Machine Learning to Predict California Housing Prices

Authors: Bryan Rivas, Timothy Vu, Justin Dong
Professor: Rong Jin
Department: Computer Science and
CPSC 483: Intro to Machine Learning
December 8, 2024

Abstract*—The increasing prices of housing in California have made homeownership increasingly unattainable for our generation, with instability that has persisted since the early 2000s and worsened by the pandemic. Understanding market trends is crucial for aspiring property owners in making informed decisions on when to buy or sell property. This paper goes in depth of the performance comparison of different machine learning models for housing price prediction by considering various feature parameters such as property size, the number of bedrooms, amenities, and many more. These machine learning models have been analyzed using metrics such as R^2 (accuracy), Mean Squared Error, and Mean Absolute Error. It proves how these models have tremendous potential for real-world applications in finding a way through the tough real estate market in California.

Keywords*—Linear Regression, Decision Tree, Naive Bayes, Neural Network, Polynomial Regression, Random Forest Regression, Support Vector Machine, price, predict

I. Introduction

Owning property or a home is an essential component of the American Dream. Since the early 2000s, the housing market has faced significant instability and soaring prices, and the addition of the pandemic created an even bigger affordability issue. For many people in our generation, owning a home increasingly feels out of reach as time goes on. The average price of a home in California is now set to exceed $900,000. This is driven by a couple of factors which include high demand, limited inventory, and other various economic factors [1].

The effects of this housing crisis have affected other aspects of life as well, from rental property affordability to long-term financial planning. Almost half of California residents consider the cost of housing to be a financial burden. Data compiled has confirmed that California has higher rates of cost-burdened households compared to the entirety of the United States [2]. The cost of buying a home has increased over the median income. The annual household income needed to qualify for a mortgage for a low-tier home is about $136,000 which is about 40% higher than the median household income back in 2023 [3].

In this project, we utilize various types of machine learning models like Linear Regression, Decision Tree, Naive Bayes, Neural Network, Polynomial Regression, Random Forest Regression, and Support Vector Machines to analyze and predict housing prices in California. By applying algorithms, we aim to find the model that would best fit this application. By evaluating metrics like R-squared (R^2), Mean Squared Error (MSE), and Mean Absolute Error (MAE), our research identifies the best model. We intend to use machine learning to make what seems to be a distant dream, of owning a home, a reality in the near future.

II. Related Work

In the field of housing price prediction, several studies have provided valuable insights into the application of machine learning models, metrics, and techniques. These works have served as critical references for our project, helping to shape the methodologies and evaluation strategies we adopted.

The research study available on ResearchGate, *"House price prediction based on different models of machine learning"*, provided foundational guidance

on evaluation metrics[4]. The use of metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) was particularly critical in measuring the performance of our models. Additionally, the paper deepened our understanding of Random Forest Regression and Linear Regression, detailing how these models can be applied effectively to predict housing prices. This understanding directly influenced our experimental setup, particularly in the selection and implementation of regression-based models in our study.

The IRJET research paper, *"Prediction of Housing Prices Using Classification Models"*, offered valuable insights into the use of classification techniques for housing price prediction[5]. While regression models are more commonly applied in this domain, the IRJET study demonstrated how classification approaches could categorize housing prices into distinct segments. Inspired by this approach, we explored classification models such as Naive Bayes and Decision Trees to predict housing prices, allowing us to compare the performance of regression versus classification techniques in our project.

Lastly, the study presented in the arXiv paper, *"Housing Price Prediction Using Machine Learning: A Case Study of Florida"*, focused on the Florida housing market and demonstrated the effectiveness of advanced methods such as XGBoost with target binning[6]. The empirical results showing the efficiency of XGBoost for housing price prediction inspired us to experiment with variations in model selection and feature engineering. The similarity in project goals, and predicting housing prices, helped us draw connections and adopt best practices for model optimization and evaluation.

III. Dataset and Data Processing

A. Data Preprocessing

For testing we used the datasets from Kaggle that contained all of the input and output features required for our models [7]. While some of the features remained useful for training, many other features were unusable such as main road, basement, etc. These features used “yes” and “no” to indicate the status of the feature, but cannot be used for testing because most machine learning algorithms cannot handle categorical data directly, so they must be converted to 1 and 0 by using binary encoding to help process the data. The same can be said for the input feature, “furnishing status”, where it contained multiple categories such as furnished, semi-furnished, and unfurnished. To fix this issue we used one-hot encoding to create binary columns for each category to ensure that the data is numerical and is suitable for the models.

