How Much is Your House Worth?

Introduction

In October of this year, a landmark court case, Sitzer v National Association of Realtors (NAR), exposed the NAR, America's largest trade association, for colluding with other real estate brokerages to artificially inflate home sale prices. The motivation behind this unethical practice was to boost the overall sales price, consequently increasing the commission earned by both buyer and seller agents. In the traditional real estate compensation model, where agents receive a percentage of the sales price (typically between five and six percent), there is an inherent incentive for agents on both sides to drive up prices, maximizing their earnings. This court case not only reignited debates about the fairness of the existing compensation structure but also revealed concrete evidence of collusion by the NAR in inflating prices. Consequently, there is now a strong push for reform, with discussions revolving around alternative compensation models such as flat rates, hourly fees, variable fees, or even decoupling the compensation from the sales price. Beyond the immediate issue of compensation, the broader concern is the erosion of trust among people seeking a buyer agent to represent them. We feel there can be great value created by developing algorithms to determine the true value of a home, alleviating the apprehensions of potential homebuyers.

This issue arises during a challenging economic backdrop within the real estate market. With thirty-year mortgage rates soaring to a 23-year high of 7.79%, there is a substantial demand-supply imbalance, leading to escalating prices. This inflation is exacerbated by a high Fed interest rate implemented to counter inflation, along with disruptions caused by COVID-19, which delayed construction and inflated material costs. In such a dynamic market, people urgently need a reliable method to ascertain the accurate price of a house they intend to buy or sell. In response to these challenges, our initiative aims to develop a machine learning algorithm capable of comprehending various house characteristics to provide precise price predictions. The algorithm will be trained on extensive past sales data, considering close to eighty different house features to enhance the accuracy of its predictions. To inform our approach, we delved into three notable studies.

Related Works

In a noteworthy MDPI study, Alejandro Baldominos and his team delved into economic factors to enable real-time pricing analysis. Their objective was to uncover underpriced houses and identify lucrative investment opportunities. The study revealed that regression trees outperformed other models, including k-nearest neighbors, support vector machines, and neural networks, demonstrating superior performance based on error metrics.

In a parallel exploration, an ASCE journal study employed a sophisticated approach, utilizing a deep belief-restricted Boltzmann machine and a distinctive nonmating genetic algorithm. This innovative methodology aimed at predicting housing prices not only during the design or construction phase but also factored in economic patterns, offering a holistic perspective.

Similarly, in Ping-Feng Pai's MDPI journal study, which mirrors our own initiative, machine learning was applied to estimate pricing based on historical transactions. Notably, least squares support vector regression emerged as the most accurate model, surpassing the performance of classification and regression tree (CART), general regression neural networks (GRNN), and backpropagation neural networks. Through our comprehensive approach, we aspire to provide a solution that not only addresses the immediate challenges in the real estate market but also cultivates trust and transparency in the intricate processes of buying and selling homes. These studies did not focus on seeing which variable was the most important but instead focused on which model was the best.

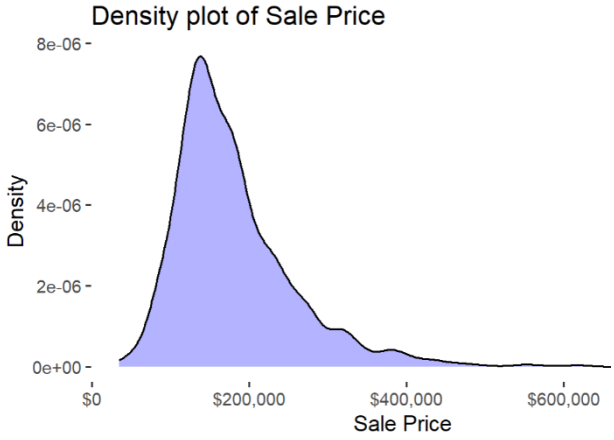
Data Description

The data for this project was from a Kaggle Competition Dataset. It consisted of 1460 observations and 79 explanatory variables. The datasets include both categorical and numerical variables that primarily focus on the physical aspects of the house such as size, condition, and number/type of rooms. We were interested in creating a model that was able to predict the SalePrice for houses.

