STAT1361 Final Project Technical Report

The housing market is a crucial component of the economy, affecting millions of people through the buying and selling of homes, the construction of new properties, and the financing of real estate transactions. Predicting housing prices is essential for buyers, sellers, lenders, and policymakers to make informed decisions in a rapidly changing market, and to avoid financial risks associated with housing investments. Designing a strong model to predict housing prices allows for the opportunity to gain insights into the underlying factors that drive housing prices and inform our understanding of the broader trends that shape the housing market.

For this project, an initial training dataset containing information about 700 homes previously sold in Pennsylvania was analyzed for the development of a housing price prediction model. A preliminary examination of the training set revealed that it consisted of four categorical predictors and twelve quantitative predictors. These predictors included features such as square footage, number of stories, and zip code, which are common variables considered when determining a home's value. The price column of the dataset was our desired response and was considered a quantitative variable in this analysis. A more detailed analysis of the dataset indicated that the training data was complete, with no missing values that needed to be removed or imputed. After gaining an understanding of the dataset's structure, I began a more in-depth analysis of the statistical relationships between the predictors and the price.

To determine the how each predictor related to the price (response), I first utilized simple scatterplots and boxplots to identify any trends. With the **ggpairs()** function from the GGally library, I created a pairs plot for the quantitative predictors. From this plot, 9 of the 12 predictors were identified to be significantly correlated with price. Around 5 of those 9 also had a moderate to strong linear correlation with price (r > 0.5) with the scatterplots visually supporting this linear correlation. Figure 1 shows each of the 9 significant predictors plotted against price. Predictors such as “sqft”, “bathrooms”, and “totalrooms” show a strong linear correlation. With

Chart, box and whisker chart

Description automatically generatedChart, scatter chart

Description automatically generated Figure 1 Figure 2

these linear correlations as well as small 95% CIs across all the predictors (Figure 1), it seemed to support a true linear relationship between the predictors and house price. Boxplots were used to relate the categorical variables in the training dataset against price. For all four categorical variables, the boxplots further supported the possibility of a linear relationship as the central tendencies among the subcategories were not significantly different. In both the scatterplot and boxplot graphs, there is support for the presence of outliers and leveraged points. I considered performing an analysis to identify the outliers or leveraged points and adjust the dataset accordingly, but it ultimately didn’t make sense to do so as these extreme data points provide useful information to the data (i.e., since a real house with these features and price was sold). Finally, I decided to further test for linearity by checking the normality and heteroscedasticity conditions for the predictors. My results were inconclusive as some predictors passed both conditions while others failed. In all, I decided to test both linear and non-linear models since there was evidence for both a true linear and non-linear relationship.

To perform my analysis, I first used one-hot encoding on the categorical variables to transform the categories into a numerical representation for models unable to work with categorical data. I also created a validation set with an 80/20 split of the training data. The first model I considered and trained used basic multiple linear regression that incorporated all the variables (i.e., no modification to the training dataset). I used this since it was very simple to implement, and because it provided a baseline to compare other models to. The next few models I wanted to train included another linear regression model adjusted for collinearity, the forward stepwise selection, and the lasso. After one-hot encoding, the training data jumped up to around 30 total predictors. It seemed highly unlikely that all these predictors would be significant in predicting price, so I sought models that could reduce the number of predictors and identify those that were beneficial. For my non-linear model, I decided that a Random Forest Regression model would be a good model to test given my significantly large training set and Random Forest’s ability to adapt well to noise in the data.

A summary of the initial regression model showed that five predictors ended up having high linear dependencies with other predictors, so they didn’t contribute to the model. These five predictors were removed from the training set. For the next model, VIF values were used to identify and solve any co-linearity problems in the training data. Predictors “totalrooms”, “fireplace”, and “exteriorBrick” were removed with a new feature “bed+bath” being added as the linear combination of the bedrooms and bathrooms in the house. This model ended up performing worse on the validation set than the basic model likely due to large increases in bias from removing collinear features. Forward stepwise selection was used with adjusted r squared as the metric to determine the best subset of predictors, but also performed worse than basic regression. Lasso with cross-validation performed slightly better in terms of MSE but was more importantly able to reduce the number of important predictors to 13. In all, the less flexible linear models were not able to adjust well to the training data likely due to underfitting against the non-linear relationships in the data. The non-linear Random Forest on the other hand proved to be able to fit better (Figure 3). The Random Forest specifically trained on the features selected by lasso performed the best. With these features, I performed a basic 10-fold cross validation to tune

Application

Description automatically generated with medium confidence

Figure 3

Chart, histogram

Description automatically generatedthe mtry hyper-parameter. The mtry value of 4 with 500 trees per forest performed the best most of the time. The final model was a Random Forest Regression model with an mtry of 4 and ntree of 500 trained on the whole original training set (adding validation back to the 80% training set). The resulting predictions on the test set produced a distribution very similar to the distribution of prices in the training set (Figure 4). The range of my predicted values, however, was much smaller with a maximum prediction of ~1.6 million versus the 3.3 million in the training dataset. However, given the extreme right-skewedness of the price distribution, the smaller range is not a large concern.

Figure 4

Based on my analysis, it can be concluded that a house price can be reasonably predicted using its corresponding features. In particular, the number of stories, the year it was built, and the square footage of the house, among other features, all significantly play a part in determining a house’s predicted price. The final Random Forest model I decided upon furthermore shows that the relationship between a house’s price and its various features has non-linear relationships that can’t be captured well by linear models. Ultimately, while my analysis produced a strong model for predicting housing prices, my methodology could be improved upon. To me, most critically is the issue with predictors having vastly different ranges. One option that could benefit a future analysis could be to standardize the predictors such that they fall between 0 and 1. The obvious issue is that this may cause some information loss in the conversion. For example, zip codes aren’t necessarily a continuous variable, so standardizing them might end up suggesting an ordinal relationship in which zip code 10101 is seen as less than 95050. Nevertheless, I believe performing more rigorous pre-processing of the predictors so that the ranges of numerical values aren’t so different could be beneficial when training a model. Additionally, I started out by training linear models, but came to the realization that the high bias of linear models was unable to fit well to the training data. I believe spending more time analyzing other non-linear models such as splines and further tuning my Random Forest hyperparameters could further improve the model to predict housing prices.