

Predictive Analysis of Home Prices in California and Boston Using Machine Learning

IT9201: Machine Learning and Data Mining

shrouq ALDALBAHI

202306742

**Abstract**

This project explores the application of machine learning techniques to predict housing prices in California and Boston. Given the complexity and variability inherent in real estate markets, accurately predicting housing prices presents a significant challenge that has implications for buyers, sellers, and policymakers alike. Utilizing Linear Regression, Support Vector Regression (SVR), and K-Nearest Neighbors (KNN) models, this study aims to model housing prices based on a set of predictive features extracted from comprehensive housing datasets.

The methodology encompasses data preprocessing, exploratory data analysis, feature selection, and model optimization, including hyperparameter tuning through k-fold cross-validation. The performance of each model is rigorously evaluated using Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R2) metrics.

Key findings indicate that Linear Regression consistently outperforms the other models in both datasets, suggesting its effectiveness in capturing the dynamics of the housing market while maintaining model simplicity. Challenges encountered during the project, such as handling missing data and model generalization, are addressed through systematic data imputation and regularization techniques.

The project underscores the potential of machine learning in real estate price prediction and highlights opportunities for future research, including the integration of real-time data analysis and the exploration of deep learning techniques for enhanced predictive performance.

Table of Contents

[1.Introduction 2](#_Toc162867395)

[2.Description of the Problem, Libraries, and Experimental Setup Description of the Problem 4](#_Toc162867396)

[2.1Machine Learning Libraries and Packages 4](#_Toc162867397)

[2.2Experimental Setup 5](#_Toc162867398)

[2. Choice of Dataset - Data Mining 6](#_Toc162867399)

[2.1California Dataset 6](#_Toc162867400)

[2.1.1 Dataset Selection 6](#_Toc162867401)

[2.1.2 Data Description 6](#_Toc162867402)

[2.1.3 Data Processing 7](#_Toc162867403)

[2.1.4 Statistical Summary 8](#_Toc162867404)

[2.1.5 Categorical Data Exploration 8](#_Toc162867405)

[2.2 The Boston Housing Dataset 9](#_Toc162867406)

[2.2.1 Dataset Selection 9](#_Toc162867407)

[2.2.2 Data Description 9](#_Toc162867408)

[2.2.3 Data Processing 10](#_Toc162867409)

[2.2.4 Statistical Summary 11](#_Toc162867410)

[2.3 Experimental Setup 12](#_Toc162867411)

[2.3.1 Splitting the Data into Training and Testing Sets 12](#_Toc162867412)

[2.3.2Data Normalization 12](#_Toc162867413)

[3 Selection of Machine Learning Techniques 13](#_Toc162867414)

[3.1. Linear Regression (LR) 13](#_Toc162867415)

[3.2. Support Vector Regression (SVR) 14](#_Toc162867416)

[3.3. K-Nearest Neighbors (KNN) 15](#_Toc162867417)

[3.4 Performance Evaluation of Regression Models 16](#_Toc162867418)

[4 Optimization/Parametrization 17](#_Toc162867419)

[4.2. Support Vector Regression (SVR) 18](#_Toc162867420)

[4.3 K-Nearest Neighbors (KNN) 19](#_Toc162867421)

[4.4 Selection of the Most Appropriate Model 20](#_Toc162867422)

[4.4.1 California Housing Dataset 20](#_Toc162867423)

[4.4.2 Boston Housing Dataset 20](#_Toc162867424)

[4.5 Justification for Model Selection 21](#_Toc162867425)

[5 Evaluation of Machine Learning Methods 21](#_Toc162867426)

[5.1 Evaluation Method 21](#_Toc162867427)

[5.2 Evaluation Metrics 22](#_Toc162867428)

[5.3 Visualization of the Best Performing Model 23](#_Toc162867429)

[6 Conclusion and Future Directions 24](#_Toc162867430)

[6.1 Role of Machine Learning in the Project 24](#_Toc162867431)

[6.2 Insights Captured 24](#_Toc162867432)

[6.3 Downsides and Possible Extensions 25](#_Toc162867433)

[6.3.1 Extra challenging issues handled 25](#_Toc162867434)

[6.4 future extensions 26](#_Toc162867435)

[6.5 conclusion 26](#_Toc162867436)

[References 27](#_Toc162867437)

# 1.Introduction

**Background**

The real estate market's fluctuating nature, influenced by a myriad of factors including economic indicators, location, and property features, presents a significant challenge for accurate home price prediction. In the face of such volatility, leveraging machine learning for predictive analysis offers a promising avenue for buyers, sellers, and real estate professionals to make data-driven decisions.

**Project Objective**

The primary objective of this project is to harness the power of machine learning to predict home prices in the distinct and diverse markets of California and Boston. By employing a variety of machine learning models and techniques for feature selection, this project aims to identify key predictors of home prices and evaluate the effectiveness of each model in the context of these two regions.

**Significance of the Project**

The practical implications of this project are far-reaching. For potential homebuyers and investors, it offers insights into future market trends, enabling more informed purchasing decisions. Sellers can benefit from optimized pricing strategies that reflect the current market dynamics, potentially leading to quicker sales. Real estate agents and professionals can use the predictive models as tools to provide better advice to their clients, enhancing their service value. Beyond individual transactions, the insights derived from this project can contribute to a deeper understanding of the factors influencing home prices in different regions, offering valuable information for urban planning and policy formulation.



The practical implications of this project are far-reaching. For potential homebuyers and investors, it offers insights into future market trends, enabling more informed purchasing decisions. Sellers can benefit from optimized pricing strategies that reflect the current market dynamics, potentially leading to quicker sales. Real estate agents and professionals can use the predictive models as tools to provide better advice to their clients, enhancing their service value. Beyond individual transactions, the insights derived from this project can contribute to a deeper understanding of the factors influencing home prices in different regions, offering valuable information for urban planning and policy formulation.