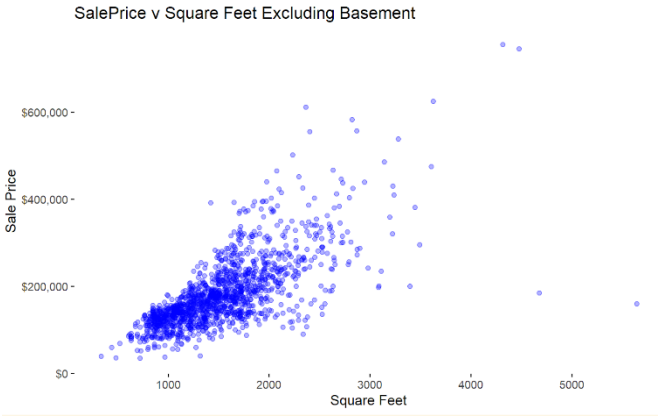
The key characteristics of SalePrice are as follows.
Min. : 34900 ## 1st Qu.:129975 ## Median :163000 ## Mean :180921 ## 3rd Qu.:214000 ## Max. :755000

One of the pre-processing steps taken prior to running the XGBoost was converting features from our dataset to dummy columns. We used the dummy_cols command from the fast dummies package to accomplish this goal. Another pre-processing step we did was a data partition that split our data into train and test validation sets. We used a 60-40 split.

Our data exploration began by looking at a density plot of our target variable of sale price. From this graph, we were able to see the distribution of Sale Prices. It appears that the majority of Sale Prices are around $180,000 with a few houses reaching into the $400,000 range. However, we decided our data was not skewed enough to run a log transformation. This could have been a possible source of error when we created our models.

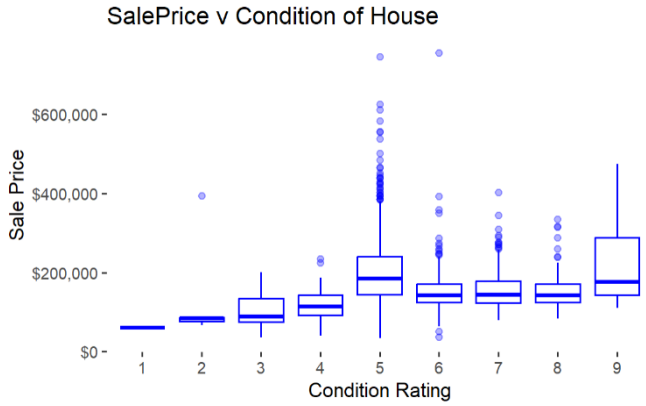


Next, we wanted to get an idea of the relationship between sale price and square feet excluding basement. Based on the following graph it appears that as square feet excluding basement increases, the Sale Price also increases.



In addition, we wanted to look at the relationship between sale price and number of full bathrooms. This resulted in a positive relationship where more bathrooms led to higher sale prices.

The last variable we explored was the condition of the house. The general pattern from our visualization was that as condition ratings increase, the sale price also increases.



Method

We planned to create three different models and let them compete for the most accurate. We chose Bagging, Random Forest, and XGBoost as our models. Each of these models does well with datasets that have a lot of variables, each is efficient, and each is robust. We ran each model separately, tuning each one using a for loop and evaluating success via RMSE results.

A bagging model is a shorthand for bootstrapping aggregation. Basically, how the model works, is it randomly creates subsets or bags of the data via sampling with replacement. Each subset generates a model where a decision tree is trained based on the selected data. Each subset is independent of the other. Once all subsets are finished it averages each subset's generated model or predicted result to find a final aggregate prediction. This is a useful technique as it helps reduce variance via averaging multiple trials. This also leads to overfitting reductions as each subset has different data present and therefore cannot overfit to the overall training data set.

