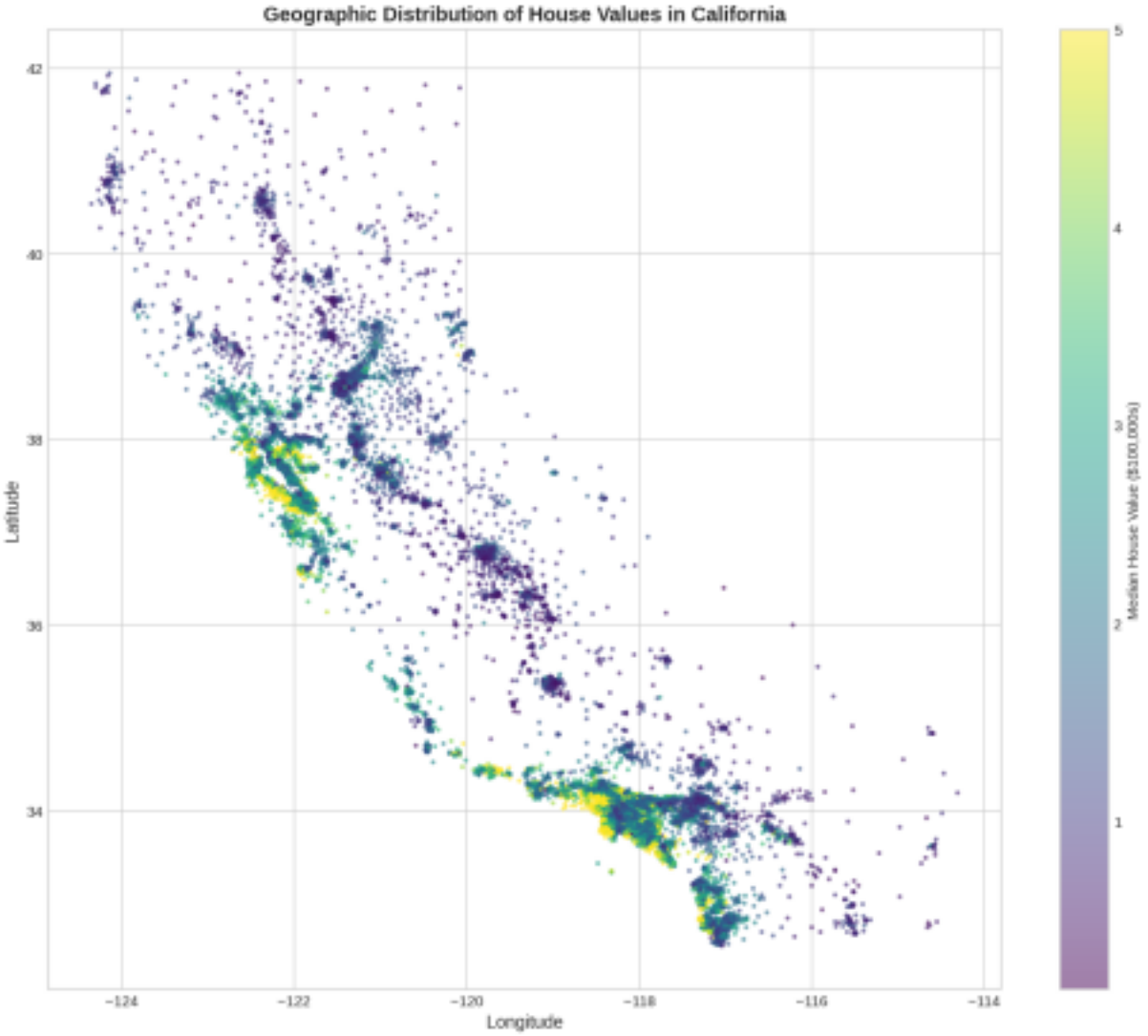

Figure 5: Geographic Distribution of House Values

2.3 Data Preprocessing

Based on the exploratory analysis, the following preprocessing steps were applied:

- Missing Value Handling: Not required as the dataset contains no missing values
- Feature Scaling: StandardScaler applied to normalize features to zero mean and unit variance. This is essential for Linear Regression which is sensitive to feature scales
- Train-Test Split: 80-20 split (16,512 training samples, 4,128 test samples) to evaluate model generalization on unseen data
- Outlier Handling: Outliers were retained as they represent real housing scenarios (luxury homes, densely populated areas)

Note: Tree-based models (Random Forest, Gradient Boosting) were trained on unscaled features as they are inherently scale-invariant due to their decision boundary mechanism.