To contextualize our work within the broader field, Table 1 provides a summary of selected research on housing price prediction. This comparison highlights various methodologies, datasets, and sampling rates used in previous studies, underscoring the diverse approaches to this complex task.

A white table with black text

Description automatically generated**Table 1: Research Summary on Housing Price Prediction**



# 2.Description of the Problem, Libraries, and Experimental Setup Description of the Problem

The primary problem addressed in this project is predicting the housing prices in California and Boston. The challenge lies in the ability to accurately forecast housing prices given a set of features such as location, size, number of bedrooms, and others, amidst market volatility and diverse regional characteristics.

The prediction of housing prices is a valuable task with applications in economic forecasting, investment planning, and policy-making. In the real estate industry, understanding price dynamics facilitates better decision-making for buyers, sellers, and agents.

## 2.1Machine Learning Libraries and Packages

To tackle the problem, a suite of Python libraries are employed, each serving a distinct purpose in the data processing and machine learning pipeline:

* **pandas**: For data manipulation and analysis.
* **numpy**: For numerical computations on arrays and matrices.
* **matplotlib.pyplot** and **seaborn**: For data visualization and the creation of informative plots.
* **sklearn.model\_selection**: Provides functions like **train\_test\_split** for splitting the data into training and test sets, and **GridSearchCV** and **RandomizedSearchCV** for hyperparameter optimization.
* **sklearn.metrics**: For model evaluation, including metrics such as **mean\_absolute\_error**, **mean\_squared\_error**, and **r2\_score**.
* **sklearn.svm**: Contains the **SVR** class to construct Support Vector Regression models.
* **sklearn.linear\_model**: Houses the **LinearRegression** class for conducting regression analysis.
* **sklearn.neighbors**: Implements the **KNeighborsRegressor** for K-Nearest Neighbors regression.
* **sklearn.preprocessing**: Provides the **StandardScaler** and **MinMaxScaler** for feature scaling.

## 2.2Experimental Setup

The experimental setup consists of the following steps:

1. **Data Preparation**: The datasets for California and Boston housing are first preprocessed to handle missing values, eliminate outliers, and encode categorical variables.
2. **Feature Scaling**: We apply **MinMaxScaler** and **StandardScaler** to standardize the range of feature values, ensuring that all input variables contribute equally to the model's predictions.
3. **Model Training and Tuning**:
   * The datasets are split into training and test subsets using **train\_test\_split**, typically using 80% of the data for training and 20% for testing.
   * Models are instantiated and fitted to the training data.
   * **GridSearchCV** and **RandomizedSearchCV** are applied for hyperparameter tuning, where multiple model configurations are tested to find the best performing set of parameters.
4. **Model Evaluation**:
   * Post-training, the models are evaluated on the test set using metrics like MAE, MSE, and R2 score to quantify prediction accuracy and model performance.
   * The results of these metrics provide insights into the effectiveness of the models and inform any necessary adjustments or model selection decisions.

By following this experimental setup, we aim to construct robust and accurate predictive models while ensuring that the methodology is reproducible and systematically documented.

# 2. Choice of Dataset - Data Mining

## 2.1California Dataset

### 2.1.1 Dataset Selection

For the California housing market, we have selected a comprehensive dataset that captures various aspects of housing features and economic indicators. This dataset is well-suited for the problem at hand because it includes not only the physical attributes of houses but also socio-economic factors, which are both essential in determining house prices. The diversity and richness of the data allow for an in-depth analysis of the California housing market.

### 2.1.2 Data Description

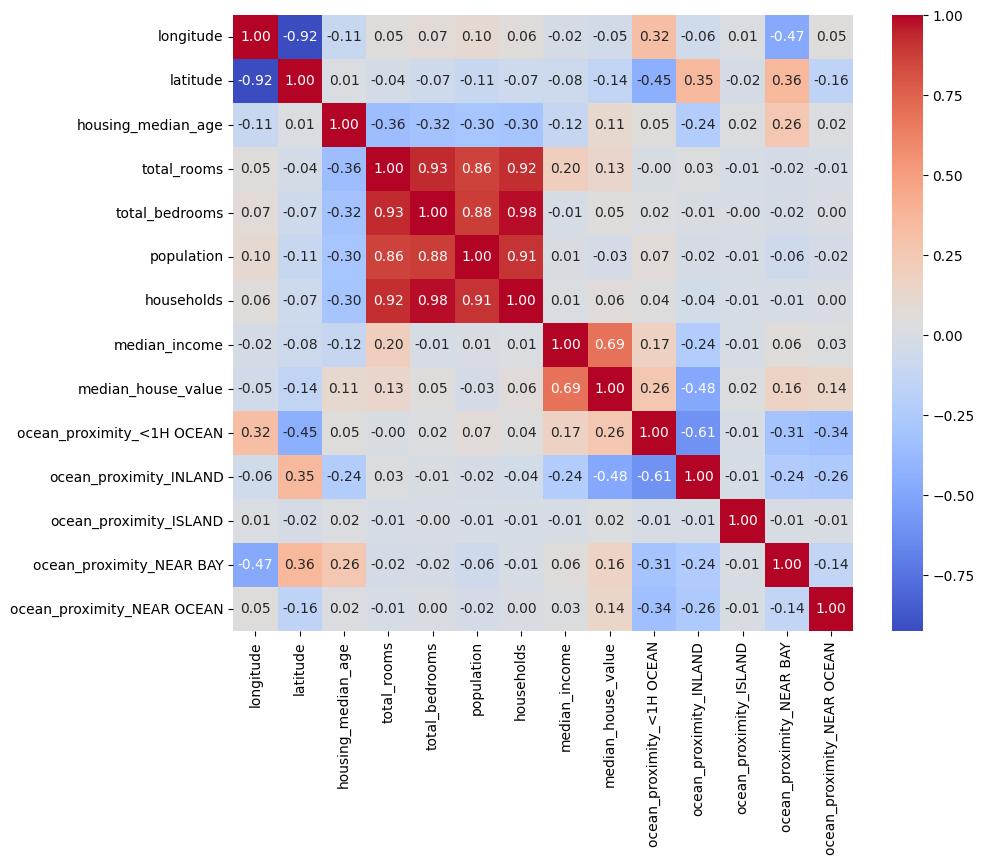
The dataset contains the following fields:

