Kaylynn Mosier
February 22, 2025

Housing Price Prediction Using Regression

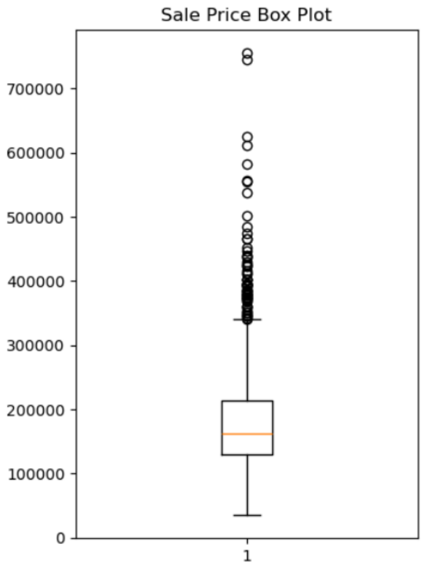
Background

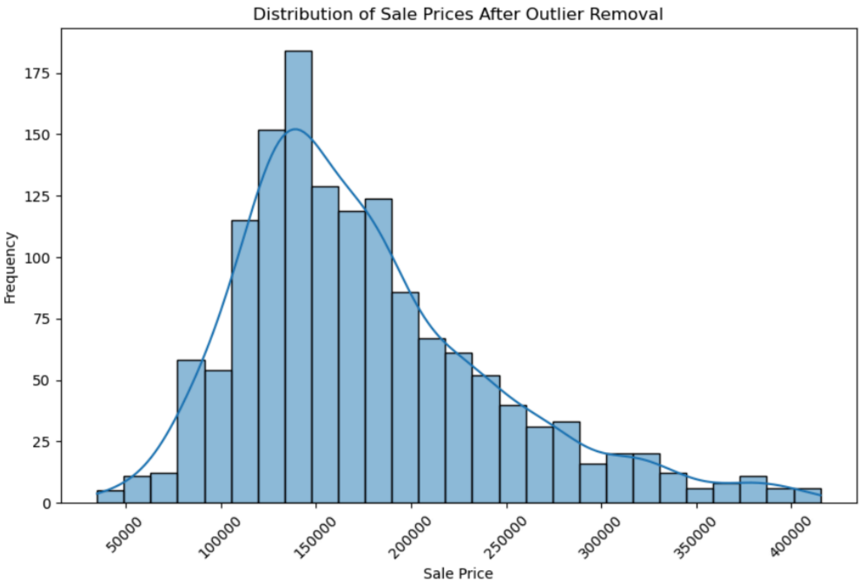
Accurately estimating house prices is essential for both the buyer and seller. Buyers want to sell their hose for as much, or more than it's worth, to profit the most while buyers want to ensure they are not overspending. Currently, appraisers are utilized to estimate house prices. However, those individuals can be swayed in their pricing by outside factors. To keep both parties interest in mind, a real estate office would benefit from a machine learning model that can accurately predict housing prices.

Data Explanation & Preparation

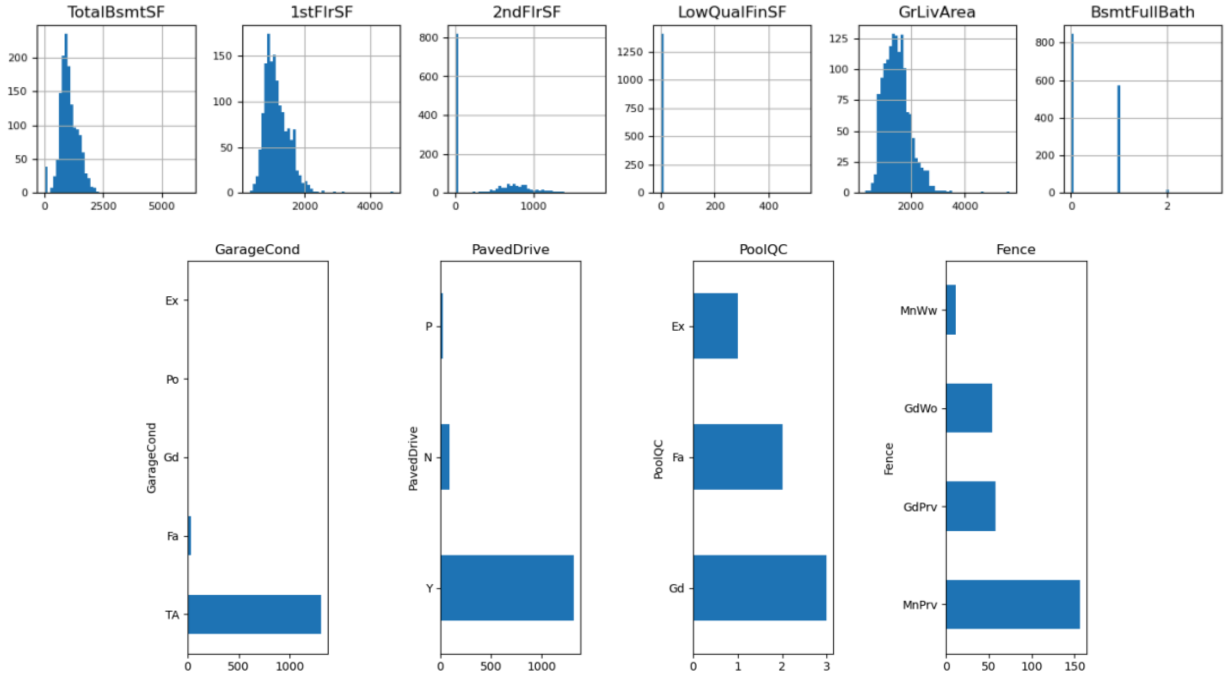
The dataset I am utilizing is the Ames Housing dataset containing information about nearly 3,000 houses sold in Iowa between 2006 and 2010. The target variable in this dataset is the price the house was sold for. Numerous versions of this dataset exist online. The version I used, was originally from a Kaggle challenges and is already split into training and test set with the intent that users will create a model using the training set and then predict unknown values with the test set. Predicting values from the test set is outside the scope of this project so, I utilized only the training set.

