# Using Machine Learning to Determine New Construction Home Value

Nick Blanding nblanding@wisc.edu
Kate Chesney kchesney2@wisc.edu
Jonah Maroszek jmaroszek@wisc.edu
Nolan Peterson nconville@wisc.edu

## Abstract

*The aim of this project is to systematically estimate the value of a new construction home in the US. The data for this project comes from Census Bureau's Survey of Construction (SOC). The data set is comprised of approximately 40 features which mostly describe the physical characteristics of the home. Using linear regression as a baseline, the following models were trained on the data with the goal of maximizing prediction accuracy of the sale price of the home: Decision Tree (SciKit Learn), Decision Tree (GUIDE), Bagging Regression, Gradient Boosting, Stacking Regression, Random Forest (SciKit Learn), and Random Forest (GUIDE). To evaluate our models' accuracy, mean squared error (MSE) and coefficient of determination* $(R^2)$ *were used as metrics. After comparing each model, it was found that Random Forest (SciKit) was the best performing model, yielding a test accuracy of 0.79* $(R^2)$ *and a MSE of* $9.19 \times 10^{10}$.

## 1. Introduction

Housing is a primary need of every individual. Furthermore, it is an excellent investment vehicle that is readily attainable for many middle class Americans. Because housing is a such a core asset to so many Americans and to the economy in a broader sense, it is highly important that we be able to accurately determine the value of it. Newly constructed homes are an especially important sector of the housing market. There are two main groups that have a large stake in the value of homes in this sector.

The first of these two groups are home buyers looking to purchase a new house. For many, purchasing a new house can represent a financial benchmark. Along with this benchmark comes the hope that they are making a sound financial decision and investing in an appropriately valued asset. Knowing the value of their new asset will certainly bring these home buyers peace of mind.

The second group are the ones building these homes. Whether that be real estate investors or individuals simply looking to build a home; they all face essentially the same challenges. Construction is risky business and there is a lot that can go wrong. This group has to know if the risk is worth the reward. Engaging in a speculative investment without a clear understanding of the payout is simply bad business. However, if the value of the investment could be accurately predicted with a model beforehand, it removes some of the uncertainty in the investment.

This project aims to train and tune a machine learning model which can accurately predict the value (sale price) of new construction homes. Such a model will have the following qualities:

**1.** High Performance (Generalizes well)

**2.** Low Computational Cost

**3.** Be Reproducible

**4.** Have Easy Interpretation

These qualities can be thought of as goals for the final model of the project. In the results section will explain how the chosen model accomplishes these goals.

In addition to building a model that meets the above goals, it is also an aim of this project to determine features of interest. When buying or building a newly constructed home there are many choices to make. It can be difficult to know which ones are right or at least which choice are the most consequential. By identifying features that are highly important in the models, this analysis can shed light where the priorities of homeowners or home builders should be.

## 2. Related Work

Previous work that has focused on predicting housing prices have utilized both simple linear and multivariate regression models. For instance, one project predicted real estate values by using features such as square feet, price of the home, bedrooms, and bathrooms, in order to train their models [1]. The models included in this study were simple linear regression, multivariate regression models, and polynomial regression. The root mean square error was found for each model. This study concluded that a mix of models is the best approach, as simple linear models give a

large bias while more complex models give a high variance.

Additional studies utilized an ensemble methods approach, in order to obtain the best predictive performance. For instance, one study predicted house prices by using various regression models including kNN, Artificial Neural Networks, Bootstrap Aggregating, and Boosting [2]. Similarly, Baldominos and others used this type of methodology to identify opportunities in the housing market [3]. Regression trees, kNN, support vector machines and neural networks were tested. This study found that ensembles of regression trees obtained the best performance. These projects serve as a potential framework and example for our methodology, as various studies have concluded that an ensemble learning strategy is optimal.

Recently the company Zillow got into financial trouble when they used algorithms to price and buy homes [4]. Zillow started as a website that would value homes with a Zestimate, their signature valuation metric. The company then started transitioning into buying and selling homes, attempting to quickly do patch work on a home and then selling it for a profit. When Zillow started, they had an error rate of fourteen percent in accurately pricing homes, and as of recently they claimed to have lowered the error rate to two percent. They attempted to profit off of this "accurate" modeling and started buying up houses quickly, and often overprice. Then a problem came where they could not flip the homes fast enough, or for a profit. They paused home buying in October, laid off two-thousand staff, and are expected to have to sell the homes at a four and a half percent discount then to what they purchased for. Zillow's failures serve as a lesson into attributing statistical models to fast-changing markets.

## 3. Proposed Method

Our goal for this project is to utilize an ensemble methods approach, with multiple machine learning algorithms, in order to determine an optimal predictive model for values of new construction homes. Once data cleaning and feature selection were conducted, as described below, training and test data sets were created for the use of each model. The training set was made up of 80% of the data, while the test set contained 20% of the data. We explored our general data by gathering summary statistics, such as mean final sale price and standard deviation of final sale price, as well as creating a visualization of the final sale price distribution.

Once obtaining a general understanding, we utilized the following method: initially, a simple linear regression model was implemented as a baseline and assessed using the coefficient of determination on the test set. We then implemented and assessed various machine learning models including Random Forest, Decision Trees, Gradient Boosting, Stacking, and Bagging. All models used the same training set and were assessed and compared to one another using R-squared, or the coefficient of determination.

### 3.1. Random Forest

Random forests are one of the most popular ensemble learning methods for regression. It generally involves bagging with decision trees, yet instead of using the complete feature set, random feature subsets are utilized. In our case, 500 trees were fit on different bootstrap samples. The choice of 500 was determined through hyper-parameter optimization using a grid search layout. Once this regressor was implemented, R-squared was computed to assess performance.