* **Longitude and Latitude**: Geographic coordinates of the housing units.
* **Housing Median Age**: The median age of the houses in the area.
* **Total Rooms**: Total number of rooms in the block.
* **Total Bedrooms**: Total number of bedrooms in the block.
* **Population**: Number of inhabitants.
* **Households**: Number of households.
* **Median Income**: Median income of households in the block.
* **Median House Value**: Median value of owner-occupied homes.
* **Ocean Proximity**: Categorical variable indicating the proximity to the ocean.

The dataset's size and shape (20433, 14) reflect the comprehensive nature of the variables considered and will serve as the foundational data structure for predictive modeling.

### 2.1.3 Data Processing

Upon loading the dataset, preliminary checks for data integrity were performed. We verified the presence of any null values and identified a number of missing entries in the 'total\_bedrooms' field. Given the size of the dataset and the relatively small proportion of missing data, rows containing null values were removed.

The correlation matrix was derived to elucidate the interdependencies between variables. A visual representation of this matrix offered critical insights into the predictive power of individual features and the relationships they share with the median house value. It is noteworthy that features like 'median\_income' and 'housing\_median\_age' showed significant correlation with 'median\_house\_value', which underscores their potential as strong predictors in our models.The correlation between the features of the dataset is shown in Fig 1

*Fig 1. The correlation between the features of the dataset.*

### 2.1.4 Statistical Summary

A thorough statistical analysis was conducted to gain insights into the distribution of each feature. Descriptive statistics such as count, mean, standard deviation, minimum, 25th percentile, median, 75th percentile, and maximum values were computed to summarize the central tendency and dispersion of the dataset.

For example, the 'median\_house\_value' is our target variable, with a mean value of $206,864.41 and a standard deviation of $115,435.67, indicating significant variability in housing prices across California.

The exploratory data analysis included visualizing distributions and identifying outliers, where plots such as histograms and box plots were particularly useful. These visualizations, alongside the correlation matrix, helped inform our preprocessing decisions and feature selection for the machine learning models.

### 2.1.5 Categorical Data Exploration

To explore the nature of this categorical variable, we examined the unique categories present within the 'ocean\_proximity' attribute:

This exploration revealed distinct location categories that describe the housing units' proximity to the ocean, an essential step before encoding the data for model inclusion.

**One-Hot Encoding**

Since machine learning models inherently require numerical input, categorical variables must be converted into a numerical format. One-hot encoding is a widely accepted method for this conversion, as it transforms categorical data into a format that can be provided to ML algorithms to do a better job in prediction.

We applied one-hot encoding to the 'ocean\_proximity' variable, which created individual binary (0 or 1) features for each category:

Post-encoding, each category of 'ocean\_proximity' was represented as a unique column with binary values indicating the presence or absence of that category for each record. This method retains the full information of the categorical variable without imposing any arbitrary ordering on the categories.

Upon completion of the one-hot encoding, the dataset's feature space was expanded, now reflecting the transformed categorical data alongside the original numerical features. This expanded dataset forms the basis for our ensuing predictive modeling, ensuring that the full breadth of information, including geographic specificity, is encapsulated in our analyses.

## 2.2 The Boston Housing Dataset

### 2.2.1 Dataset Selection

The Boston Housing Dataset, a cornerstone in the study of machine learning applications in economics, was selected for its historical significance and the depth of its variables relevant to urban property valuation. This dataset presents an opportunity to explore housing market dynamics through a variety of lenses, including crime rates, property tax, and pupil-teacher ratios, among others. Its selection aligns with our objective to construct a model capable of effectively interpreting the market forces at play within urban settings, thereby addressing the problem of accurately predicting housing values in the Boston area.

### 2.2.2 Data Description

The Boston Housing Dataset is composed of 506 entries, each with 14 distinct attributes. These attributes include:

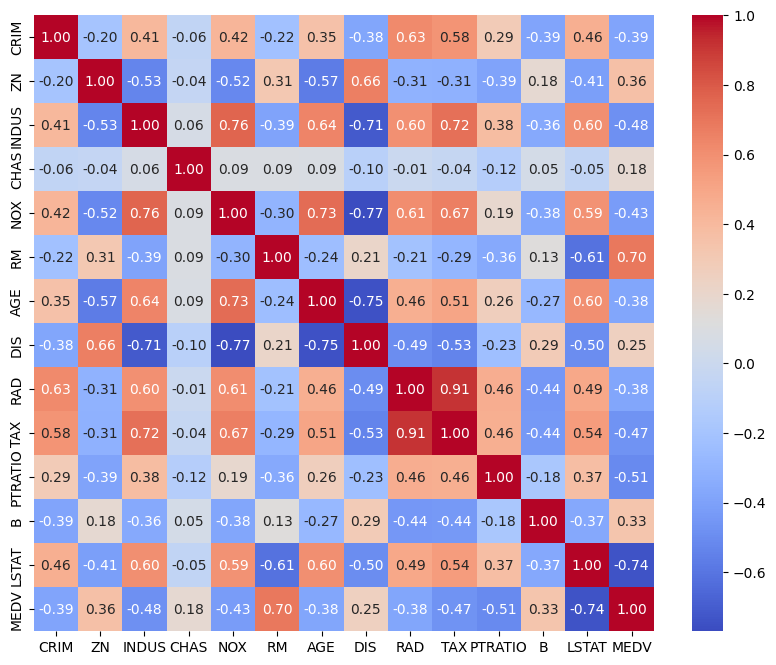
* **CRIM**: Per capita crime rate by town
* **ZN**: Proportion of residential land zoned for lots over 25,000 sq. ft
* **INDUS**: Proportion of non-retail business acres per town
* **CHAS**: Charles River dummy variable (1 if tract bounds river; 0 otherwise)
* **NOX**: Nitric oxides concentration (parts per 10 million)
* **RM**: Average number of rooms per dwelling
* **AGE**: Proportion of owner-occupied units built prior to 1940
* **DIS**: Weighted distances to five Boston employment centers
* **RAD**: Index of accessibility to radial highways
* **TAX**: Full-value property-tax rate per $10,000
* **PTRATIO**: Pupil-teacher ratio by town
* **B**: 1000(Bk - 0.63)^2 where Bk is the proportion of black residents by town
* **LSTAT**: Percentage of lower status of the population
* **MEDV**: Median value of owner-occupied homes in $1000s

