Exploring Redefining Prediction of Real Estate Prices

Cameron Gonzalez[1] **Susan Rathbun**[2] **Hannah Turner**[3]
Master of Applied Data Science
University of North Carolina at Chapel Hill
[1]Durham, NC [2]Cary, NC [3]Greenville, NC
[1]camgonzo@unc.edu, [2]susangr@email.unc.edu, [3]turnerha@ad.unc.edu

Abstract

This project aims to improve the accuracy of housing price predictions through the application of feature engineering on various regression and classification models. The goal is to establish an effective approach to predicting house prices which can be used on other housing price datasets in the future. Regression predictions are made using Linear Regression, Support Vector Machines (SVMs), Random Forest Regression, and XGBoost models. Classification predictions are made using Logistic Regression and Random Forest Classifier Models. The dataset, sourced from retail housing data in Southern Florida, will be analyzed using Python tools including scikit-learn for machine learning, as well as seaborn and matplotlib for visualization. The project will measure success by evaluating the accuracy and Mean Absolute Error (MAE) for classification models, as well as R^2, and Root Mean Squared Logarithmic Error (RMSLE) for regression models, with the objective of achieving a classification accuracy of 75% or higher.

1 Introduction

Accurately predicting housing prices is a significant challenge in the real estate industry, with direct implications for prospective home-buyers, sellers, and real estate agents. Our project aims to enhance price prediction accuracy by utilizing feature engineering and exploring multiple machine learning models, including Linear Regression, Classification, and Support Vector Machines (SVMs). The ultimate goal is to establish a baseline approach for effective housing price prediction that balances model performance and simplicity.

We will investigate the usefulness of Linear Regression, especially after utilizing feature engineering, assess the effectiveness of SVMs, and explore a classification-based approach to find the best approach for predicting housing prices based on a variety of different features. We will be investigating the effectiveness of working with classification models for our predictions by dividing our prices up into 'buckets' containing ranges of prices which will be utilized in the classification models.

Our dataset consists of retail housing data from the southwest region of Florida, and we will not be generating synthetic data. Due to the inherent variability in real estate datasets, we will be operating with an assumption of the features that should be included in a base retail dataset, such as square footage and the age of the house, even though the model may not be universally applicable. We will implement our models using Python, leveraging scikit-learn for machine learning, with Seaborn and Matplotlib for visualization. We will determine success by measuring the accuracy of the classification model, with careful consideration of the Mean Absolute Error (MAE) and finding the R^2 and Root Mean Squared Logarithmic Error (RMSLE) of our regression based models. We will also compare the metrics against current models to judge how much our techniques improve these predictions. Our goal is to achieve a classification model with an accuracy at or above 75% for predicting housing prices in Southwest Florida, while considering model simplicity.

Preprint. Under review.

2 Related Work

Given the importance of accurately predicting housing prices, a wide range of methods have been extensively explored. A quick internet search will show the vast majority of models for this task fall into the regression category, from simple Linear Regression to Hybrid Regression or a regression model with XGBoost (eXtreme Gradient Boosting), CatBoost, LightGBM (Light Gradient Boosting Machine), or another of many algorithms [1], [2], [4]. Support vector machines (SVMs) have also been used to tackle this problem, with varying success [4], [5]. More advanced methods have employed Artificial Neural Networks (ANNs) [3].

A common theme among the literature is utilizing a dataset based on one city or region. This is intuitive, as location is the undisputed most important factor in housing price prediction [2]. Some researchers have improved their models using feature engineering of location-based features. For example, in their dataset of houses in Beijing, researchers Truong, Nguyen, Dang, and Mei, added a feature that held the distance of each home from the center of Beijing [1].

As expected, various metrics are employed to test the predictive power of the housing price prediction models including Mean Squared Error (MSE), Root Mean Squared Logarithmic Error (RMSLE), and Mean Squared Absolute Error (MSAE) [1], [4]. Often, a simple linear regression model will perform with a R^2 value between 0.60-0.70 [4], [6]. Models employing more complex methods like XGBoost, LightGBM, or Hybrid Regression have achieved a RMSLE near 0.113 [4].

As mentioned before, the most important factor for predicting housing prices is location [2]. Our dataset contains real estate information from homes in Southwest Florida, however, no additional information about the location is provided. While some successful models have utilized feature engineering on the location related features, that approach is not feasible for our dataset because we do not have access to those features. Thus, we will shift the problem away from predicting all real estate prices with given features and towards building a model that accurately reflects the state of housing prices in Southwest Florida, *independent of individual location*.

