Introduction

On the 13th of January 2023, Zachary Corradino and Samantha Reyes filed an antitrust lawsuit against eighteen major players in the Florida real estate market[1]. They claim that these property owners and managers colluded to fix the price of rents using RealPage AI across major metropolitan areas including Jacksonville, Miami, Orlando, and Tampa. This comes on the heels of a 20% surge year over year in property values[2] in Miami-Dade County in September 2022, the highest in any city in the United States and in a county where rents have increased by more than 60% since early 2020[3]. Simultaneously, median property values have soared by 32%.

Compared to other markets, Miami's real estate market sees a huge number of foreign buyers with a priority for luxury homes. About 60% of luxury homes are bought in cash[4]. On the other hand, large real estate investment funds target large single-family homes to subdivide and rent out as multiple units. This leaves a limited supply of homes for the first-time homebuyer in Miami. The silver lining is that there is less competition for this constrained supply. These homes receive less than a handful of offers, and a deal is struck within 60 days[1]. For buyers looking at homes within this subset of the Miami market it becomes even more important to understand the property value. While listing websites such as Realtor and Zillow offer their own in-house public estimates, this information does not do enough to equip buyers with knowledge of the property and can often favor sellers.

The main goals of this project are to:

1. Accurately predict/estimate the selling prices of real estate in Miami

2. Determine the features that have the greatest influence on price

State of the Art

On Kaggle, there are several users who have performed some exploratory data analysis on Miami housing prices, but only one that has truly attempted to predict the prices with several models. User Vijayaragavan.s.k fit the data to a linear regression model and several ensemble models. The result of their analysis is tabled below.

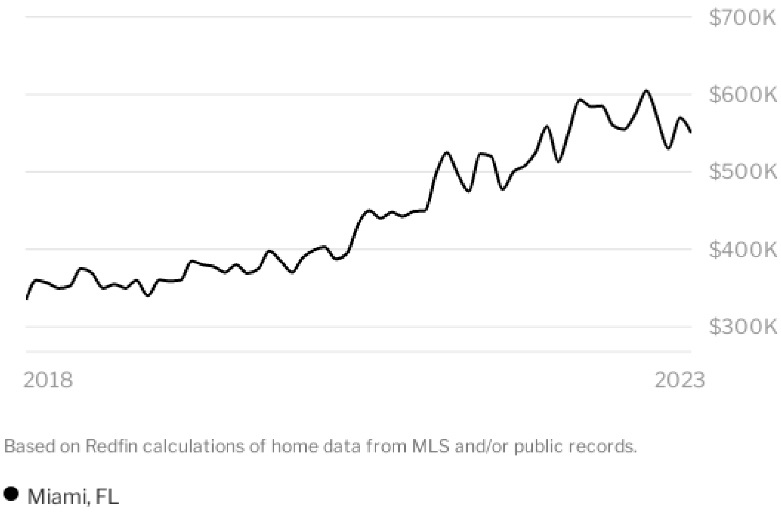
Model	R^2	RMSE ($)
Linear Regression	0.641442	197254.08
Decision Tree	0.775201	156186.38
Bagging Ensemble	0.865749	120699.32
Random Forest	0.870875	118372.79
Gradient Boost Ensemble	0.845584	129447.00
CatBoost Regressor	0.886907	110780.51
LGBM Regressor	0.881951	113181.95

From the table, it can be seen that the CatBoost Regressor model has the lowest Root Mean Square Error (RMSE) and highest R^2 value. Therefore, that was the model with the best performance.

There are many approaches that have been researched to predicting house prices outside of Kaggle as well. In 2021, Chen et al found that using deep neural networks, Naive Bayesian, AdaBoost, and SVM methods allowed for a high degree of accuracy[6]. The best performing method was SVM which yielded an R2 value of 0.9908. Another approach was researched to predict housing prices was researched by Piao et al[7]. Their research is notable because it uses many of the same variables that are contained within the Miami housing dataset, but for homes in China, and because their aim was not only to predict the sale price of homes but to also determine which feature had the greatest influence on prices. Their research found that the housing area was the most important predictor when using XGBoost and using a tuned CNN model yielded an R^2 of 0.9868.

There are several companies who have a vested interest in being able to accurately predict and estimate the selling prices of real estate in any city. The Miami real-estate market is the seventh largest in the U.S. and is the second largest housing market in the southeastern region[8]. Real estate companies are not the only type of company interested in the Miami housing market, as home prices are of interest to many other companies like insurance companies for policy information, banks for appraisal information, and governments for potential property taxes. Reports are generated every year that typically track previous time point prices and forecast future housing prices. The chart below was generated specifically for the Miami real-estate market by Redfin, a residential real estate brokerage.