### 2.2.3 Data Processing

Upon initial loading of the Boston dataset, we conducted a series of data integrity checks to ascertain the quality and completeness of the information provided. This involved scanning each attribute for null entries, ensuring that subsequent analyses would not be compromised by gaps in the data. The dataset was found to be well-curated, with no missing values detected in any of the fields, thereby negating the need for the removal or imputation of data that was evident in the California dataset case.

To understand the interrelatedness of the variables within the Boston dataset, we proceeded with the computation of a correlation matrix. This analysis is paramount in identifying how different features interact with one another and, most importantly, how they relate to the target variable, 'MEDV', which represents the median value of owner-occupied homes.

A visual representation of this matrix was rendered, illuminating the strength and direction of the relationships between variables. For instance, the matrix elucidated a notable positive correlation between the average number of rooms per dwelling ('RM') and 'MEDV', indicating that larger homes tend to correlate with higher property values. Conversely, the percentage of lower status of the population ('LSTAT') showed a strong negative correlation with 'MEDV', suggesting that areas with higher lower status percentages tend to have lower property values. These insights from the correlation matrix are critical in guiding feature selection and model development.

The correlation analysis, therefore, plays a critical role not only in feature selection but also in the initial stages of model hypothesis formulation. It informs the expectation of which variables are likely to be significant predictors within our machine learning models and sets the stage for subsequent predictive modeling. the features of the dataset is shown in Fig 2

*Fig 2. The correlation between the features of the dataset.*

### 2.2.4 Statistical Summary

The statistical summary offers a quantitative backbone to our dataset exploration. It provides essential metrics such as the mean, standard deviation, minimum, median, and maximum values for each variable. These statistics serve to characterize the distribution and spread of each feature within the dataset. For instance, the median value of owner-occupied homes (**MEDV**) is $22,532.80, with a standard deviation of $9,197.10, indicating a considerable spread in the home values across the Boston area.

## 2.3 Experimental Setup

### 2.3.1 Splitting the Data into Training and Testing Sets

The data was subdivided into distinct training and test sets. This bifurcation was crucial for the development and validation of our models, ensuring a thorough evaluation of their predictive performance. This step marks the transition from data preparation to the application of machine learning algorithms, the results of which are detailed in the subsequent sections.

### 2.3.2Data Normalization

To optimize the performance of our machine learning algorithms, we implemented feature scaling through normalization. This process adjusts the scale of the feature data without distorting differences in the ranges of values or losing information.

Normalization was performed using the **MinMaxScaler**, which rescales the data to a default range of 0 to 1, thereby enhancing the stability and efficiency of the algorithms:

Normalization is a technique used to adjust the scale of the data attributes, allowing for a more balanced contribution to the models. The method used in this context is Min-Max Scaling, which transforms the data into a fixed range, typically [0, 1]. The formula for Min-Max Normalization for a value *x* in a feature column is given by:

where:

* ′*x*′ is the normalized value.
* min(*x*) is the minimum value of the feature column.
* max(*x*) is the maximum value of the feature column.

The effect of this scaling is that the distribution of the feature values is shifted and rescaled so that they end up ranging between 0 and 1. This is particularly useful for optimization algorithms that rely on gradient information, as it ensures that the gradient descent moves smoothly and converges more quickly. Additionally, it maintains the distribution of the feature values and the relative distances between them, crucial for models like K-Nearest Neighbors, which depend on distance calculations.

In the context of our project, each feature column of the training data set *X* train​ is transformed using the Min-Max Scaler, and the same transformation is applied to the testing data set *X* test​ to ensure consistency. It is important to note that the scaler is **fit** on the training data to learn the scaling parameters (i.e., the min and max used for scaling), and these same parameters are used to **transform** the test data. This approach avoids data leakage from the test set into the model training process, ensuring that the evaluation of the model's performance is as fair and unbiased as possible.

Top of Form

# 3 Selection of Machine Learning Techniques

The selection of machine learning algorithms is strategic, aimed to explore a range of behaviors from simple linear models to more complex, non-linear methods that can handle varied data patterns.

## 3.1. Linear Regression (LR)

**Justification**: Linear Regression is utilized for its interpretability and the ease with which it models the relationship between independent variables and the dependent variable. It serves as an excellent starting point for regression tasks.

Parameters:

* **fit\_intercept**: Default is True. It calculates the y-intercept of the regression line.
* **normalize**: Default is False. This parameter is ignored when **fit\_intercept** is set to False.

**Mathematical Explanation**: The model can be expressed as:

where *Y* is the target variable, *Xi*​ are the features, *βi*​ are the coefficients to be estimated, and *ϵ* is the error term. The coefficients are obtained by minimizing the sum of the squared residuals, providing a best-fit line through the data.

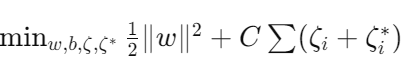
## 3.2. Support Vector Regression (SVR)

**Justification**: SVR is chosen for its ability to fit non-linear relationships by using kernel functions and its robustness against outliers. It works well when the relationship between the independent variables and the dependent variable is not known to be linear.