3 Methods

Before beginning any analysis, we preprocessed data by converting non-numerical features into integer outputs. Missing values had been previously removed from the dataset and we found no further data cleaning necessary, besides removing the "sqft" feature which was redundant to the "Living Area" feature. To evaluate the effectiveness of feature engineering on our housing dataset, we developed a variety of models and compared their performance.

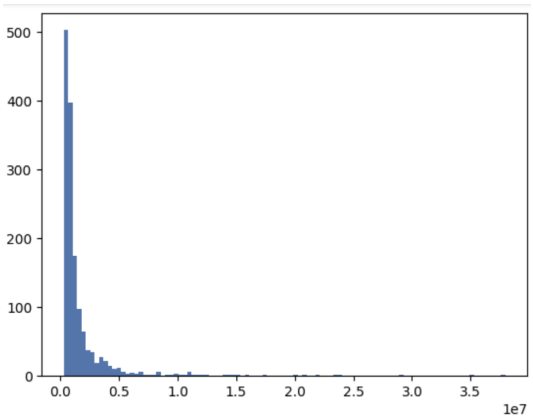

Figure 1: Histogram of price distribution

We used various metrics to evaluate our different types of models. For our regression models, which are our Linear Regression, Random Forest Regression, SVM Regressor, and XGBoost models, we used the following metrics:

R-Squared (R^2)

The R^2 metric, also known as the coefficient of determination, measures how well the predicted values approximate the true values. An R^2 of 1 indicates perfect predictions, while an R^2 of 0 indicates no correlation between predicted and actual values.

Root Mean Squared Logarithmic Error (RMSLE)

The Root Mean Squared Logarithmic Error (RMSLE) is a metric that measures the ratio of predicted values to actual values in a logarithmic scale. A lower RMSLE score indicates better model performance. A score of 0 would indicate perfect predictions. Higher RMSLE values indicate more significant errors. RMSLE tends to penalize underestimations more than overestimations.

Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) measures the average of the absolute differences between the predicted and actual values. MAE treats all errors equally and is less sensitive to large errors compared to Mean Squared Error (MSE). Like MSE and RMSLE, a lower MAE indicates better model performance.

We use the following metrics to evaluate the performance of our classification models, our Logistic Regression, SVM Classifier, and Random Forest Classifier models:

Accuracy

Accuracy simply measures the percentage of total predictions which were correct, but does not consider how close the prediction was to the actual result.

Precision

Precision for each class measures the proportion of true positive predictions for that class out of all instances predicted as that class. Precision is calculated for each class separately and specifically measures the amount of false positives for each class. A higher precision means there are fewer false positive predictions.

Recall

Recall is similar to Precision, except it measures the true positive rate rather than the false positive rate for each class. A higher recall value for a class means that there are fewer false negatives for predictions of houses that belong in that class.

F1 Score

The F1 score is the harmonic mean of Precision and Recall scores, where a high F1 score indicates a better performing model.

Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) for our classification models functions very similarly to its usage to measure the performance of regression models. Using MAE on our classification model takes into account how far away the prediction was from the actual bucket the property belongs in, providing more insight than a simple accuracy score. This metric penalizes the model less for a prediction which is close, but not exactly correct. A MAE of 0 means perfect predictions, a MAE of 1 means the model is, on average, 1 bucket off, a MAE of 2 means the model is 2 buckets off, etc.

Prior to feature engineering, we created a basic Linear Regression containing all features except "ID". This model achieved an R^2 value of 0.658 and a RMSLE of 5.254. We added a “House Age” feature, which captures the total age of the house since construction, a “Base Rooms” feature which includes both the number of baths and bedrooms, compared the ratio of living area space to the total area of

the house as “Living Area Ratio”, and squared the living area, “Living Area Squared”, to further capture nonlinearity. As observed in figure 2, the added “Base Rooms” feature appears to capture an additional piece of information not captured by “Bedrooms” and “Baths” alone and is positively correlated with the price of the houses. We constructed a Linear Regression model that included the feature engineering and observed better performance, with an R^2 value of 0.692 and a RMSLE of 4.164.

