Deciphering House Price Dynamics Using Predictive Modeling on Housing Data

Allen Zou, Ting-An Lu, Zachary Chao, and Casey Hild

May 12, 2024

Abstract

1 Introduction

The real estate market is a complex ecosystem influenced by various factors, ranging from property characteristics to broader economic trends. Understanding the dynamics of house prices is essential for stakeholders involved in buying, selling, or investing in real estate. This research seeks to unravel the intricacies of house price dynamics through predictive modeling, offering insights that can inform strategic decision-making in the housing market.

In this study, we aim to address two fundamental research questions: What factors influence house prices, and can we construct a predictive model to estimate house prices based on these factors? By delving into these questions, we seek to unravel the complex interplay of factors shaping house price dynamics and explore the feasibility of developing a robust predictive model.

Based on the dataset variables, we hypothesize that certain factors such as property type, location, lot size, overall quality, and condition will have a significant impact on house prices. Additionally, we anticipate that incorporating these factors into predictive models will enable accurate estimations of house prices.

2 Dataset

The dataset used in this study consists of two comma-separated values (CSV) files: train.csv and test.csv. The train.csv file comprises a tabular format with 1,460 rows and 81 columns, while the test.csv file contains 2,919 rows and 81 columns. Each row in both datasets represents a distinct property sale transaction, serving as a unique sample in the dataset.

The first column of each dataset is labeled "Id" and contains a unique identifier for each property sale. The target variable, "SalePrice," is included as the last column, representing the sale price of each property. The features of each property sale transaction are captured in the columns between the first and last columns of the dataset.

The dataset encompasses a wide array of features that describe various aspects of residential properties. These features include categorical variables such as MSSubClass, which identifies the type of dwelling involved in the sale, and MSZoning, which indicates the general zoning classification of the sale. Additionally, numerical variables such as LotFrontage (linear feet of street connected to property) and LotArea (lot size in square feet) provide quantitative measurements of property characteristics. Other essential features cover aspects such as property configuration (LotShape, LotConfig), neighborhood information (Neighborhood), and conditions related to the property (Condition1, Condition2). Variables like OverallQual and OverallCond rate the overall material/finish and condition of the house, respectively, while variables like YearBuilt and YearRemodAdd provide information about the original construction and remodel dates. These features, along with many others like Heating, KitchenQual, and GarageType, offer comprehensive insights into the properties included in the dataset.

2.1 Data Pre-processing

The dataset initially comprised separate files for training and testing, with an unusual split of 1461:1460. To rectify this, the data were combined and shuffled, followed by an 80:20 train-test split. Subsequently, due to the presence of numerous missing entries, preprocessing was necessary to ensure compatibility with scikit-learn, which does not support NaN values. We devised an approach to minimize data loss while addressing missing val-

ues. Columns with a substantial proportion of NaN values (approximately 10% or more) were initially pruned, removing entries such as LotFrontage, Alley, FireplaceQu, PoolQC, Fence, and MiscFeature. This step significantly reduced the number of NaN values while retaining most of the variables. Subsequently, the remaining rows containing NaN values were removed, resulting in a total of 1581 rows being eliminated. Despite this, the dataset retained 1070 rows, deemed sufficient for subsequent modeling tasks involving linear regression, decision trees, and support vector machines.

3 Methods

3.1 Model Selection

In our study, we adopted a deliberate approach to model selection, aiming to achieve multiple objectives with each chosen algorithm to answer our research questions. In short, we employed three distinct models to analyze housing price data and extract insights into the factors influencing property values. Our initial choice of linear regression was motivated by its simplicity and interpretability. Subsequently, we turned to decision trees as our second model choice. Decision trees also offer interpretability, aligning with our goal of identifying key variables driving house prices. We also recognized the importance of predictive accuracy in our analysis, so we included support vector machines (SVMs) as another model in our study.

3.2 Benefits and Limitations of Models

Before delving into the hyperparameter selection and training process, it's essential to understand the respective strengths and weaknesses of these models, as they influenced our decision to choose them. Let's explore the benefits and limitations of linear regression, decision trees, and Support Vector Machines (SVMs) for modeling tasks.