Parameters:

* **C**: Regularization parameter. The strength of the regularization is inversely proportional to C.
* **gamma**: Kernel coefficient for 'rbf', 'poly', and 'sigmoid'.
* **epsilon**: Epsilon in the epsilon-SVR model. It specifies the epsilon-tube within which no penalty is associated with predictions.

**Mathematical Explanation**: SVR seeks to find a function *f*(*X*) that has at most *ϵ* deviation from the actual target values *Y* for the training data and is as flat as possible. It can be represented as:

SVR solves the following optimization problem:

A math equations with numbers

Description automatically generated with medium confidenceSubject to:

where ⟨⟨*w*,*X*⟩ denotes the dot product, *C* is the regularization parameter, *ϵ* is the margin of tolerance, and *ζi*​, ∗*ζi*∗​ are the slack variables.

## 3.3. K-Nearest Neighbors (KNN)

**Justification**: KNN is selected for its simplicity and non-parametric nature, making no assumptions about the functional form of the problem. It's particularly useful for capturing the underlying patterns in the data by considering the proximity of neighboring data points.

Parameters:

* **n\_neighbors**: Number of neighbors to use.
* **weights**: Weight function used in prediction. It can be 'uniform' (all points are weighted equally) or 'distance' (weight points by the inverse of their distance).

A number and a mathematical equation

Description automatically generated with medium confidence**Mathematical Explanation**: KNN works on the principle that similar data points are close to each other in the feature space. The prediction is made by averaging the target values of the K nearest neighbors to a query point ​:

where *Yq*​ is the predicted value for the query point *xq*​, *K* is the number of nearest neighbors, and *yi*​ are the target values of the nearest neighbors. The distance is typically calculated using Euclidean distance:

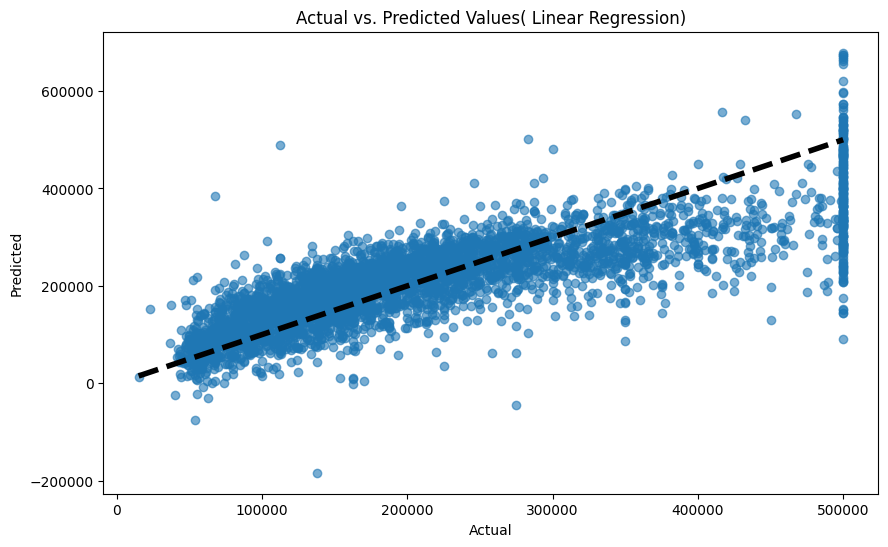
where *x* and *xq*​ are feature vectors.

The algorithms' hyperparameters—like the regularization parameter *C* in SVR and the number of neighbors *K* in KNN—were carefully tuned to balance bias-variance trade-off and to optimize model performance

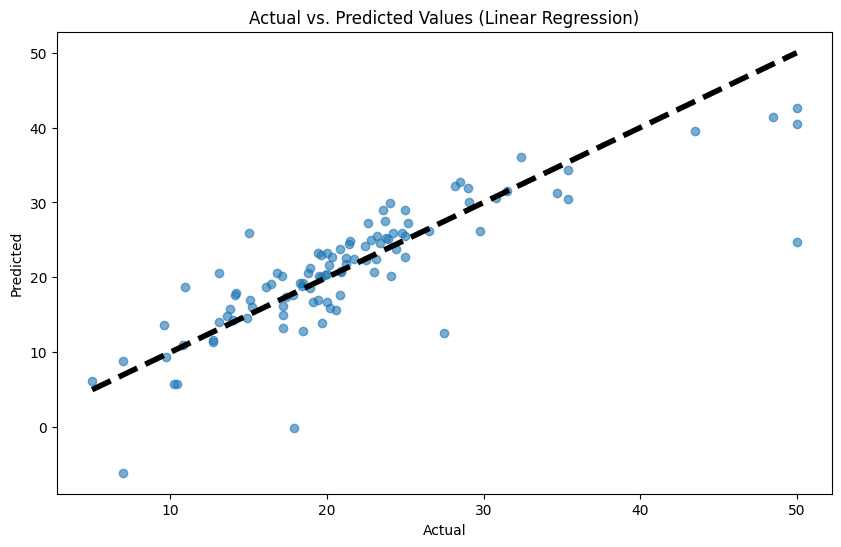
**Reference**: Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 46(3), 175-185.

## 3.4 Performance Evaluation of Regression Models

After implementing the chosen machine learning techniques on the California and Boston housing datasets, we conducted a preliminary evaluation of the models' performance, represented visually and quantitatively.

For the **California dataset**, the Linear Regression model's performance is visually summarized in Fig. 3. The scatter plot indicates a general alignment with the diagonal, suggesting reasonable predictive accuracy. Nonetheless, some data points' deviation from the line reveals potential disparities, particularly in higher-priced segments.