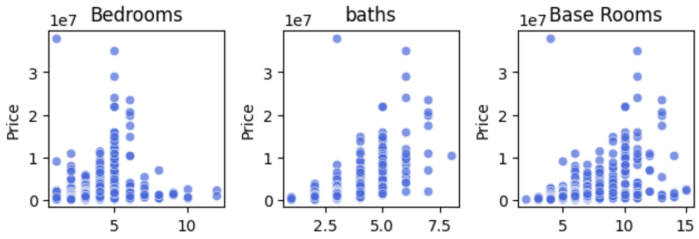

Figure 2: Visual of relationships captured with feature engineering

Heatmap of features after feature engineering

Based on the heatmap we created, we were able to observe the correlation and relationship between features. Specifically, we can see that price is highly correlated with the features ‘Living Area’, ‘Total Area’, and ‘Living Area Squared’. The engineered feature, ‘Base Rooms’, also shows a moderate positive correlation with price, capturing the combined effect of both the number of bedrooms and baths, contributing to the overall size and functional utility of the home. While engineered features like ‘House Age’ and ‘Living Area Ratio’ may not show strong correlation with the outcome variable in the heatmap, they improve interpretability and help reduce multicollinearity among the features by transforming and summarizing existing information in meaningful ways.

Figure 3: Correlation matrix after feature engineering

To acknowledge the results found in the literature and obtain a better understanding of our data, an XGBoost model was built. This allowed us to more accurately evaluate our model performance and establish a baseline. One source compared XGBoost, CatBoost, and SVMs to simple Linear Regression and found the most accurate Mean Squared Percentage Error (MSPR) from the SVM at 0.187 and the highest R^2 with the CatBoost model at 0.89 [1].

Lastly, we built a Random Forest Regressor and tested it with feature engineering to see if we could further improve model performance. We observed better performance than with the Linear Regression model however, a slightly lower R^2 value than our Random Forest Regressor without feature engineering.

We then shifted our approach, viewing the problem through the lens of classification. In order to do this, we modified our dataset to place each property into a specific "bucket" based on its price. We used two different techniques to classify our data into buckets. The first was creating quantile-based buckets, or dividing the properties evenly into each bucket so that each bucket contained the same number of properties. We compared the accuracies of our Logistic Regression model on various bucket counts shown in figure 4, and settled on 9 buckets as an ideal balance of the number of buckets compared to prediction accuracy. The higher the number of buckets we have defined, the smaller the price ranges in each prediction, meaning a more useful prediction overall, however as the number of buckets increases the overall accuracy of these predictions decreases.

The second technique we experimented with was dividing each property based on price ranges to reflect how housing prices are typically divided in the market. For our manually defined price range, we used the following methodology to define the price ranges for each bucket:

- 7 buckets based on price ranges spaced evenly every \$100,000 between \$250,000 and \$950,000
- One bucket containing the price range defined by \$950,000 $\leq$ price < \$2,050,000
- The remaining bucket containing all prices greater than or equal to \$2,050,000

This method for dividing based on price allows us to evenly set price ranges at the lower price values, where the data points are far more concentrated, while having larger price ranges at the higher prices, where data is far more spread out. This manual separation of buckets also contains 9 total buckets, which is consistent with the ideal number of buckets in our quantized buckets.

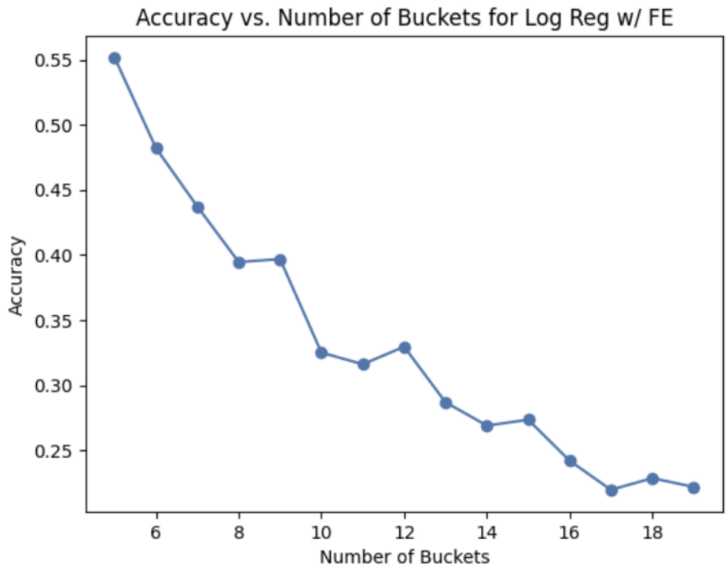

Figure 4: Number of buckets vs accuracy

The advantage of the quantized technique was its even distribution, meaning that the model would weigh each bucket equally as each one contained an equal number of properties, while the second technique allowed the buckets to more accurately fit the non-linear distribution of our data, where many more properties were contained in the lower price ranges than the higher ones.