My first step in this project was to gain a better understanding of the target variable; sale price. To begin, I plotted the frequency of each sale price to visualize the distribution of values. That visualization revealed sale price had a normal distribution. However, a long tail on the right side of the plot indicated there were likely outliers. Plotting a boxplot revealed there were some significant outliers in the data. To handle this problem, I removed values that had a z-score greater than 3. These values fall more than 3 standard deviations from the mean and are considered statistical outliers in a normal distribution. The below plot shows the distribution of sale price values after outlier removal.





This dataset contains a mixture of ordinal categorical, nonordinal categorical and numerical variables, so preparing the data for modeling was extensive. To begin, it was useful to plot the distribution of numerical and categorical variables. This helped me decide what steps needed to be taken. Due to the large number of variables, I have included only a small snippet of plots created. Numerical features were plotted using a histogram while categorical features were plotted using a bar plot.



After visualizing the numerical features, it was evident there were large amounts of missing data. Visualizing categorical features revealed some features contained values for only one category. These features will add little to the model and were removed in later steps. There also appeared to be many missing values in the categorical values as well, but this was more difficult to tell without digging further into these variables.

Numerical Features

The first step I took in cleaning numerical variables was to analyze the correlation of each numerical variable with the target variable. A feature with very low correlation to the target will not increase the effectiveness of the model and will often introduce noise into the model that can decrease accuracy. I removed any features that had correlation values between -0.24 and 0.24.

I then had to deal with the missing values in remaining numerical features. I did not want to simply remove all columns or rows that contained missing values because that would have greatly decreased the size of my dataset. I removed any columns that had more than 80% of their values missing. I then filled any remaining missing values with the median of that column.

The final step in preparing numerical features was to apply a scaler to the values. Features that were numerical before any data preparation steps had much higher values that features that began as categorical and had been encoded. In the model building stage, there is a possibility the model could consider features with higher values as more important than feature with lower values. To counteract this, I applied a scaler to all features that were numeric in the original dataset. This scaled and normalized all values.

Categorical Features

The categorical features in this dataset were composed of both ordinal and non-ordinal features. Non-ordinal features were recoded using one hot encoding, where each category of the feature is assigned an arbitrary value between 0 and one minus the number of categories. For example, a feature with four categories, would end up with 0.0, 1.0, 2.0, and 3.0 instead of categories.

The ordinal features were encoded by hand using ordinal encoder. This encoder maintains ordinality of categories by encoding lower ranked categories as a lower value and higher ranked categories as a greater value. For example, for a column with values of NA, poor, moderate, and great; NA would be encoded as 0.0, poor as 1.0, moderate as 2.0, and great as 3.0. Ensuring each category was encoded properly required thorough analysis of feature documentation provided with the dataset. This process also revealed that many NA values in the categorical features occurred not because the values were missing, but because that house did not have that feature. For example, when grading garage quality, houses that did not have a garage received and NA in this column. Had I removed all NA values at the beginning of analysis, powerful predictive capabilities would have been lost.