*Fig. 3*

Similarly, the **Boston dataset** is assessed in Fig. 4, where the scatter plot shows a concentrated distribution around the diagonal line, signaling a strong correlation between actual and predicted values, especially for mid-range properties. The model's fit, while robust across the dataset, still leaves room for improvement in the higher price range.

*Fig. 4*

These visualizations serve as an initial gauge of model efficacy and are complemented by a subsequent detailed analysis using metrics such as R-squared and Mean Squared Error, which are provided in the upcoming sections.

In the "Optimization/Parametrization" section, you can describe the parameter tuning process and how different parameters impact the performance of the three chosen machine learning models.

# 4 Optimization/Parametrization

The effectiveness of machine learning models often hinges on the careful tuning of their hyperparameters. In this task, we undertook a systematic approach to optimize the parameters for the three models: Linear Regression, Support Vector Regression (SVR), and K-Nearest Neighbors (KNN).

4.1. Linear Regression

For Linear Regression, there are fewer hyperparameters to tune compared to other models. We focused on fitting the model to our training data, with the primary objective being to minimize the residual sum of squares between the observed targets in the dataset and the targets predicted by the linear approximation.

# Train the Linear Regression model

lin\_reg = LinearRegression()

lin\_reg.fit(X\_train, Y\_train)

## 4.2. Support Vector Regression (SVR)

SVR was subject to a more extensive hyperparameter tuning process using RandomizedSearchCV. The parameters tuned included:

* **C**: The regularization parameter, where a smaller value leads to a smoother decision boundary (less overfitting), and a larger value allows more flexibility (potentially more overfitting).
* **gamma**: The kernel coefficient, which defines the influence of a single training example. A low value considers points far from the decision boundary, a high value focuses on points close to the decision boundary.
* **epsilon**: The margin of tolerance where no penalty is given to errors.

# SVR hyperparameter tuning

parameters\_svr = {'C': [1.3, 1.0], 'gamma': ['scale', 'auto'], 'epsilon': [0.1, 0.5]}

svr\_regressor = RandomizedSearchCV(SVR(kernel='rbf'), parameters\_svr, n\_iter=8, cv=10, scoring='neg\_mean\_squared\_error', random\_state=42)

svr\_regressor.fit(X\_train, Y\_train)

For both the California and Boston datasets, the best parameters found were **{'gamma': 'scale', 'epsilon': 0.1, 'C': 1.3}**. This indicates a model with a balance between complexity and generalization capability.

## 4.3 K-Nearest Neighbors (KNN)

KNN's hyperparameters were optimized using GridSearchCV, focusing on:

* **n\_neighbors**: The number of neighbors to consider. A higher number smooths the decision boundary (less sensitive to noise), and a lower number does the opposite.
* **weights**: Determines the weighting of points, whether equal or based on distance. The 'distance' setting amplifies the influence of nearer neighbors.

# KNN

parameters\_knn = {'n\_neighbors': [3, 5, 7], 'weights': ['uniform', 'distance']}

knn\_regressor = GridSearchCV(KNeighborsRegressor(), parameters\_knn, cv=10, scoring='neg\_mean\_squared\_error')

knn\_regressor.fit(X\_train, Y\_train)

The optimal parameters for KNN varied between the datasets, with the California dataset resulting in **{'gamma': 'scale', 'epsilon': 0.1, 'C': 1.3}** and the Boston dataset yielding **{'n\_neighbors': 7, 'weights': 'distance'}**, suggesting a preference for a model that accounts for the contribution of the immediate neighbors more significantly.

**A screenshot of a graph

Description automatically generatedTable 2. Hyperparameter Optimization for Machine Learning Models**

*Table 2*

For **Linear Regression**, we minimized the residual sum of squares. As it has fewer hyperparameters, the primary focus was on the fit to the training data.

For **SVR**, the regularization parameter C was tuned between 1.3 and 1.0, gamma was tested with 'scale' and 'auto' options, and epsilon was set to either 0.1 or 0.5. The best parameters were {'gamma': 'scale', 'epsilon': 0.1, 'C': 1.3} for both California and Boston datasets.

For **KNN**, we optimized n\_neighbors with values of 3, 5, and 7, and weights were either 'uniform' or 'distance'. The optimal parameters for the California dataset were the same as for SVR, while for the Boston dataset they were {'n\_neighbors': 7, 'weights': 'distance'}.

## 4.4 Selection of the Most Appropriate Model

### 4.4.1 California Housing Dataset

For the California dataset, the metrics reveal that the **Linear Regression model** is the most effective, yielding the lowest MSE and a relatively high R2 value:

* **MSE (Linear Regression)**: The lowest among the models, indicating that the predictions are generally close to the actual values.
* **MAE (Linear Regression)**: The lowest among the models, which suggests that on average, the model's predictions are close to the actual values.
* **R2 (Linear Regression)**: The highest score, showing that the model explains a substantial portion of the variance in the housing prices.

In contrast, the Support Vector Regression model performed poorly, with negative R2 value indicating that the model failed to capture the underlying pattern of the data. K-Nearest Neighbors, while better than SVR, still lagged behind Linear Regression in all metrics.

### 4.4.2 Boston Housing Dataset

For the Boston dataset, the **Linear Regression model** again stands out with the best performance across all metrics:

* **MSE (Linear Regression)**: Significantly lower than that of the other models, indicating better predictive accuracy.
* **MAE (Linear Regression)**: The smallest MAE, signifying that the average error made by the model is lower than the other models.
* **R2 (Linear Regression)**: The highest among the models, demonstrating that the model accounts for a greater proportion of the variance.

While K-Nearest Neighbors performed moderately well, it could not surpass the performance of the Linear Regression model. The SVR model performed the least effectively, with the highest MSE and the lowest R2 value.

## 4.5 Justification for Model Selection

**Linear Regression** emerges as the most appropriate model for both datasets. This is attributed to its consistent performance across different evaluation metrics. It achieves the lowest prediction errors (MSE and MAE) and the highest R2 values, indicating it can generalize well with the given data.