Median Sale Price of Single Family Homes in Miami, FL



Based on Redfin calculations of home data from MLS and/or public records.

Charts such as these are made for public consumption and as a result sacrifice deep insights for the sake of interpretability. In the chart the median sale price is the only variable being tracked. A more detailed report can be found on the Redfin website here: https://www.redfin.com/city/11458/FL/Miami/housing-market.

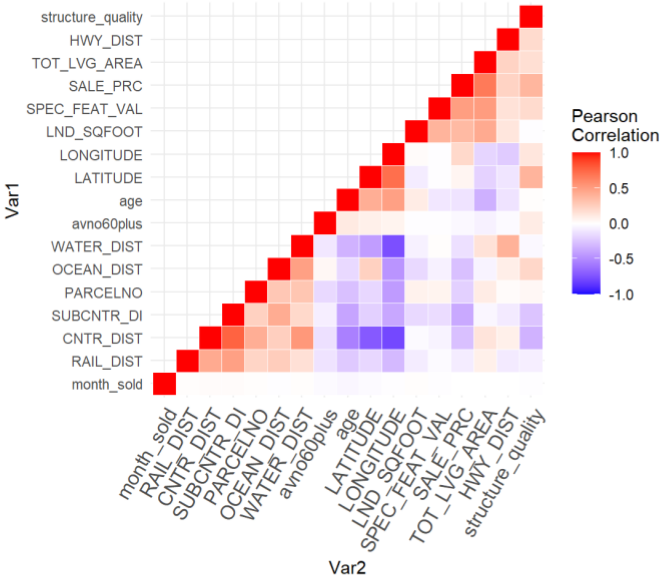
Many people who wish to sell or buy a home can search for a reasonable estimate of what the home price may be, but typically do not have much insight into why it may be that price. Part of that problem is what we seek to answer with our analysis of the housing market data set and give some insight into what features have an influence on the price of a house. There are other external factors that can also affect housing prices such as inflation and other trends that affect the nation at large.

Methods

The dataset contains 13,932 instances of single-family homes sold in Miami. It is linked below: https://www.kaggle.com/datasets/deepcontractor/miami-housing-dataset?resource=download\
There are no missing values in the dataset. The dataset contains the following columns:

- **Categorical variables**
  - PARCELNO: unique identifier for each property. About 1% appear multiple times.
  - avno60plus: dummy variable for airplane noise exceeding an acceptable level
  - month_sold: sale month in 2016 (1 = January)
- **Numerical variables**
  - SALE_PRC: sale price ($)
  - LND_SQFOOT: land area (square feet)
  - TOT_LVG_AREA: floor area (square feet)
  - SPEC_FEAT_VAL: value of special features (e.g., swimming pools) ($)
  - RAIL_DIST: distance to the nearest rail line (an indicator of noise) (feet)
  - OCEAN_DIST: distance to the ocean (feet)
  - WATER_DIST: distance to the nearest body of water (feet)
  - CNTR_DIST: distance to the Miami central business district (feet)
  - SUBCNTR_DI: distance to the nearest subcenter (feet)
  - HWY_DIST: distance to the nearest highway (an indicator of noise) (feet)
  - age: age of the structure
  - structure_quality: quality of the structure
  - LATITUDE
  - LONGITUDE

There was no pre-processing necessary as the data had no missing values. Before building any models, correlations were run as a sort of preliminary analysis, to give an idea of which input variables may be correlated to the sale price of a home, and to give an idea of any correlations between input variables, such as distance to ocean and distance to nearest body of water. The correlation matrix is presented below.



Before further analyzing the data, we created a training and test set to separate the training of the model, and then test the model's generalization. The training set contains 80% of the samples in the data set, and the testing set contains the remaining 20% of the data set. The actual number of samples in the training and testing set were 11073 samples and 2859 samples, respectively.

For each analysis, the sale price was regressed against all the other variables, except for the parcel number, as the parcel number was a unique identifier for each home and was irrelevant for analytical purposes. It can be seen in the code that it is intentionally excluded from every analysis.

Evaluation and Cross-Validation

Each of the following models was evaluated based on its proportion of variance in the sale price that is explained by the model (R^2) and by the model root mean square error (RMSE), which in this case can be directly interpreted as a model's average dollar amount error when predicting the sale price. Lasso regression has cross-validation built into its function in R. 10-fold cross-validation was performed on all of the other models.