Random Forest models are very similar to Bagging models, but they add another element of randomness. Random forest models limit the amount of predictive variables present to train each subset model. Bagging has one hundred percent of predictive variables present to train the model, but random forests have a fraction of the total that had been randomly sampled. This is useful as it lets the model decipher how influential each predictive variable is. It also stops individual predictive variables from dominating over every subset. Random forest models build off what bagging models do and can be more accurate.

The final model we used was XGBoost (Extreme Gradient Boosting) with regression tuning. This was suitable for our dataset because our goal was to predict the sale price of houses, which is a continuous variable. XGBoost begins by creating a naive model and calculating the errors from the first attempt. Next, a new model is created that predicts the errors and reweighs incorrect samples to rebuild the model. This model is added to the ensemble and the process is repeated until the models converge or a specified number of trees is reached. The advantage of building trees sequentially instead of independently (random forest) allows for corrections to be made on the previous trees to refine predictions and improve overall accuracy. The first step we took in utilizing this method was converting the data to DMatrix format. We then began tuning the parameters of the model to achieve optimal performance. This involved creating a vector for max depth and min child weight that was then used in a for loop to run in the algorithm to find the optimal levels of each parameter. We repeated this process to find the optimal gamma as well. Lastly, we did eta tuning to determine the optimal learning rate for our model. This was determined by looking at a graph of the error rate vs. the number of trees for each learning rate. Based on the graph, we selected the optimal eta and number of rounds. We then ran a final model using the optimal number of each parameter and used these results to make our final predictions and calculate the RMSE. In addition, we were interested in looking at the SHAP values to determine the impact of each variable while also considering interactions with other variables. It also allowed us to look at the direction of the relationship between the variables and the sale price.

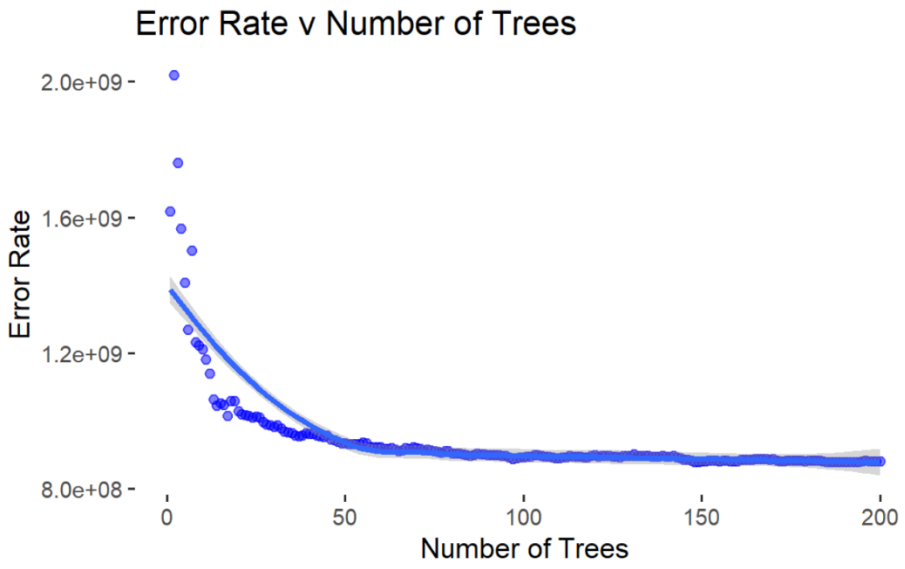
Results

A Bagging, Random Forest, and XGBoost were all executed, tuned, and compared to find the most accurate model. Starting with the Bagging model we ran the model with a rough estimate of 200 trees. We then compared this to the test result and graphed the error rate to the number of trees. The results are shown below. Our takeaway was the error rate started to flatten out around 100 trees. It is extremely flat with 200 trees. There is no downside to running more