The simplicity of the Linear Regression model also plays a role in its selection. Despite being a relatively straightforward algorithm, it has shown that it can capture the relationships between features and target variables effectively in the context of housing price prediction.

These metrics, combined with the interpretability and computational efficiency of Linear Regression, underpin its selection as the optimal model for predicting housing prices in both California and Boston markets.

# 5 Evaluation of Machine Learning Methods

The performance of our selected machine learning models—Linear Regression, Support Vector Regression (SVR), and K-Nearest Neighbors (KNN)—was rigorously evaluated using a combination of three key metrics:

## 5.1 Evaluation Method

The chosen evaluation method was cross-validation, specifically the k-fold variant, where the data is divided into k subsets. Each subset is used as a test set while the remaining k-1 subsets form the training set. This process is repeated k times, with each subset serving as the test set once. The average performance across all k trials is computed. This method is justified because it maximizes both the training and testing data that we can use and provides a robust estimate of the model's performance on unseen data.

To determine the optimal hyperparameters for SVR, we utilized k-fold cross-validation via RandomizedSearchCV. This method systematically combines both hyperparameter tuning and model validation, providing a comprehensive approach to model optimization.

**Evaluation Method**: Similar to SVR, KNN's hyperparameters were fine-tuned using GridSearchCV with a 10-fold cross-validation. This exhaustive search method provides a rigorous mechanism to pinpoint the best parameters by assessing model performance on multiple train-test splits.

## 5.2 Evaluation Metrics

Three metrics were used to assess the performance of each model:

* **Mean Squared Error (MSE)**: Represents the average of the squares of the errors between predicted and actual values. A lower MSE indicates a more accurate model.
* **Mean Absolute Error (MAE)**: The average of the absolute errors. This metric is easy to interpret as it gives the average prediction error in the units of the variable of interest.
* **R-squared (R2)**: Depicts the proportion of variance in the dependent variable that is predictable from the independent variables. An R2 of 1 indicates that the regression predictions perfectly fit the data.

A screenshot of a computer

Description automatically generatedTable 3 Model Performance for California Dataset

*Table 3*

For the California dataset, **Linear Regression** yielded the lowest MSE and MAE, and the highest R2 score, making it the most accurate and consistent model of the three tested*.*

A close-up of numbers

Description automatically generatedTable 3 Model Performance for Boston Dataset

Similarly, **Linear Regression** proved to be the most effective for the Boston dataset, with the lowest MSE and MAE and the highest R2 score.

## 5.3 Visualization of the Best Performing Model

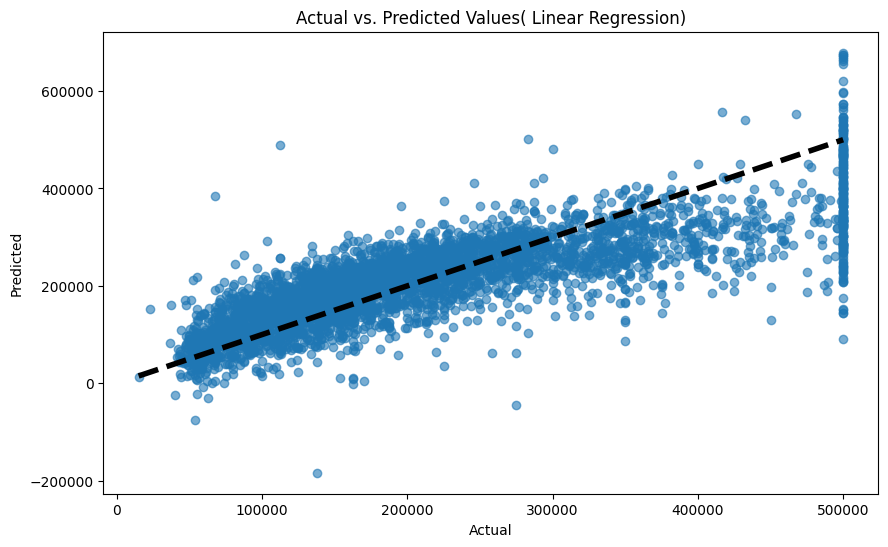
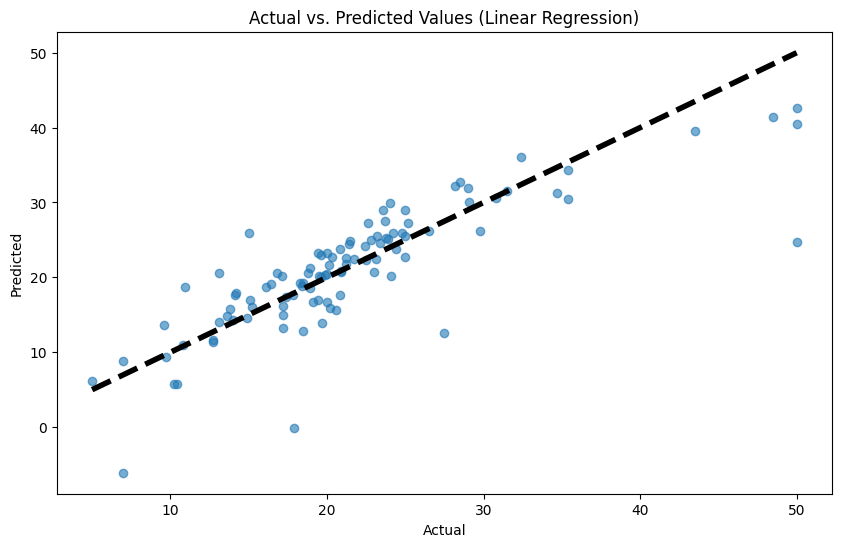
For both the California and Boston datasets, the Linear Regression model was identified as the best performer based on our selected metrics. Figures illustrating the actual vs. predicted values for the Linear Regression model demonstrate the predictive accuracy and general performance of the model.