Multiple Linear Regression

Multiple regression is a statistical method that models the relationship between a dependent variable (response) and several independent variables (predictors) to predict the value of the dependent variable. The predictors can be either numerical or categorical. The response is a continuous variable that we aim to predict based on the values of the predictors.

The response is modeled as a linear combination of the predictors as shown by the following equation: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$. The beta coefficients indicate the slope of the relationship between the response and each individual predictor. The greater the slope, the greater the impact. Linear regression is a highly biased model, and the results will only be meaningful under the assumption that the variables are linear, that each observed data point is not affected by the other observations (i.e., independence), and that the variance of the residuals is the same at any given value of the predictors.

For our data set, multiple regression was calculated using sale price regressed against all the input variables (except for Parcel number). The summary of the model gives useful information such as estimates of the values of the coefficients, the standard error of the coefficients, the t-value (a measure of how many standard deviations the estimated coefficients is away from the hypothesized value of 0), the p-value (the probability of observing a coefficient as large as the one estimated if the true coefficient is 0), the AIC value, and the model RSS. Dividing the model RSS gives by the number of observations yields the Mean Squared Error, which was used to compare the effectiveness of the model to others.

Lasso

Lasso (Least Absolute Shrinkage and Selection Operator) is a linear regression technique that is useful for variable selection and regularization. To accomplish this, it adds a penalty term to the standard linear regression cost function, or the RSS. Lasso works in the following manner:

1. A penalty parameter λ controls the amount of shrinkage applied to β, the vector of regression coefficients. The cost function is given by: $RSS + \lambda|\beta|$

2. $|\beta|$ is the sum of the absolute values of the regression coefficients. It encourages sparsity in the coefficient estimates by pushing many of them towards zero. Consequently, some coefficients may be exactly zero, removing the corresponding variables from the model. This is how lasso performs feature selection.

3. The value of the penalty parameter λ determines the strength of the penalty and controls the amount of shrinkage. A higher λ results in a stronger penalty and more shrinkage, leading to a simpler model with less predictors. Conversely, a lower λ results in less shrinkage and a more complex model with more predictors.

Lasso regression is helpful for both variable selection and regularization. By setting some coefficients to zero, Lasso effectively chooses only the most important variables, making it an ideal technique for high-dimensional data. Additionally, by shrinking the coefficient estimates towards zero, Lasso can help reduce overfitting and improve the generalization performance of the model.

Since our dataset contains 15 possible predictors, we used lasso regression to see which predictors had the greatest impact on the sale price and which could be removed from the model altogether. By default, LASSO is cross-validated. As in the previous model, we evaluated the performance of lasso regression by comparing its MSE with the others.

After running several linear regression models, we ran a series of decision-tree methods to see if the non-linear models would better capture the data. All ensemble methods were put through 10-fold cross-validation in order to

Random Forest Regression

Random forest regression is a technique that predicts numerical values based on input variables by combining multiple decision trees. Decision trees make are used to make predictions by splitting the data into different groups based around their features. Random forest regression works by creating many decision trees, i.e., a forest of trees.

1. The algorithm selects a random subset of input variables, also known as features, and builds a decision tree based on that subset.

2. It repeats this process several times, each time choosing a different random subset of features and building a new decision tree.

3. To predict a new input's value, the algorithm averages the predictions made by all the decision trees.

Using multiple decision trees helps to reduce overfitting and improve the model's accuracy by making predictions based on different sets of features. Taking the average of all the trees' predictions results in a more accurate and stable estimate of the target value.

The default model uses a value of m = p/3 to calculate the number of variables tested at each split. With 15 predictors, this would mean a split of 5 variables. This parameter value was confirmed using the TuneRF function for hyper parameterization. The model was then trained on the training data and used to predict values for the test set. The model's performance was evaluated using the test MSE, and a plot was generated to show the predicted versus actual values. Additionally, a variable importance plot was created to identify the variables that had the greatest impact on the decision trees' node purity and mean squared error.

Boosting

Boosting is a machine learning technique that creates a strong model by combining multiple weak models. It does this by repeatedly training a series of weak models on the same dataset, where each model learns to correct the errors of the previous model.

To start, an initial weak model is trained on the data. This model is then used to make predictions on the data, and the errors of the predictions are calculated. A second weak model is trained on the same data, but with more focus on the samples that were poorly predicted by the first model. This process is repeated for a series of weak models, where each model corrects the errors of the previous models.

The weak models are then combined to create a strong model by taking a weighted average of their predictions. Boosting is a powerful technique that can often outperform other machine learning algorithms, but it can lead to overfitting if the weak models are too complex or if the data is noisy. To prevent overfitting, regularization techniques such as early stopping or shrinkage can be used.