trees except computational cost, therefore we decided 200 trees was an acceptable number of trees to run. To make sure this was a valid claim we included it in our parameter tuning. Parameter tuning simply refers to testing a range of parameters to find the most accurate. We looked at varying node size and number of trees, the goal was to see the lowest RMSE (root mean squared error) based on the combination of the two variables. RMSE was chosen as a key metric because it shows how far off the model's prediction was from the actual result. The squaring and un-squaring aspect of RMSE is utilized to find the absolute value of the difference between results. A for loop was completed and a heat map was analyzed to decipher the results. The map is clear to decipher, the number of trees does not affect RMSE, but the Node Size does. Node Size refers to the number of observations required to split and start a new branch. We found a Node Size of one to be the best option and the lowest error. Therefore we found our tuned bagging model to have a Node Size of one, and the number of trees to be 200. When comparing these model predictions to actual results we found an RMSE of 35,972.96 or in other words, our prediction on average was $35,972.96 off the mark. The mean sale price of a house in the data set was $179,318 so bagging on average was 20.06% off.

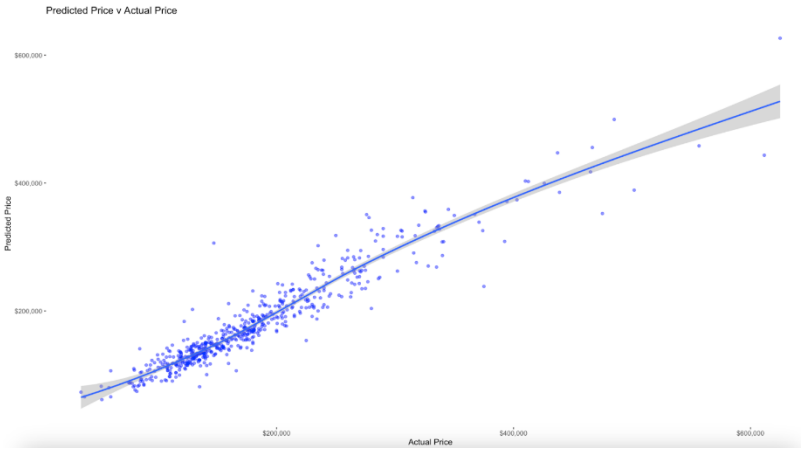
Next, we looked at the random forest model to continue our progression. Random Forest models and Bagging models are mostly the same, except Random Forest models vary the number of variables present to create the prediction trees. Therefore we utilized the same 200 Trees and 1 Node Size, then we ran the Random Forest model through a for loop testing the best number of variables present. The same method was taken looking at RMSE to evaluate success and the number of variables was chosen to be 64. It generated an RMSE of $35,933.29 which on average was 20.03% off. In terms of observational importance we found overall quality to be the most important factor. The rest are shown in graph below labeled bag_mod_4.

The results of our parameter tuning for XGBoost led to min child weight of 3, max depth of 5, gamma of 0, nrounds of 100, and an ETA of 0.1. We used these values in our final model and created predictions using the model. These predictions resulted in an estimate for RMSE of $28022.7, which meant that our model was 13.6% off.