IV. Modeling Approach

In this work we explore different machine learning models and the techniques used to improve the models. These machine learning models include linear regression, decision trees, naive Bayes, neural networks, polynomial regression, and random forest regression that are adapted from the Sckit-learn library.

1) Linear Regression - It is a machine learning model that fits a model with coefficients to minimize the residual sum of squares between the observed targets in the dataset and the targets predicted by the linear approximation.
2) Decision Tree - A supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.
3) Naive Bayes - It is a type of probabilistic machine learning model based on Bayes’ Theorem. It is primarily used for classification tasks, and the “naive” part of the name comes from the assumption that the features are conditionally independent, which simplifies the calculations.
4) Neural Network - It is a type of machine learning model that is inspired by the structure and functioning of neural networks in the human brain. It is designed to recognize patterns and relationships in data.
5) Polynomial Regression - This is a type of regression analysis where the relationship between the input feature and output feature is modeled as an n-degree polynomial. This is also an extension of linear regression, in which it allows for modeling non-linear relationships between variables.
6) Random Forest Regression - This is a machine learning method that is used for regression tasks. It functions by combining the predictions of multiple decision trees to

improve the model's accuracy and generalizability.

7) Support Vector Machine - This is a powerful and versatile supervised machine learning algorithm that is used for classification and regression. SVMs are based on the concept of finding the optimal hyperplane that best separates the data into classes.

V. Experiments

When splitting the datasets into testing and training sets, we decided to choose the optimal sizes to capture the highest accuracy for each model utilized. For the Decision Tree model and Support Vector Machine model, we divided the data into 30% testing and 70% training. For the Neural Network model, the ratio was 20% testing and 80% training. Then, for the remaining models we divided the data to 40% testing and 60% training.

1) Linear Regression - After splitting the data and standardizing the features, the model was evaluated using common regression metrics. The results showed an R-squared score of 67.55%, indicating that the model explains approximately 67.55% of the variance in the target variable. The Mean Squared Error (MSE) was 1,506,230,725,917.46, with a Root Mean Squared Error (RMSE) of $1,227,285.92 and a Mean Absolute Error (MAE) of $902,975.64. These metrics highlight that the model captured a significant proportion of variance, which explains why it was our second-best model.
2) Decision Tree - The decision tree regressor model demonstrated a relatively poor performance. With an R-squared value of 42.79%, the model explained only 42.79% of the variance in housing prices, indicating that it failed to capture significant patterns in the data. The Mean Squared Error (MSE) was extremely high at 2.46 trillion, suggesting that the model's predictions were far from accurate. The Root Mean Squared Error (RMSE) of approximately $1.57 million indicates that, on average, the predicted prices deviated by over $1.5 million from the actual prices. Similarly, the Mean Absolute Error (MAE) of about $1.19 million reflects a substantial error margin, with the model being off by nearly $1.2 million on average. These results highlight the decision tree's struggle to effectively model the housing price predictions, especially when dealing with complex non-linear relationships in the dataset.
3) Naive Bayes -The results indicate moderate performance for the Naive Bayes model. The model achieved an R-squared value of 57.65%, explaining just over half of the variance in the housing price categories. While this value suggests some predictive power, it also reveals room for improvement. The Mean Squared Error (MSE) was 0.28, which suggests relatively small errors in the predicted values compared to actual ones. However, the nature of the dataset indicates that even small errors can lead to significant financial differences. The Root Mean Squared Error (RMSE) of $0.53 implies that, on average, the predictions were off by $530,000. Similarly, the Mean Absolute Error (MAE) of $0.17 reflects an average prediction error of about $170,000. Although lower than the decision tree, these metrics still highlight the challenges of accurately predicting housing prices, especially with a model like Naive Bayes.
4) Neural Network - When building the neural network model, this model used 1 input layer (responsible for receiving the dataset's features), 2 hidden layers (layers that capture the complex patterns and relationships between the input and output features), and 1 output layer (provides the final prediction for the housing prices). For each hidden layer, L2 regularization, also known as ridge regularization, was applied to help fix the overfitting issue by adding a penalty to the loss function based on the squared magnitude of the weights, which encourages the model to use smaller weights, leading to a simpler and more generalized model. After testing the dataset with this model it received a R-squared

score of 62.86%, a Mean Squared Error (MSE) of 1.87 trillion, a Root Mean Squared Error (RMSE) of $1,369,639.80, and a Mean Absolute Error (MAE) of $1,006,581.19. This Neural Network model performed well compared to the other models used for this experiment but did not achieve the best performance. Despite this, the model demonstrated strong predictive capabilities and its ability to generalize due to the use of L2 regularization, which makes it a valuable approach for this problem.