4 Results

4.1 Regression Results

We evaluated the impact of feature engineering across our models and found some consistent improvements in performance. For our base Linear Regression model, feature engineering increased our R^2 score from 0.658 to 0.692, while reducing the RMSLE from 5.254 to 4.164. This suggests that our feature engineering had a positive impact on the performance of our Linear Regression model.

The XGBoost model performed moderately with an R^2 of 0.733 and a RMSLE of 0.364. This was much better than the SVM model which came in at an R^2 of -0.095 and a RMSLE of 0.837. The negative R^2 score indicated the model performed worse than a model predicting the mean price for every single house.

The Random Forest Regressor achieved strong results even before feature engineering, with an R^2 of 0.867, a mean absolute error (MAE) of 446078.863, a RMSLE of 0.330, and a MSE of 936434.809. After feature engineering, we observed slightly poorer performance, however, still a relatively accurate model. Specifically, our results were an R^2 of 0.849, a MAE of 469526.278, a RMSLE of 0.336, and a MSE of 992225822937.485. This suggests that for the Random Forest Regressor model, the feature engineering did not lead to higher performance than without, despite its success with other models.

Lastly, we applied the Random Forest Regressor model on a real estate data set based in California. The model performed exceptionally poorly with an R^2 of -10.754. This means the model predicted worse than a model guessing the mean price for every house. This result was anticipated as real estate pricing prediction is dependent on location and the same house in California is much more expensive than in Florida.

Regression Model Performance Table:

	R^2	RMSLE	MAE
Linear Reg w/o FE	0.658265	5.25439	833327
Random Forest Reg w/o FE	0.86691	0.32969	446079
Linear Reg	0.691693	4.16431	740439
Random Forest Regressor	0.849408	0.335548	469526
XGBoost	0.733212	0.363915	529136

Classification Model Performance Table:

	Accuracy	Mean Absolute Error	Precision	Recall	F1
Logistic Regression B1	0.496644	1.10067	0.492949	0.496644	0.462861
Logistic Regression B2	0.412752	1.00671	0.390739	0.412752	0.395602
Random Forest Classifier B1	0.483221	0.90604	0.487467	0.483221	0.480142
Random Forest Classifier B2	0.47651	0.815436	0.468358	0.47651	0.468776

Figure 5: Model performance summary

4.2 Classification Results

Both models trained using our manually defined price ranges outperformed our models trained on quantized price ranges in terms of the simple accuracy score.

The Logistic Regression model using the manually defined price range resulted in a test accuracy score of 49.6%, while the model using the quantized price ranges had an accuracy of 41.3%. This accuracy score simply calculates the number of correct predictions the model made, without taking into account how close its incorrect predictions were to the actual result. The Mean Average Error (MAE) score allows us to see this, with the two models scoring an MAE of 1.100 and 1.000 respectively. This means that the predictions from the model using manually defined ranges were, on average, 1.1