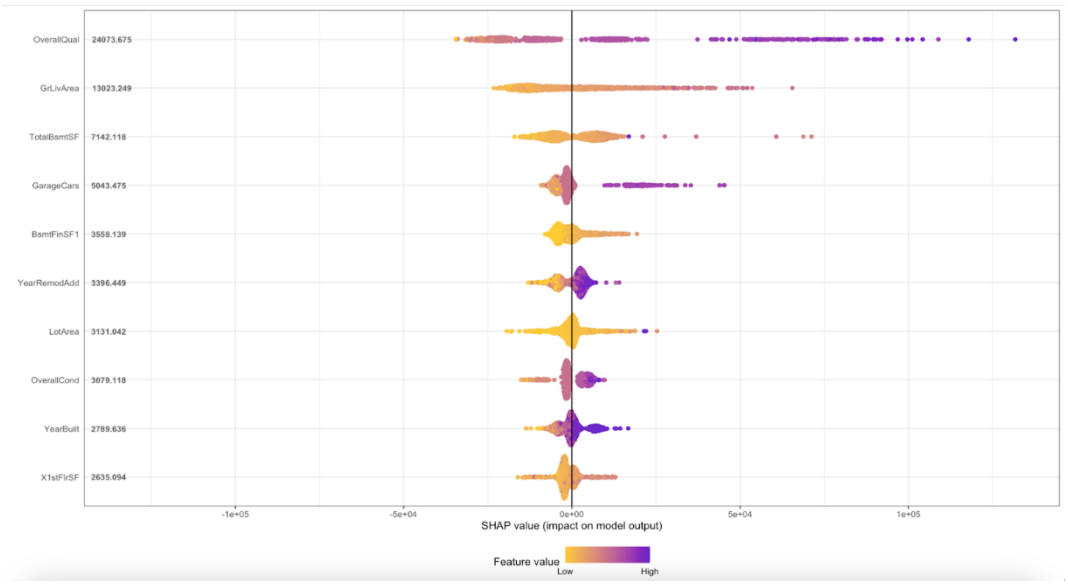


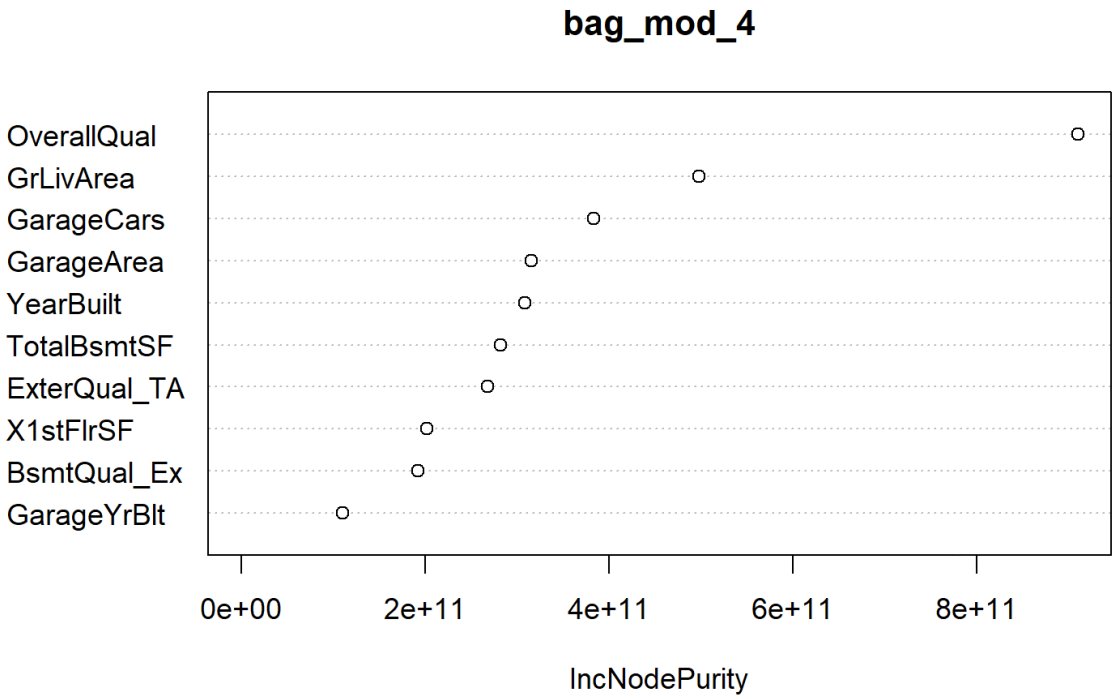
Discussion

After comparing the three models, it became evident that the XGBoost was the most accurate predictor with the lowest RMSE. The process of building trees sequentially and making corrections proved to have a superior performance over the other models. Even though Bagging and Random Forest had respectable accuracy (with Random Forest being better than Bagging), their average prediction deviations were higher than that of XGBoost. Since XGBoost was the best model, we used it to make predictions for the sale prices of houses. We plotted the predicted vs. actual price to visualize the performance of the XGBoost model. Our graph indicated that our model was relatively accurate in predicting sale prices under $200,000. However, as sale price increases, the points are distributed further from the line. This indicates that the model did not perform as well for higher sale prices. An action that can be taken based on these results is training our model using higher sale prices. Looking back at our density plot for sale price we can see that the majority of our data centered around $180,000 with a slight tail reaching into the $400,000 range. We decided it was not skewed enough to do a log transformation, but this is something that can be done in the future.



We also calculated the SHAP values for the variables and found that the top five most impactful variables were OveralQual, GrLivArea, TotalBsmtSF, GarageCars, and BsmtFinSF1. These results were similar to the results from bag_mod_4 which showed the top five impactful variables of OveralQual, GrLivArea, GarageCar, GarageArea, and YearBuilt. The insights gained from these visualizations was that Overall Quality was the most important predictor of sale price. We also saw that one of the variables from our initial data exploration, overall condition, was listed as one of the most important variables from our SHAP values. These are all factors that should be considered when homebuyers are trying to determine the true value of potential homes.





Conclusion

In conclusion, XGBoost was the most accurate model with an RMSE of $28,022.7 or on average 13.6% off the correct value. We found in both XGBoost and Random Forest that Overall Quality was the most important predictor of house value. When tuning our models we used RMSE as a guide.