5) Polynomial Regression - After applying L2 regularization through Ridge regression, the polynomial regression model showed substantial improvement over other models tested in the experiment. With an R-squared value of 68.48%, this model explains nearly 68.5% of the variance in housing prices, making it the most accurate model in the experiment. Initially, without regularization, the model's performance was in the 40th percentile range, which was significantly poorer. The use of Ridge regression which introduces a penalty on the magnitude of the coefficients helped mitigate overfitting and stabilized the model's predictions. Additionally, trials with various polynomial degrees revealed that a 2nd-degree polynomial transformation provided the best results, striking a balance between capturing non-linear relationships in the data and avoiding excessive complexity within the dataset. Despite the model's improved performance, the Mean Squared Error (MSE) of 1.46 trillion and the Root Mean Squared Error (RMSE) of $1.21 million still indicate that the model's predictions have some degree of error, reflecting the challenges in accurately predicting housing prices. The Mean Absolute Error (MAE) of $894,967 suggests that on average, the model is off by almost $900,000, yet it still outperforms other models in this experiment.
6) Random Forest Regression - When developing the structure for this model, the hyperparameters needed to be tuned to best suit the random forest regressor by using grid search. Tuning the hyperparameters is crucial in optimizing the performance of this machine learning model to better ensure that this model operates at its best capacity. Once the hyperparameters are tuned and ready for testing, this model was able to reach an r-squared value of 66.31%, Mean Squared Error (MSE) of 1.56 trillion, Root Mean Squared Error (RMSE) of $1,250,468.19, and a Mean Absolute Error (MAE) of $902,726.41. Although this model was able to achieve the third highest scores because of its robustness to overfitting, the high RMSE and MAE indicate that this model is struggling with certain aspects of the target variable's variability. This can be due to the noise in the dataset, insufficient data preprocessing, or issues with feature engineering. Overall, the Random Forest Regressor demonstrates its capability as a strong baseline model in this project, especially for problems requiring a balance between accuracy and interpretability.
7) Support Vector Machine - Support Vector Machines are robust supervised machine learning models that can be used for both classification and regression tasks. For this experiment, we utilized the Support Vector Regression (SVR) algorithm. An SVR functions by finding a hyperplane that best fits the data while ensuring that errors within a certain tolerance are ignored. When training the model the kernel was set to rbf (rbf kernel maps the input data into a higher-dimensional space to capture nonlinear relationships), C (penalizes large deviations less, leading to a simpler, less overfitted model) to 1, and epsilon (determines the tolerance within which errors are ignored) to 0.1. After testing the dataset with this model, we received an R-squared score of -3.62%, a Mean Squared Error (MSE) of 4.46 trillion, a Root Mean Squared Error (RMSE) of $2,112,388.76, and a Mean Absolute Error (MAE) of $1,578,104.56. With a negative R-squared score and a very high MSE, RMSE, and MAE, this indicated that the model was unable to capture the complex relationships

within the dataset. Although this model should be one of the best-performing machine learning models out of all of the models used, due to the complexity of this machine learning model we failed to demonstrate its full capability for this experiment.

A. Figures and Tables

TABLE I. TABLE TYPE STYLES

Models	R^2 x 100	Mean Squared Error (MSE)	Root Mean Squared Error (RMSE)	Mean Absolute Error (MAE)
Linear Regression	67.55	1,506,230,725,917.46	$1,227,285.92	$902,975.64
Naive Bayes	57.65	0.28	$0.53	$0.17
Decision Tree	42.79	2,463,688,650,437.66	$1,569,614.17	$1,186,486.17
Polynomial Regression	68.48	1,463,188,621,192.95	$1,209,623.34	$894,967.45
Random Forest Regressor	66.31	1,563,670,698,302.80	$1,250,468.19	$902,726.41
Neural Network	62.89	1,875,913,192,473.16	$1,369,639.80	$1,006,581.19
Support Vector Regression	-3.62	4,462,186,287,255.04	$2,112,388.76	$1,578,104.56

Fig. 1. Results table indicating the percentage of accuracy and the metrics used for each of the models tested.