The boosting model was trained using the training data, and a summary was created which included a variable importance plot. The shrinkage parameter used for the boosting model was set to the default value of 0.001. To correct for this small shrinkage parameter, 5000 trees were used during training to ensure high accuracy. The test MSE was calculated for this model, as with the previous models, to compare their performance. Ultimately, this model was optimized using 10-fold cross validation.

Bagging

Bagging is a technique that involves training multiple independent models on different subsets of the training data and then combining their predictions to make a final prediction. This method of ensemble learning is used to improve the overall accuracy and robustness of the model while reducing the variance of individual models.

To utilize bagging, a random subset of the training data is selected with replacement to create a new training dataset of the same size as the original dataset. A base model, in our case a decision tree, is trained on the new training dataset. This process is repeated multiple times to create a set of independent models. When making a prediction on new data, each model is used to make a prediction, and the predictions are combined using averaging, voting, or other aggregation techniques.

The bagging model was trained on the training data using the default settings of 500 generated trees and m = p, the total number of predictors. To evaluate the model's performance, the test MSE was calculated by comparing the predicted values with the actual values in the test set. The error rate was plotted against the number of trees generated to visualize the model's accuracy with increasing trees. Furthermore, a variable importance plot was generated to compare variable importance with other decision tree methods.

Results

Multiple Linear Regression

Our initial MLR model looked at the sale price of the home against the rest of the predictors with the exception of the Parcel Number. We can see from the coefficient summary table that two predictors are not significant: the distance from the water and the month sold. The Miami real estate market is relatively stable, and the data does not need to be seasonally adjusted, which can explain why the month sold has no effect on the sale price. The distance from the water not being considered significant could be due to a correlation with a similar predictor: distance from the ocean. These two variables have a correlation coefficient of 0.4907.

We then tested this model against another MLR that excluded the two variables above. Their respective AIC values are 299053.8 and 299050.4. The two values are similar, as are their RMSE at $166,839 and $166,857, respectively. Their R^2 values are also similar at 0.6952707 and 0.6953094.

Lasso

Since there is some correlation between some of the predictors, we ran a lasso regression to determine the features that had the greatest impact on sale price and to potentially build a simpler model.

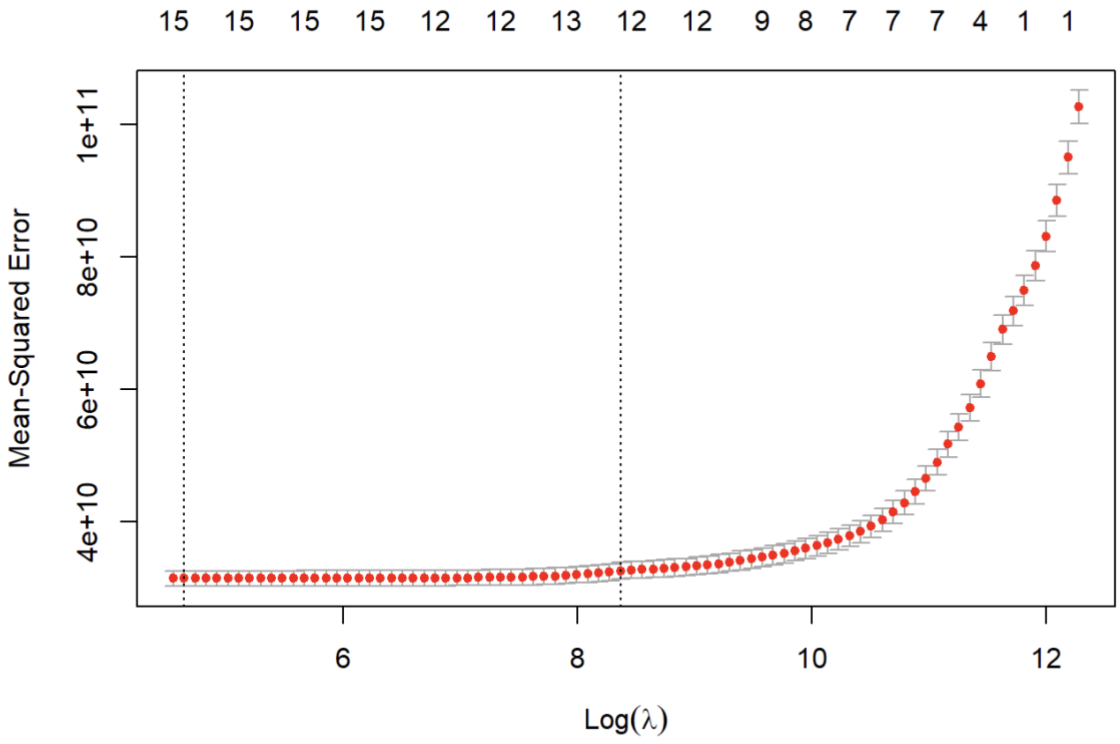


Fig 1. MSE vs log of penalty parameter. The smaller the parameter the more features included in the model.