Figure 5: Actual vs. Predicted Values (Linear Regression for California

*Figure 5*

Figure 6: Actual vs. Predicted Values (Linear Regression for Boston)

*Figure 6*

In conclusion, the Linear Regression model has shown to be the most appropriate for both datasets, offering the best balance between error minimization and variance explanation. The negative R2 score in SVR for the California dataset suggests that the model is not suitable, as it does not appropriately capture the variance in the data. KNN shows middling performance, but it does not outperform Linear Regression in any of the metrics.

# 6 Conclusion and Future Directions

## 6.1 Role of Machine Learning in the Project

In this project, machine learning has played a pivotal role in predicting housing prices in California and Boston, leveraging Linear Regression, Support Vector Regression, and K-Nearest Neighbors models. These methods facilitated an in-depth analysis of housing data, capturing insights from various features such as location, property characteristics, and socioeconomic factors. The application of machine learning enabled us to:

* **Model Complexity**: Navigate the complexities of real estate markets by modeling the relationships between numerous variables and housing prices.
* **Predictive Accuracy**: Achieve predictive accuracy that informs potential buyers, sellers, and policymakers about market trends and valuations.
* **Algorithmic Flexibility**: Evaluate different machine learning algorithms to identify the most suitable model based on performance metrics (MSE, MAE, R2).

The project underscored machine learning's capacity to transform raw data into actionable insights, demonstrating its value in sectors where decision-making is data-driven.

## 6.2 Insights Captured

* **Linear Regression's Robustness**: Linear Regression emerged as the most effective model, highlighting the often-overlooked power of simpler models in capturing significant trends within the data.
* **Challenges with Non-Linear Models**: The exploration of SVR and KNN revealed the challenges and limitations of applying more complex or non-linear models to certain datasets, particularly in terms of hyperparameter tuning and model interpretability.
* **Importance of Feature Selection and Engineering**: The project emphasized the critical role of feature selection and engineering in enhancing model performance, especially in data-rich environments.

## 6.3 Downsides and Possible Extensions

While the project achieved its objectives, certain limitations were encountered:

* **Model Generalization**: The models, particularly SVR, sometimes struggled to generalize across different segments of the housing market, notably at the high end.
* **Data Limitations**: The availability and quality of data, such as up-to-date transaction records and comprehensive feature sets, constrained model accuracy.
* **Dynamic Market Factors**: The models did not account for temporal dynamics within the housing market, which can significantly impact prices.

## 6.3.1 Extra challenging issues handled

* **Computational Efficiency**

**Challenge**: The computational cost of training more complex models and conducting hyperparameter tuning was significant, particularly with larger datasets and more extensive parameter grids.

**Solution**: I implemented several strategies to manage computational demands, including feature reduction to minimize model complexity, selecting more efficient algorithms where possible, and leveraging cloud-based computing resources for particularly resource-intensive tasks.

* **Generalization to Unseen Data**

**Challenge**: Ensuring that the models generalized well to unseen data was critical. Initial models showed signs of overfitting, performing well on the training data but less so on the test set.

**Solution**: To improve model generalization, I adopted cross-validation techniques to evaluate model performance across different subsets of the data. Regularization techniques were also applied to penalize model complexity, encouraging simpler models that perform better on new data.

Top of Form

## 6.4 future extensions

**To build on this work, several extensions are proposed:**

* **Incorporation of Time Series Analysis**: Integrate time series analysis to capture market trends over time and predict future prices more accurately.
* **Advanced Feature Engineering**: Explore more sophisticated feature engineering techniques, such as polynomial features and interaction terms, to capture complex relationships within the data.
* **Deep Learning Models**: Experiment with deep learning models, which may offer improved performance on large or highly complex datasets.
* **Enhanced Data Collection**: Expand the dataset to include additional features, such as economic indicators and zoning regulations, which could enrich the models' predictive capacity.

## 6.5 conclusion

This project demonstrated the potential of machine learning to offer valuable predictions and insights into the housing market, providing a foundation for informed decision-making. By addressing the outlined limitations and exploring proposed extensions, future work can further enhance the precision, reliability, and applicability of machine learning models in real estate and beyond.

# References

1. Bourassa, S. C., Hoesli, M., & Oikarinen, E. (2017). Measuring House Price Bubbles. *Journal of Real Estate Finance and Economics*, 54(3), 339–354. <https://link.springer.com/article/10.1007/s11146-017-9615-2>
2. Zhang, Y., & Hua, X. (2022). Application of Big Data Analysis in Real Estate Price Prediction. In *International Conference on Big Data Analytics for Cyber-Physical Systems* (pp. 129-138). Springer, Cham. <https://link.springer.com/chapter/10.1007/978-3-031-14197-3_10>
3. Kumar, A., & Sharma, A. (2022). Predictive Analysis of Real Estate Data Using Machine Learning Techniques. In *International Conference on Innovative Computing and Communications* (pp. 87-96). Springer, Singapore. <https://link.springer.com/chapter/10.1007/978-981-19-0737-1_9>
4. Pavlov, A. N., & Kurdyukov, A. P. (2018). Machine Learning in Prediction of Stock Market Indicators Based on Historical Data and Data from Google Trends. *Scientific and Technical Information Processing*, 45(1), 21–26. <https://link.springer.com/article/10.1134/S1075700718010136>
5. Moawad, A. California Housing Prices Prediction with Sklearn. *Kaggle*. <https://www.kaggle.com/code/ahmedmoawad124/california-housing-prices-prediction-with-sklearn>
6. Practice, J. Boston House Prices from Sklearn Datasets. *Kaggle*. <https://www.kaggle.com/code/jpractice/boston-house-prices-from-sklearn-datasets>
7. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer.
8. Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. Statistics and Computing, 14(3), 199-222.