VI. Discussion

After testing the dataset that was collected previously for our many machine learning models, we noticed that some models functioned better than others, and some were not usable at all. Based on our observations, we detected that the linear regression model, polynomial regression model, random forest regression model, and neural network model yielded better results than the other models we tested because a linear regression model can capture a simple linear relationship between the input and output features, and the other models mentioned can capture the non-linear trends and patterns from the data provided. Although the R-squared score is low for each model, this is mostly due to the limitations of the dataset we were able to collect when conducting research for this project.

The next highest score was from utilizing the Naive Bayes model to predict housing prices, but the results from the test do not seem to be accurate due to this model being designed for classification tasks rather than regression tasks, and relying on assumptions (e.g. feature independence, Gaussian distributions) that do not align with the nature of housing price data.

Lastly, when utilizing the decision tree model to predict housing prices, we noticed that although it can technically predict housing prices, there are better machine learning models to use since there were many limitations when using this model such as difficulty handling continuous data, bias in splits, lack of robustness, etc.

Note: A support vector machine is suitable for predicting housing prices because of its ability to handle non-linear relationships, ability to handle small to medium datasets effectively, feature engineering flexibility, etc., but due to the computational complexity and requirements to fully utilize this machine learning model it can be difficult to find an accurate prediction for the housing prices.

VII. Conclusion and Future Work

In this project, we were able to explore various different machine learning models that could potentially aid in our search for the best time to buy or sell property in the future. This project identified their potential as well as limitations. Among the seven models tested, Polynomial Regression was shown as the most promising model in this application with an R-squared accuracy score of 68.48%. Followed by Linear Regression with an R-squared accuracy score of 67.55% and Random Forest Regression with 66.31%.

However, some of the models we used faced some challenges in this application. The Decision Tree model struggled to handle the complexity of the dataset, resulting in an R-squared score of 42.79%.

Naive Bayes, which was a model designed primarily for classification tasks, resulted in a 57.65% R-squared accuracy score but the nature of it being a classification model was not the best fit for this application. The Support Vector Regression model faced computational difficulties that limited its performance, culminating in a negative R-squared score of -3.62%. These results show that model selection is extremely important if you want accurate results.

In the future, we want to continue improving each model's performance and prediction accuracy. One thing that would drastically help us with this improvement is gathering a newer, larger, and more diverse dataset. Having such a disappointing accuracy score of -3.62% from the Support Vector Regression model, we plan to research and do more testing with this model as it may have potential in this application. Additionally, we'd like to utilize other models like gradient-boosted trees for larger datasets.

References

[1] M. Ritter, "California home prices are set to exceed $900,000. Here's what local real estate agents expect.," CoStar, 2024. https://www.costar.com/article/277028088/california-home-prices-are-set-to-exceed-900000-heres-what-local-real-estate-agents-expect (accessed Oct. 05, 2024).

[2] "Californians and the Housing Crisis," Public Policy Institute of California, 2019. https://www.ppic.org/interactive/californians-and-the-housing-crisis/

[3] A. Bentz, "California Housing Affordability Tracker (January 2024) [EconTax Blog]," lao.ca.gov, Jan. 24, 2024. https://lao.ca.gov/LAOEconTax/Article/Detail/793

[4] Ni Chuhan, "House price prediction based on different models of machine learning," *Applied and computational engineering*, vol. 49, no. 1, pp. 47–57, Mar.2024,doi:https://doi.org/10.54254/2755-2721/49/20241058.

[5] N. Khot, "HOUSE RESALE PRICE PREDICTION USING CLASSIFICATION ALGORITHM," 2021. Accessed: Oct. 24, 2024. [Online]. Available: https://www.irjet.net/archives/V8/i7/IRJET-V8I7419.pdf

[6] S. Bhushan Jha, R. Babiceanu, V. Pandey, and R. Jha, "Housing Market Prediction Problem using Different Machine Learning Algorithms: A Case Study." Available: https://arxiv.org/pdf/2006.10092

[7] "Housing Prices Dataset," *www.kaggle.com*. https://www.kaggle.com/datasets/yasserh/housing-prices-dataset