The figure above shows us that the model with the lowest MSE again includes all 15 predictors. However, the difference between a model with 15 predictors and one with 12 predictors is small. The RMSE for the lasso regression with 15 predictors is $166,808 and the R^2 value is 0.6956288, comparable to the two previous MLR models. From the coefficient summary, however, we can start to see a pattern emerge. Two variables with the largest effect are the total living area and the structure quality. Because there are predictors that have higher degrees of correlation, linear regression models may not be the best choice of model. Instead, we applied nonlinear models to our data.

Random Forest Regression

The first nonlinear model was a random forest regression with the number of variables tested at every node set to a third of the predictors, or in this case 5 total predictors, and 5000 trees. As previously mentioned, this parameter was determined through hyper parameterization. We also ran a variable importance plot from the model.

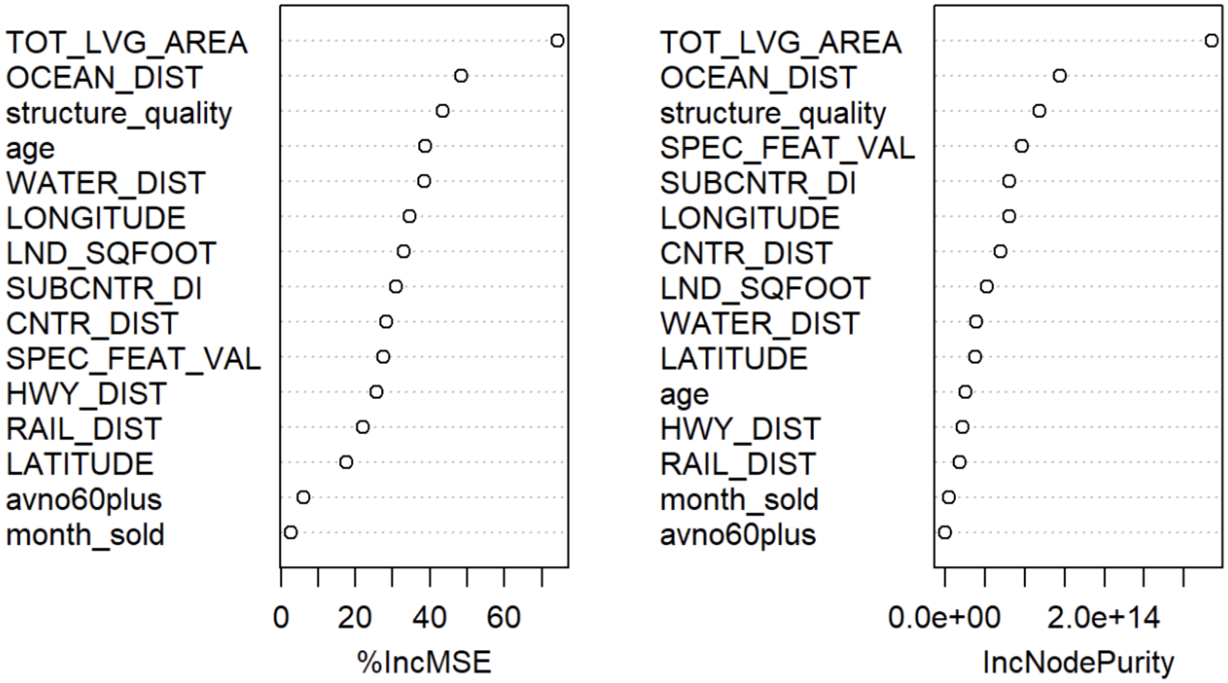


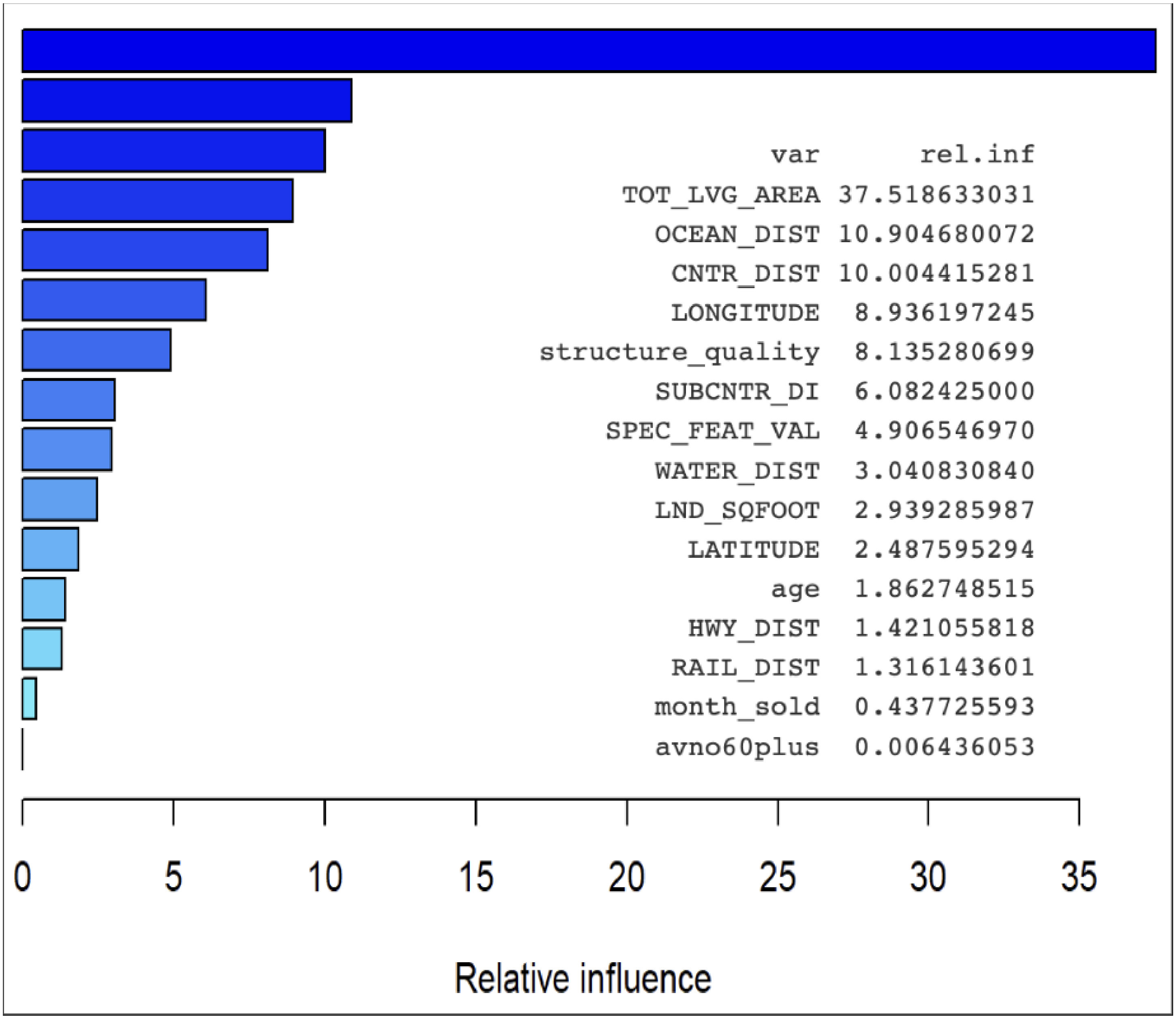
Fig 2. Plot showing the predictors that have the greatest impact on predicting the sale price.