3.2.1 Linear Regression

Linear regression offers a few strengths that make it a popular choice for modeling tasks. One of its primary advantages is its simplicity, as it is straightforward to implement and interpret. This simplicity makes linear regression particularly suitable for initial exploratory analysis or as a baseline model for comparison. Additionally, linear regression models are computationally efficient, making them practical for quick model prototyping and experimentation. Another key strength is the interpretability of linear regression coefficients, which provide insights into the relationship between predictor variables and the target variable.

However, linear regression also has notable weaknesses. One limitation is its assumption of a linear relationship between predictors and the target variable, which may not adequately capture complex nonlinear patterns present in the data. Linear regression is also sensitive to outliers, as they can disproportionately influence the model's predictions. Furthermore, linear regression relies on several assumptions about the data, including linearity, homoscedasticity, and independence of errors, which may not always hold true in real-world datasets. Nevertheless, we still chose to include linear regression to serve as a useful baseline to compare with the other models.

3.2.2 Decision Trees

Decision trees offer unique strengths that distinguish them from linear regression models. One of their primary advantages is their ability to capture nonlinear relationships between predictors and the target variable, providing flexibility in modeling complex data. Decision trees also inherently rank predictors based on their importance in predicting the target variable, facilitating feature selection and providing insights into variable importance. Additionally, decision trees are highly interpretable, as they provide intuitive visualizations of decision paths, making it easy to understand the model's decision-making process.

Decision trees also have notable weaknesses. They are prone to overfitting, especially with deep trees that capture noise in the data rather than true patterns. Decision trees can also exhibit instability, as small variations in the data can lead to significantly different decision trees, impacting model robustness. Furthermore, decision trees produce piecewise constant predictions, which may not capture gradual changes in the target variable across predictor space. However, we do have ways to address overfitting and instability, which we will discuss in later sections, particularly through grid search to find the best hyperparameters for this model.

3.2.3 Support Vector Machines

While decision trees are valuable for feature interpretation, we also recognized the importance of predictive accuracy in our analysis. Support Vector Machines (SVMs) offer versatility and robustness in modeling various types of data. One of their key strengths is their ability to model both linear and nonlinear relationships through different kernel functions, providing flexibility in capturing complex patterns in the data. SVMs aim to maximize the margin between classes, leading to robust generalization performance and resistance to overfitting, especially with large margins. Additionally, SVMs perform well in high-dimensional spaces, making them suitable for datasets with many predictors such as the one described in the **Dataset** section.

However, SVMs do have some limitations. They can be complex to tune, requiring careful selection of hyperparameters such as the choice of kernel function and regularization parameters. Training SVMs can be computationally intensive, particularly with large datasets or complex kernel functions, which may limit their scalability. Most imporatntly, SVMs often lack interpretability, particularly with nonlinear kernel functions, making it challenging to understand the model's decision rationale. Despite the lack of interpretability, we chose to include SVMs for their renowned robust predictive performance. Additionally, as described above, we utilize grid search to assist with parameter tuning.

3.3 Hyperparameter Selection and Model Training Process

In this section, we delve into the process of hyperparameter selection and model training for each algorithm employed in our study. We discuss our approach to fine-tuning hyperparameters for linear regression, decision trees, and support vector machines (SVMs) and insights gained during the process.

3.3.1 Linear Regression

Starting with linear regression, a straightforward yet fundamental model in machine learning, it has no hyperparameters other than epochs. We initially set the epochs to 1000; however, upon observation, we found that the model's performance plateaued after just 10 epochs. To expedite the training process without compromising accuracy, we adjusted the epochs to this lower value.

3.3.2 Decision Trees

Moving on to decision trees, we adopted a more comprehensive approach to hyperparameter tuning. Leveraging grid search, along with one-hot encoding for categorical variables (required by sklearn), we explored various combinations within predefined ranges for parameters such as 0-20 for `max_depth`, 1-4 for `min_samples_split`, 1-4 for `min_samples_leaf`, various functions for `max_features`, and 0.0-0.5 for `min_impurity_decrease`. These ranges were carefully selected based on empirical evidence from manual experimentation, where we observed optimal performance around specific values. However, no significant challenges were encountered during this tuning process, highlighting the robustness of decision trees.

3.3.3 Support Vector Machines

Hyperparameter tuning for the support vector machine (SVM) model introduced unique considerations, particularly in the context of kernel selection and parameter optimization. Leveraging grid search methodology and incorporating one-hot encoding for categorical variables (a prerequisite for SVM implementation in sklearn), we explored various parameter combinations. However, manual inspection of the tuning process revealed an unexpected trend. Despite rigorous experimentation with hyperparameters such as gamma and regularization parameter (`C`), marginal improvements in model performance were observed.