Methods & Analysis

As mentioned previously, the goal of this analysis is to predict sale price of houses. This is process will involves predicting a discrete variable and is therefore a regression task. There are many powerful regression models that could have been used. I utilized linear regression, random forest regression, decision tree regression, and gradient boosting regression. Each model was built, used to predict the target variable, and evaluated using root mean square error (RMSE) and r-squared (R2). Those values are included in the table below.

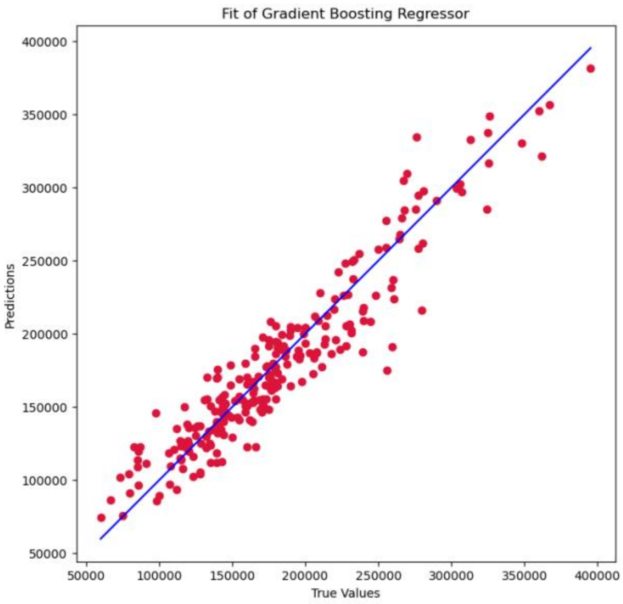
Model	RMSE	R2
Linear Regression	2391204052514976.5	-1.5362709016955883e+21
Random Forest Regression	21252.27	0.8786
Decision Tree Regression	34212.02	0.6855
Gradient Boosting Regression	19975.70	0.8927

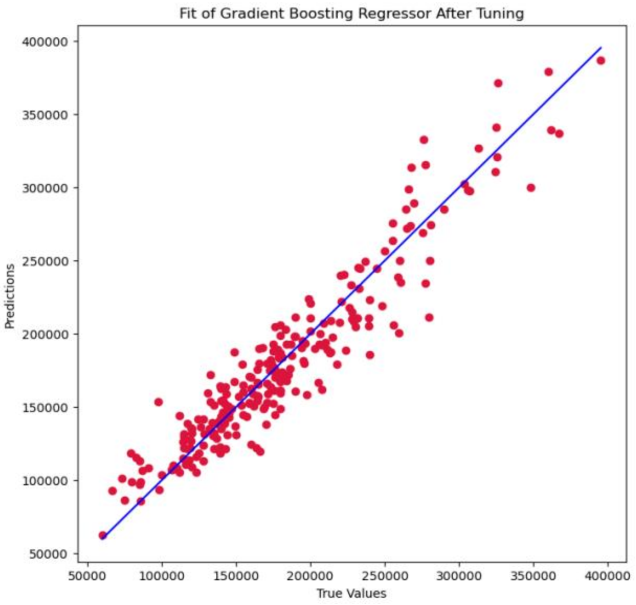
The evaluation metrics chosen are common metrics to evaluate regression models. RMSE is a measure of the difference in predicted values from the actual values. A lower RMSE indicated a model predicts values that fall closer to true values than a model with a higher RMSE. R-squared measures the amount of variation in a target variable that is accounted for by the feature variables. A value of 1.0 indicated features perfectly account for all variation within the target. An r-squared values closer to 1.0 indicates a better model than values closer to 0.0.

With these evaluation metrics in mind, it is evident that the gradient boosting regressor is the best model to predict sale price in this dataset. To ensure the model had the highest predictive power possible, I conducted hyperparameter tuning on the gradient boosting regressor using a grid search of possible hyperparameters. This process tests different combinations of hyperparameters to ensure the best hyperparameters are being used. The RMSE and r-squared values of this model were compared to the original gradient boosting model to ensure an improvement occurred. The tuned model had a reduction in RMSE and an r-squared values closer to 1.0. Both changes are improvements on the un-tuned model.

Model	**RMSE**	**R2**
Original Gradient Boosting Regressor	19975.70	0.8927
Tuned Gradient Boosting Regressor	19298.67	0.8999

For this final model, I also visualized the plot of predicted values and true values. In the below plot, the blue line indicates a perfect model where predictions fall exactly in line with true values. The plot on the left shows the untuned gradient boosting regressor, while the plot on the right shows the tuned gradient boosting regressor. This plot can visualize RMSE and help to understand if overfitting is occurring. The plot from the tuned model shows closer clustering of predicted values around true values throughout the plot. There is also an even spread of values above and below the blue line. This indicates there is likely not overfitting occurring.