Like the MLR models, the random forest model determined that total living area and structure quality had a big effect on the sale price. The random forest model also revealed that the distance from the ocean impacts the sale price.

The RMSE for the random forest regression model is $84,261, about half of the value calculated from the MLR models. The ability to explain the variance in the sale price also increased significantly, from approximately 69% across all the linear regression models to about 90%. The exact R^2 value of the model is 0.9064777.

Boosting

We continue analyzing the data with boosting methods and graph plots of feature importance. As discussed in the methods section, the shrinkage parameter was set to 0.001 and the model was built with 5000 trees. 10-fold cross validation was performed on the model.



```
                var      rel.inf
      TOT_LVG_AREA 37.518633031
        OCEAN_DIST 10.904680072
         CNTR_DIST 10.004415281
         LONGITUDE  8.936197245
 structure_quality  8.135280699
       SUBCNTR_DI  6.082425000
     SPEC_FEAT_VAL  4.906546970
        WATER_DIST  3.040830840
        LND_SQFOOT  2.939285987
          LATITUDE  2.487595294
               age  1.862748515
          HWY_DIST  1.421055818
         RAIL_DIST  1.316143601
        month_sold  0.437725593
        avno60plus  0.006436053
```

Fig 3. Influence of predictors on sale price. Top bar represents the total living area.