Remarkably, we observed that the choice of kernel type exerted the most significant influence on model performance. Notably, the linear kernel consistently produced satisfactory results, achieving a mean squared error (MSE) below 0.001. In stark contrast, employing alternative kernels resulted in a substantial escalation of MSE, often exceeding a million. This observation underscores the superior performance of the linear kernel compared to other kernel functions, highlighting its efficacy in the context of our dataset.

Furthermore, despite exploring additional hyperparameters specific to non-linear kernels (e.g., degree for polynomial kernel, coef0 for polynomial and sigmoid kernels), grid search consistently

favored the linear kernel configuration for its superior predictive capability and generalization performance.

Lastly, our experimentation revealed that lowering the learning rate below 0.1 drastically increased the training time. Upon reaching approximately 0.03 learning rate, training progress stagnated. Interestingly, variations in the learning rate did not significantly affect model performance when combined with other hyperparameters. Consequently, we omitted the learning rate as a hyperparameter to streamline computational resources without compromising model efficacy.

3.4 Final Hyperparameters

After hyperparameter tuning using sklearn's GridSearchCV, we finalized the hyperparameters for the decision tree and SVM models in the tables shown below.

Hyperparameter	Value
max_depth	10
max_features	'auto'
min_impurity_decrease	0.4
min_samples_leaf	2
min_samples_split	2

Table 1: Optimal hyperparameters for the decision tree model.

Hyperparameter	Value
C	0.1
epsilon	0.1
gamma	'scale'
kernel	'linear'

Table 2: Optimal hyperparameters for the SVM model.

Since linear regression does not involve hyperparameters and its performance plateaued after 10 epochs, we set the number of epochs to 10 for efficiency.

4 Results

4.1 Performance Metrics

The performance of each model was evaluated using two common metrics: Mean Absolute Error (MAE) and Mean Squared Error (MSE). MAE measures the average magnitude of errors in predictions without considering their direction. Conversely, MSE calculates the average of the squares of the errors, giving higher weights to larger errors and penalizing outliers more heavily.

Model	MSE
Linear Regression	427,864,778,376
Decision Tree	4,857,782
SVM	0.0008

Table 3: Mean Squared Error (MSE) for each model

Model	MAE
Linear Regression	650,448
Decision Tree	607
SVM	0.0267

Table 4: Mean Absolute Error (MAE) for each model

4.2 Interpretation of Model Performance

In this section, we compare the performance of the three selected models: linear regression, decision trees, and support vector machines (SVMs). We aim to assess the effectiveness of each model in predicting housing prices and gain insights into their strengths and limitations.

4.2.1 Linear Regression

It's evident that linear regression demonstrates the poorest performance among the models, with significantly higher MAE and MSE values compared to the others. The substantial discrepancy between the predicted and actual house prices, as indicated by the exceptionally high MAE value, renders this model unreliable for accurate price predictions. Therefore, leveraging linear regression for identifying significant factors influencing house prices might not yield meaningful insights due to its poor predictive performance.

4.2.2 Decision Trees

In contrast, decision trees exhibit a much-improved performance, with a relatively low MAE value, indicating closer predictions to the actual house prices. This suggests that decision trees offer a viable approach for predicting house prices more accurately while also providing interpretability, mak-

ing them suitable for analyzing the factors contributing to house price variations.

4.2.3 Support Vector Machines

Support vector machines (SVMs) demonstrate exceptional predictive accuracy, as reflected by their remarkably low MAE value, indicating near-perfect predictions of house prices. However, the lack of interpretability associated with SVMs poses a challenge in extracting insights into the underlying factors driving house prices.

4.2.4 Model Suitability for Hypothesis Testing

Among the models considered, decision trees and support vector machines (SVMs) emerge as promising candidates for addressing the questions outlined in our proposal. While both models exhibit strong predictive performance, with SVMs often predicting house prices with remarkable accuracy, the interpretability of SVMs remains limited. Therefore, for the interpretability aspect of our hypothesis, we rely on decision trees. Although linear regression was also explored, its notably high MAE and MSE values suggest limited effectiveness compared to decision trees and SVMs in predicting housing prices. Therefore, the subsequent analysis will primarily focus on the insights derived from decision trees and support vector machines.