Conclusion

Predicting sale prices of houses would be a powerful tool for the entirety of the real estate market. Ensuring buyers and sellers both get a fair price can help ensure faith in the real estate process. Confidence behind housing estimates can be improved using a model such as this one which could decrease the rate that sales fall through. For this dataset, hyperparameter tuned gradient boosting regression provide the most reliable results.

Assumptions

During the cleaning stage of this analysis, I encoded ordinal variables by hand. I knew these were ordinal by reading the documentation. However, in my analysis I could have mis-categorized variables as non-ordinal when they were truly ordinal. I did take care to avoid this, however this mistake would have consequences in the final model. Additionally, I filled missing vales in the numeric feature with the median of that column. In this, I made the assumption that the median was the most appropriate measure of each variable without examining each variable individually. The other alternatives were to use the mean or remove missing values. Again, this could influence the final model.

The chosen model, gradient boosting regression, is an ensemble algorithm in which each new model attempts to correct the errors of previous models. This model combines a range of learners like decision trees and linear models. This model is usually not very sensitive to outliers. Additionally, this model is very effective in complex datasets such as this one, where relationships between variables is difficult to capture with one model (GeeksforGeeks, 2023). This model does not make any major assumptions but is susceptible to weak learners. If all learners used in the model are too weak, it may not be the best model possible.

Limitations & Challenges

The dataset chosen is from a very specific time frame in a very specific area of Iowa. Learnings from this model may not be able to be widely applied to other geographic regions or time frames. Additionally, varying data cleaning method would alter the final dataset and may therefore benefit from application of a different regression model. I have tried to make decisions in cleaning and modeling that are both logical and statistically sound to reduce this risk.

Recommendations, Future Uses & Implementation

This model could be used for a larger scale implementation. However, before that can happen it must be tested in additional time frames and regions. Due to the scope of this data, it is likely the model will need to be modified before large scale implementation is possible. Additionally, due to the volatility of the housing market, I strongly recommend obtaining more recent data than was used in this exercise.

Ethical Considerations

I can think of few ethical considerations that would be detrimental to this analysis. As mentioned above, my main concern is application of this model to new data without fully evaluating the value and accuracy of the model before a large-scale launch. Failure to properly evaluate both the data, model and resulting predictions could result in financial losses to buyer, sellers, and real estate agents as well as a loss in faith in the real estate market.

10 Questions from Audience

1. Why does this research matter to us?

   This research matters because it shows there can be a more reliable and stable way of predicting housing prices than using appraisers.

2. Are there other regression models that could have been used?

   Yes, however I chose the models I did to cover many of the various types of regression models.

3. Why did you choose the evaluation metrics you did?

   RMSE allows us to understand the differences between actual and predicted values without having to go through each individual values. R-squared helps us understand if we are using the correct predictor variables. Plotting actual versus predicted values as I did helps us understand if overfitting is occurring.

4. Why was the linear regression model such a poor fit?

This model was likely such a poor fit because there is not a linear relationship between the features and target variable.

5. Could more outliers have been removed from sale price?

   Yes. I tried to remove all major outliers without limiting the dataset too much. Removing more outliers would likely have reduced the power of the model.

6. Does handling a combination of numerical and categorical features make this analysis more difficult?

   Yes it does. The feature must be handled very carefully to ensure their integrity is not degraded.

7. Would any other features have been useful in this analysis?

   The feature involved were exhaustive and seemed to cover all possible attributes of a house that could have changed the price.

8. In preparing the numerical variables, you applied a standard scaler, would a different scaler have changed the results?

   Possibly! The scaler I chose is used frequently in analysis such as this one though, so I feel confident in my decision.

9. You applied the scaler to only columns that were originally numeric, not the categorical columns that you encoded as numeric. How did you make this decision?

   The columns that were originally numeric had a much larger spread and variance than columns that were encoded as numeric. The scaler was only applied to them because they were the variables that truly needed scaling. Additionally, scaling the encoded categorical variables would have made it impossible to de-code them later on if it was needed.

10. What further steps would you take with this model now that it is complete?

    The original data source contains a test set; all the same features with the target variable removed. I'd love to see how well the model works on this dataset that it is naïve to.

References

Decisiontreeregressor. scikit. (n.d.-a). https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

GeeksforGeeks. (2023, March 31). *Gradient boosting in ML*. https://www.geeksforgeeks.org/ml-gradient-boosting/

House prices - advanced regression techniques. Kaggle. (n.d.). https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques

Linearregression. scikit. (n.d.-b). https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

Randomforestregressor. scikit. (n.d.-c). https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html