Continuing the trend from the previous models, the total living area continues to be the predictor with the greatest impact on the sale price. The RMSE obtained by the boosting method is $79,567 and the explained variance R^2 is 0.9312604, or 93%.

Bagging

We see if these results can be improved by bagging methods, but the RMSE actually increases to $84,335 which is worse than the random forest model. To understand the decision making by the model, we generate an importance plot.

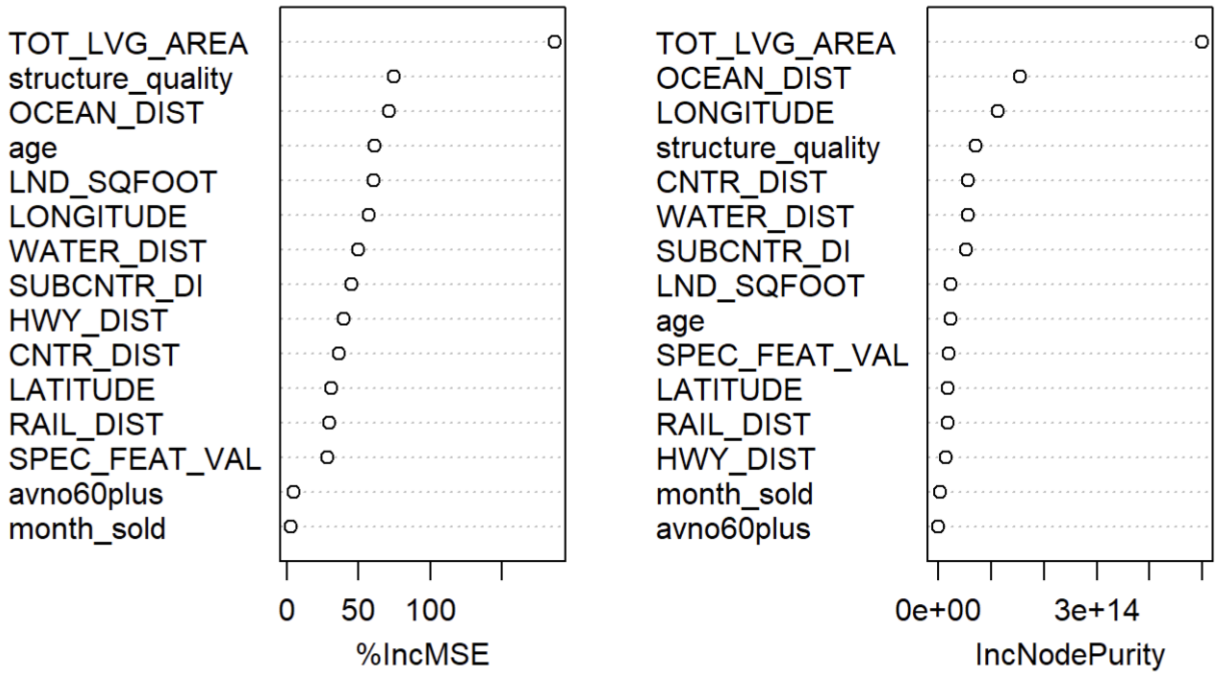


Fig 5. Importance plot for bagging method

Here we see that bagging methods find structure quality to be the second most important feature, whereas the boosting method had it 5th behind the distance to the ocean, distance to the center, and the longitude. The reason boosting might see those three variables as more important is because there is some correlation between them. The further east the property is located (longitude), the closer it is to the city center and the closer it is to the ocean.

Summary of Results

Model	R^2	Test RMSE ($)
Linear Regression (all variables)	0.6952707	166839.60
Linear Regression (only significant variables)	0.6953094	166857.20
LASSO	0.6956288	166808.90
Random Forest	0.9064777	84261.85
Boosting	0.9312604	79567.15
Bagging	0.9046859	84335.83

Conclusion

On the whole, linear regression models underfit the predicted sale price. No model ever got lower than an average error of $166,800 and more than 30% of the variance in the sale price could not be explained. This is likely because linear regression models are highly biased. These types of models have a lot of assumptions that are not met by the data. In particularly, the data must be linear, and each variable must be independent of on another. As was shown in the correlation matrix, some of the variables have correlations with other variables that are of higher magnitude that 0.5. The data for housing sale prices is too complex to be captured by a model that is as biased and simple as a linear regression model. Not even the introduction of a penalty could significantly change the outcome of linear regression models.