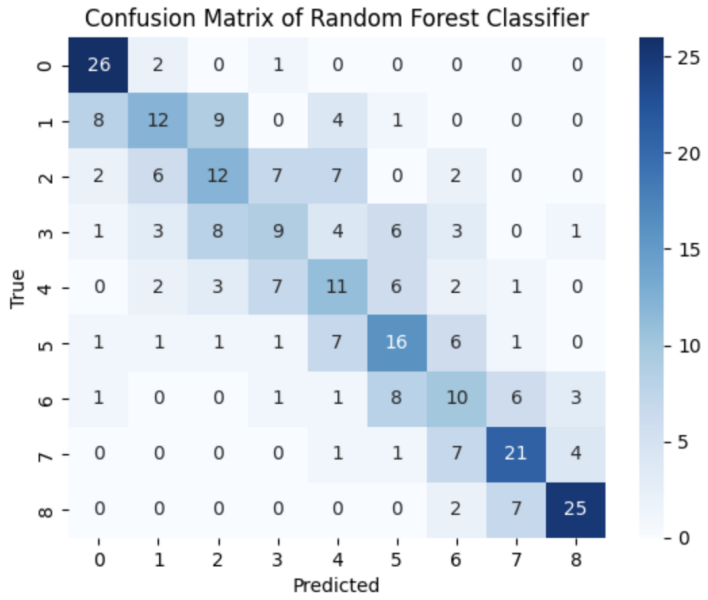

Figure 6: Confusion matrix of Random Forest Classifier

buckets away from the correct result, while the predictions of the model using quantized buckets was, on average 1.0 buckets away from the correct prediction. While the model using manually defined price ranges was correct with its predictions more often than the other model, its incorrect predictions were, on average, further away from the correct result than the model using quantized buckets.

The Random Forest Classifier performed very similarly to Logistic Regression, achieving 48.3% accuracy with manually defined buckets and 47.6% accuracy with quantized buckets. The MAE scores for the two models were 0.906 and 0.815 respectively. The Random Forest Classifier Models resulted in similar accuracy scores to the Logistic Regression Models, however its significantly lower MAE scores mean that, on average, its predictions were closer to the actual results more often than the Logistic Regression Models. As a result, the Random Forest Classifier using quantized price ranges was the best performing model overall.

A model making completely random predictions would have a prediction accuracy of 11.1%, or $\frac{1}{9}$, as there are 9 different possible values for classification. Our models all performed significantly better than a completely random model, however both accuracy and MAE can likely be improved with further testing and optimization.

5 Conclusion

The best performing model was the Random Forest Regressor (RFR) without Feature Engineering (FE). The FE likely did not affect the RFR as Random Forest models already have a means of capturing non-linearity. The FE did, however, improve the Linear Regression model. Our goal of producing a classification model with an accuracy of 75% or greater was not met.

It has been established that the most important feature for predicting housing prices is location, thus, when specific location features are not available (like longitude and latitude or zip code) we recommend building models focused on regions. This will capture the state of housing prices in that region independent of individual location.

As Linear Regression models are widely accessible, they offer a great starting point for building a housing price prediction model. If choosing to work with a linear regressor, FE should be considered, utilizing any and all location related features. Otherwise, a Random Forest Regressor should be

considered, without the need for FE. Since Random Forest models are more complex, they require more computational power. Thus, the trade-off between model performance and resources should be considered when building a housing price prediction model.

As we were unsuccessful in optimizing a classification-based approach for housing price prediction, we do not recommend this approach without further refinements.

Further exploration could include calculating a model "exchange" rate for more accurately comparing models in locations with inflated markets. Another avenue worth pursuing is a hybrid model, combining the strengths of the regression and classification approaches to achieve more robust predictions.

References

[1] Quang Truong, Minh Nguyen, Hy Dang, and Bo Mei. 2020. Housing price prediction via Improved Machine Learning Techniques. *Procedia Computer Science 174* (2020), 433–442. DOI:http://dx.doi.org/10.1016/j.procs.2020.06.111

[2] Qingqi Zhang. 2021. Housing price prediction based on multiple linear regression. *Scientific Programming* 2021 (October 2021), 1–9. DOI:http://dx.doi.org/10.1155/2021/7678931

[3] Hasan Selim. 2009. Determinants of house prices in Turkey: Hedonic regression versus artificial neural network. Expert Systems with Applications 36, 2 (March 2009), 2843–2852. DOI:http://dx.doi.org/10.1016/j.eswa.2008.01.044

[4] Deepakshi Mahajan. 2024. House price prediction using machine learning in Python. (September 2024). Retrieved March 20, 2025 from https://www.geeksforgeeks.org/house-price-prediction-using-machine-learning-in-python/

[5] Jirong Gu, Mingcang Zhu, and Liuguangyan Jiang. 2011. Housing price forecasting based on genetic algorithm and Support Vector Machine. *Expert Systems with Applications* 38, 4 (April 2011), 3383–3386. DOI:http://dx.doi.org/10.1016/j.eswa.2010.08.123

[6] P. Ravindra, Katta Meghana, Gattu Bhavitha, and Donkada Neelima. 2020. HOUSE PRICE PREDICTION USING ADVANCED REGRESSION TECHNIQUES. *Journal of Engineering Sciences* 11, 6 (June 2020), 1084–1089. https://jespublication.com/upload/2020-1106157.pdf