3. Methods

3.1 Linear Regression

Linear Regression is a fundamental supervised learning algorithm that models the relationship between features and target as a linear combination (Hastie et al. 2009). The model assumes:

```
y = b0 + b1*x1 + b2*x2 + ... + bn*xn + e
```

Where y is the predicted house value, b0 is the intercept, bi are the coefficients for each feature, and e is the error term. The coefficients are estimated using Ordinary Least Squares (OLS), minimizing the sum of squared residuals.

Rationale for Selection:

- Interpretability: Coefficients directly indicate feature importance and direction of effect
- Baseline Performance: Establishes a benchmark for comparing more complex models
- Computational Efficiency: Fast training and prediction, suitable for large datasets
- Statistical Foundation: Well-understood theoretical properties and inference capabilities

Limitations:

- Assumes linear relationships between features and target
- Sensitive to outliers and multicollinearity
- Cannot capture complex non-linear patterns in data

3.2 Random Forest Regressor

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the average prediction of individual trees (Breiman 2001). The algorithm incorporates two key sources of randomness:

- Bootstrap Aggregating (Bagging): Each tree is trained on a random bootstrap sample of the data - Feature Randomization: At each split, only a random subset of features is considered

This randomization reduces overfitting and improves generalization by creating diverse trees that capture different aspects of the data structure.

Rationale for Selection:

- Non-linearity: Effectively captures complex, non-linear relationships in data
- Robustness: Less prone to overfitting than single decision trees due to averaging
- Feature Importance: Provides built-in feature importance metrics based on impurity reduction
- Outlier Handling: Tree-based methods are robust to outliers in the data

Hyperparameters Tuned (via GridSearchCV):

- n_estimators: [100, 200] - Number of trees in the forest
- max_depth: [10, 20, None] - Maximum depth of each tree
- min_samples_split: [2, 5] - Minimum samples to split a node
- min_samples_leaf: [1, 2] - Minimum samples at a leaf node

3.3 Gradient Boosting Regressor

Gradient Boosting builds an ensemble of weak learners sequentially, where each new tree corrects the errors of the previous ensemble (Friedman 2001). The algorithm follows these steps:

- Initialize model with a constant value (mean of target)
- For each iteration: compute residuals (negative gradient), fit a tree to residuals
- Update predictions by adding the new tree multiplied by learning rate
- Repeat until convergence or maximum iterations reached

Unlike Random Forest which builds trees in parallel, Gradient Boosting builds trees sequentially, with each tree learning from the mistakes of the previous ensemble.

Rationale for Selection:

- High Accuracy: Often achieves state-of-the-art performance on tabular data
- Flexibility: Handles both linear and non-linear patterns effectively
- Regularization: Learning rate and tree constraints prevent overfitting
- Gradient Descent: Optimizes any differentiable loss function

Hyperparameters Tuned (via GridSearchCV):

- n_estimators: [100, 200] - Number of boosting stages
- learning_rate: [0.05, 0.1] - Shrinkage parameter
- max_depth: [3, 5, 7] - Maximum depth of individual trees
- subsample: [0.8, 1.0] - Fraction of samples for fitting trees

4. Results

4.1 Experimental Setup

The experimental setup was designed to ensure fair and unbiased model comparison:

- Data Split: 80% training (16,512 samples), 20% test (4,128 samples)
- Validation: 5-fold cross-validation on training set for hyperparameter tuning
- Test Evaluation: Final test set used only once to prevent data leakage
- Random Seed: Fixed at 42 for reproducibility

Evaluation Metrics:

- R-squared (R2): Proportion of variance explained (higher is better, max = 1.0)
- RMSE: Root Mean Squared Error in $100,000s (lower is better)
- MAE: Mean Absolute Error in $100,000s (lower is better)

4.2

Table 1 presents the performance metrics for all three models on the test set:

Table 1: Model Performance Comparison on Test Set

Model	R2 Score	RMSE	MAE
Linear Regression	0.5758	0.7456	0.5332
Random Forest	0.8061	0.5040	0.3260
Gradient Boosting	0.8422	0.4548	0.2964

Figure 6 visually compares the model performances. Gradient Boosting achieved the highest R-squared score of 0.8422 and the lowest RMSE of 0.4548, followed by Random Forest (R2 = 0.8061) and Linear Regression (R2 = 0.5758).