On the other hand, non-linear ensemble methods performed much better. All of the methods decreased the RMSE by a factor of 2 and were able to explain at least 90% of the variance in the sale price. In particular, boosting methods performed the best out of the three ensemble methods, and the best overall across all methods that were tested. Our research also outperformed the methods detailed by Vijayaragavan.s.k in their analysis.

An average error of $79,567 seems like a big error when it comes to buying a home. It can be the difference between getting a really great deal on a home or being denied a mortgage. However, this is an average and averages can be affected by large errors, as shown below.

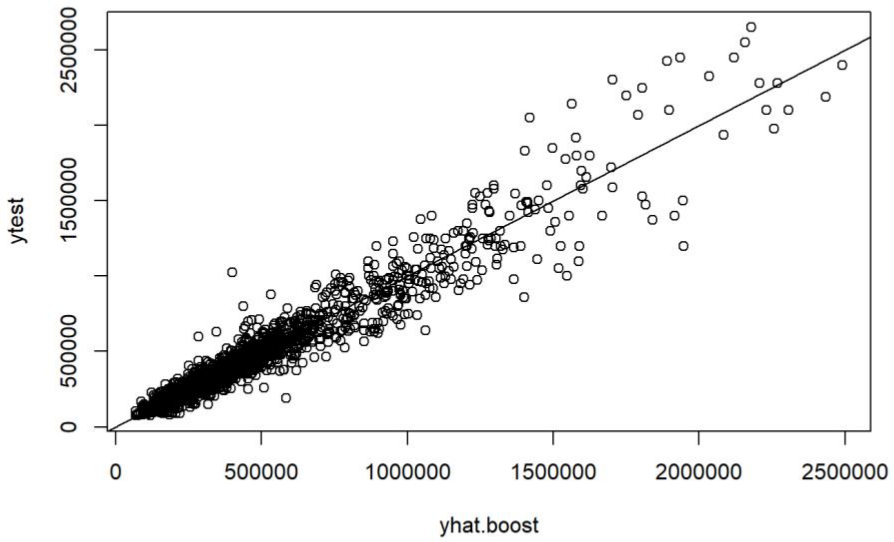


Fig 6. Actual Values vs Predicted Values by Boosting Model

The boosting model does a much better job of predicting the sale price of homes whose value is $500,000 or less. As the value of the home increases, the error in predicted sale price becomes larger and larger. An error of $79,000 on a home that is valued at $250,000 could be devastating on buyers (and profitable for sellers), whereas on multi-million-dollar homes that amount of money becomes less significant for both the buyers and sellers.

For future works, we can test the ability of neural networks to predict prices in Miami. We can also build on our work with regards to feature importance by getting better understanding of the variables that affect the price of homes on the higher end of the value spectrum.

References

[1] *Corradino et al v. RealPage et al*, 25 No. 17 (S.D. Florida. 2023). Retrieved from https://assets.inman.com/wp-content/uploads/2023/01/RealPage_Florida_Lawsuit.pdf

[2] *Miami-Dade County, FL Housing Market: House Prices & Trends - Redfin*. (2023). Retrieved from https://www.redfin.com/county/479/FL/Miami-Dade-County/housing-market

[3] *Average rent in Miami, FL*. Zumper. (2023) Retrieved from https://www.zumper.com/rent-research/miami-fl

[4] Garcia, D. (2023, February 28). *Miami-Dade pending home sales and showing appointments rise*. MIAMI REALTORS. Retrieved from https://www.miamirealtors.com/2023/02/21/miami-dade-pending-home-sales-and-showing-appointments-rise/

[5] Vijayaragavansk. (2021, December 31). *Miami House price(rfe+hyperparameter tuning)*. Retrieved from https://www.kaggle.com/code/vijayaragavansk/miami-house-price-rfe-hyperparameter-tuning

[6] Chen, Y., Xue, R., & Zhang, Y. (2021). House price prediction based on machine learning and Deep Learning Methods. *2021 International Conference on Electronic Information Engineering and Computer Science (EIECS)*. https://doi.org/10.1109/eiecs53707.2021.9587907

[7] Piao, Y., Chen, A., & Shang, Z. (2019). Housing price prediction based on CNN. *2019 9th International Conference on Information Science and Technology (ICIST)*. https://doi.org/10.1109/icist.2019.8836731

[8] Redfin. (2023). *Miami Housing Market: House Prices & Trends*. Retrieved from https://www.redfin.com/city/11458/FL/Miami/housing-market