In reality, the model we created can be utilized by home buyers to educate themselves on a rough price estimate. The model is not accurate enough to give an exact price. This makes

sense, selling and buying homes is an art and not a science. There are many uncontrollable factors in the housing market. Desperation and emotions can often be tied into the transaction. The initial problem stated was one cannot trust a buying real estate agent as they have an incentive to raise the final price. It is unlikely that this model can help decipher if an agent is artificially raising the price as this model is not accurate enough, and such an act can be extremely hard to prove. As stated earlier it can still be useful to get a general understanding of what the value of a home is.

If the process was to be continued we would like to have a larger data set with a larger range of home values. It would also be interesting to incorporate more locational data into the data set such as comparing cities or beachfront properties. We would also be interested in looking at how macroeconomic conditions impact the price of a home. When interest rates are high mortgage rates are high and the market freezes up.

Contributions

The Work was split up evenly.

Broghan: cleaned data, random forest model, bagging model
Wrote up: intro, related work, Conclusion

Kristina: data visuals, XGBoost model - organized office hours
Wrote up: Data / Description, Discussion

Both: Methods, Results

Bibliography

Studies Sources
https://ascelibrary.org/doi/full/10.1061/%28ASCE%29CO.1943-7862.0001047?casa_token=VyYumiKtJgMAAAAA%3Ar6GK0KWsiZQTATizDqFHZXgTtIsHEo66_eScE9prfaHQDTe5Mx2QSeXHEnIfWVeYseypBgqCl8CM
https://www.mdpi.com/2076-3417/8/11/2321
https://www.mdpi.com/2076-3417/10/17/5832

Economic Sources https://realestate.usnews.com/real-estate/articles/what-the-2-billion-realtor-lawsuit-means-forhomebuyers-and-sellers https://www.forbes.com/advisor/mortgages/real-estate/housing-market-predictions/

Data Source - https://www.kaggle.com/competitions/house-prices-advanced-regressiontechniques/data?select=train.csv

R code in attached documents.