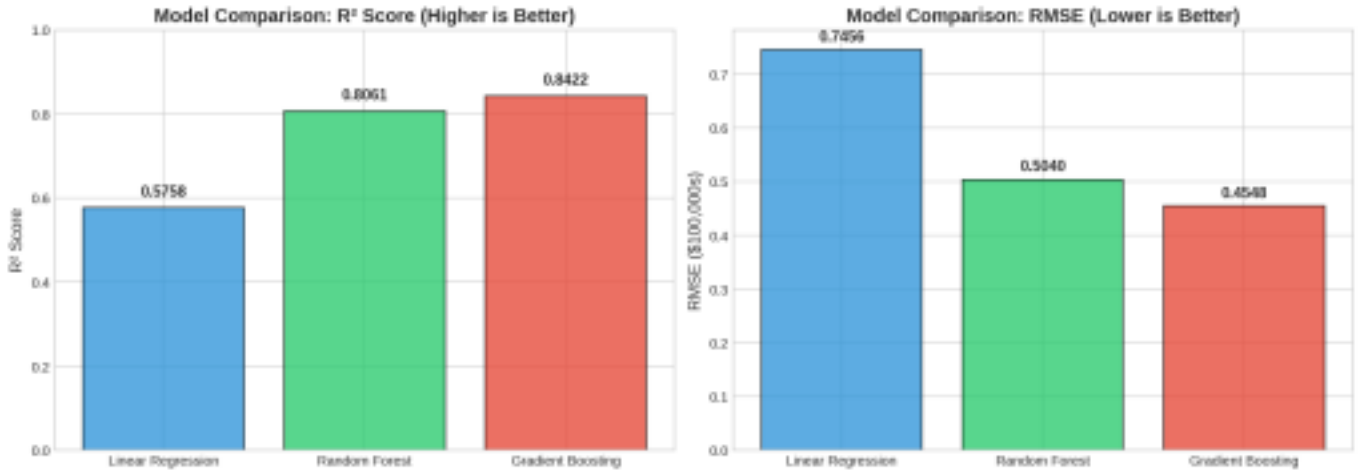

Figure 6: Model Performance Comparison

4.3 Prediction Analysis

Figure 7 shows actual vs. predicted values for each model. The ideal prediction would fall on the red diagonal line. Linear Regression shows systematic under-prediction for high-value homes, while ensemble methods provide more balanced predictions across all price ranges.

Figure 7: Actual vs Predicted House Values

Figure 8 presents residual analysis. Residuals should be randomly distributed around zero for a well-fitted model. Linear Regression shows heteroscedasticity (residual variance increasing with predicted value), while ensemble methods show more uniform residual distributions.

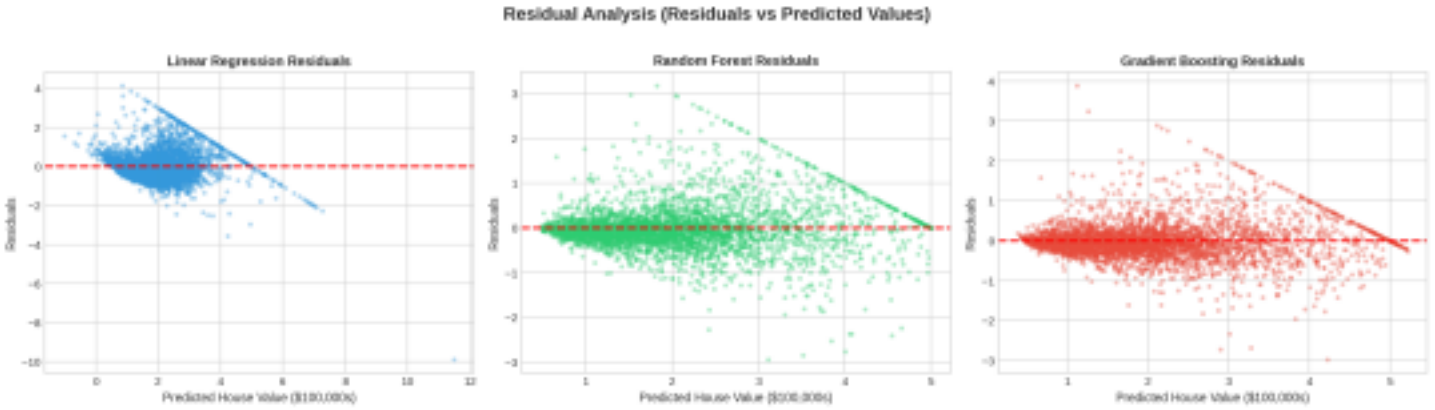

Figure 8: Residual Analysis

4.4 Feature Importance

Figure 9 compares feature importance from Random Forest and Gradient Boosting. Both models identify Median Income (MedInc) as the most important feature, followed by geographic features (Latitude, Longitude) and Average Occupancy.

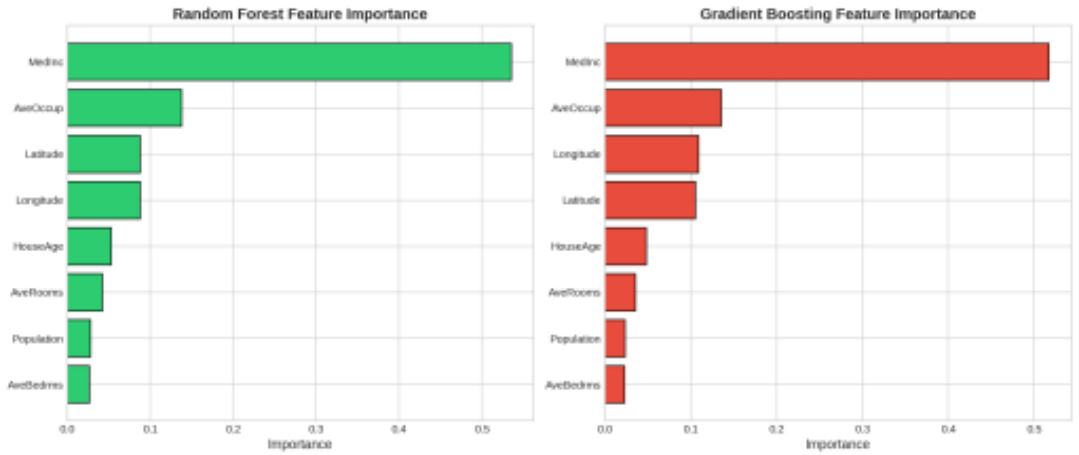

Figure 9: Feature Importance Comparison

4.5 Observations and Discussion

Based on the experimental results, several key observations emerge:

1. Model Performance Hierarchy:
Gradient Boosting > Random Forest > Linear Regression. The ensemble methods outperform Linear Regression by approximately 26-27 percentage points in R-squared, demonstrating the importance of capturing non-linear relationships in housing data.

2. Feature Importance Consistency:
Both ensemble models agree that Median Income is the strongest predictor, which aligns with economic intuition: higher-income areas tend to have more expensive housing.

3. Geographic Patterns:
The importance of latitude and longitude reflects the coastal premium in California housing markets, where properties closer to the coast command higher prices.

4. Potential Improvements:

- Feature Engineering: Create interaction terms (e.g., rooms per person)
- Spatial Features: Add distance to major cities, coast proximity
- Ensemble Stacking: Combine predictions from multiple models
- Address Capped Values: Use censored regression for $500K+ homes

5. Conclusions

5.1 Summary of Findings

This project successfully implemented and compared three machine learning algorithms for California housing price prediction. Gradient Boosting Regressor achieved the best performance with an R-squared of 0.8422 and RMSE of $45,479 (0.4548 in $100,000 units), demonstrating its effectiveness for regression tasks on tabular data. Random Forest performed competitively with R-squared of 0.8061, while Linear Regression provided a reasonable baseline (R-squared = 0.5758) but struggled with non-linear patterns.

Median income emerged as the strongest predictor of housing prices across all models, followed by geographic location features. This aligns with economic theory suggesting that housing prices are primarily driven by local income levels and

location desirability.

5.2 What I Learned

Through this project, I gained valuable practical experience in:

- End-to-end machine learning pipeline development from data exploration to model evaluation
- The importance of exploratory data analysis for understanding feature relationships
- Hyperparameter tuning with cross-validation to prevent overfitting
- Model comparison and interpretation of results in context
- The trade-off between model complexity and interpretability

6. References

Breiman, Leo. "Random Forests." *Machine Learning*, vol. 45, no. 1, 2001, pp. 5-32.
Friedman, Jerome H. "Greedy Function Approximation: A Gradient Boosting Machine." *The Annals of Statistics*, vol. 29, no. 5, 2001, pp. 1189-1232.
Hastie, Trevor, et al. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed., Springer, 2009.
Pace, R. Kelley, and Ronald Barry. "Sparse Spatial Autoregressions." *Statistics & Probability Letters*, vol. 33, no. 3, 1997, pp. 291-297.
Pedregosa, Fabian, et al. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research*, vol. 12, 2011, pp. 2825-2830.
"California Housing Dataset." *Scikit-learn Documentation*, scikit-learn.org/stable/datasets/real_world.html#california-housing-dataset. Accessed December 2024.

7. Acknowledgement

The author acknowledges the use of ChatGPT in the preparation or completion of this assignment. ChatGPT was used for brainstorming report wording, proofreading for grammar and clarity.
Claude AI (Anthropic) assisted with code organization and documentation. All analysis, modeling decisions, and interpretation of results were performed by the author.
Software and Libraries: Python 3.x, scikit-learn, pandas, numpy, matplotlib, seaborn, Jupyter Notebook.
Dataset: California Housing from scikit-learn (derived from the 1990 U.S. Census and made available via StatLib).

8. Source Code

GitHub Repository: https://github.com/Liam-M-Mays/ML-Final-Project.git

The repository contains:

- california_housing_prediction.ipynb - Main Jupyter notebook with all analysis
- requirements.txt - Python dependencies
- README.md - Project documentation and usage instructions
- figures/ - Directory containing all generated visualizations
- results/ - Directory containing model results in CSV format