### 3.2. Decision Trees

The next model incorporated into our project was a decision tree algorithm. This is a tree-like model of decisions where data is divided into subspaces or nodes representing instances with similar values. This model can be used for classification or regression. Hyper-parameter optimization was used to determine the best value for the following parameters for the sklearn decision tree: *criterion, splitter, max_depth, min_samples_split, and max_features.* A GUIDE tree was also created and rendered using Latex. This tree can be seen on page seven of the report. The criteria used to assess the trees are mean squared error and coefficient of determination.

### 3.3. Gradient Boosting

Gradient Boosting is a machine learning ensemble method using weak prediction models to form a stronger algorithm with a higher predictive performance. In general, this involves creating a base model, building another model based on the errors of this first model, and combining these again and again to form the final, stronger algorithm. In our case, hyper-parameter optimization was run with grid search to find the best values for the following parameters: *learning_rate, n_estimators, and max_depth*. The optimal parameters were used for our Regressor, which was fit and assessed using the coefficient of determination.

### 3.4. Stacking

We also utilized stacking within our project. Stacking is an ensemble machine learning algorithm that takes the predictions from other models, combines them, and makes a final prediction. In our stacking ensemble for regression, we combined the predictions from the three following algorithms: Random Forest, Decision Trees, and Gradient

Boosting. We fit our stacking regressor, and assessed it by finding the coefficient of determination and mean squared error.

### 3.5. Bagging

Finally, we decided to use bagging because it is a machine learning algorithm that can improve the stability and accuracy of less stable models. In general, this algorithm applies a bootstrapping procedure to an algorithm with high variance, normally a decision tree algorithm. In our case, we did use this regression model on our decision tree algorithm. Hyper-parameter optimization was run with grid search to find the best values for the following parameters: *base_estimator and n_estimators*. The optimal parameters were used for our bagging algorithm, which was fit and assessed by finding the coefficient of determination and mean squared error.

## 4. Experiments

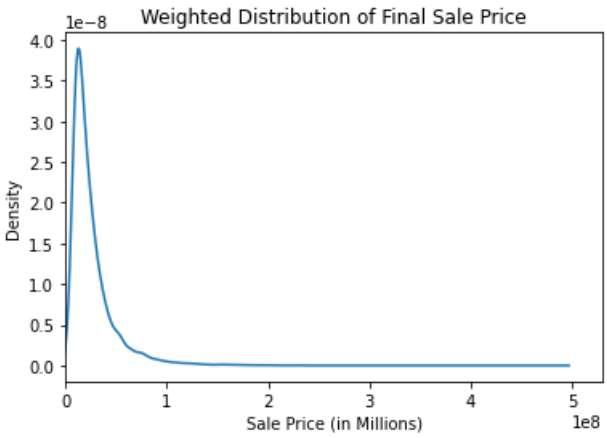
### 4.1. Dataset

The data used comes from Census Bureau's 2019 Survey of Construction (SOC). The data is publicly available (can be found here `https://www.census.gov/construction/chars/microdata.html`) and represents real observations of new construction homes. For this reason continuous features such as sale price or square footage of the home (and others) are top and bottom coded to protect the anonymity of the survey participants. Additionally the data has sample weights assigned to every observation to insure that the data is representative of the actual population of new construction homes. In total the raw data set has 24,810 rows and 61 columns.

The data required some reprocessing before it could be useful to the aims of this project. For instance, the response of our analysis, sale price, was not always reported. The training examples that did not have this available represent homes that were not sold usually. The examples were dropped from consideration. Some training examples reported a sale price and a final sale price. These training example represent homes which were sold on contract before they were built, but had to adjust the price of the home after construction was complete. For these examples only final sale price was considered. The square footage of the home was also reported in this manner. In the event that sale price was reported, but final sale price was not, the sale price was taken to be the final sale price. There were 78 cases where no square footage was reported. For these 78 cases mean imputation was used to fill the missing values. The features LOTV (lot value) and AREA (lot area) also had a considerable number of unreported training examples. These were again imputed using the mean of the examples that were reported.

In addition to the cleaning mentioned above, a number of features were dropped from the data set that were not useful to the analysis. They are summarized in the table 4.1. Over all, the cleaned data has 13,257 training examples, 40 features and 1 response variable. 34 of the features are categorical in nature, while the remaining 6 are continuous features. No feature scaling or transformations were applied to the data. The response variable has an approximately normal distribution with a heavy right skew. As in Figure 1.

Figure 1. Response Distribution




After the data was cleaned to a satisfactory level it was split into a training and test set. 80% of the data went to the training set while the remaining 20% went to the test set. After the training and test splits were made the respective sample weights were also separated into two distinct one dimensional arrays which were called 'training_weights' and 'testing_weights'. These arrays were used in the 'sample_weight' argument of the Scikit learn ML functions.

### 4.2. Software

The code for this analysis and model building in this project was primarily written in python 3.10.0. The editing was done in Jupyter Labs and VS Code IDEs. The Pandas and Numpy libraries were also used to clean the data in preparation for fitting the models. The Sci-Kit Learn framework was used for developing most of the machine learning models. The Matplotlib library of Python was used to create data visualizations. The library ggplot2 in R was used to create some additional Visualizations. GUIDE version 38.1 (Wei-Yin Loh, 2021) was used to generate additional machine learning models and visualizations.

| Feature | Reason Dropped |
|---|---|
| SLPR (Sale Price) | Merged with Final Sale Price |
| SQFS (Sq. Ft. of Home) | Merged with Final SQFS |
| CONPR (Contract Price) | Proxy for Sale Price |
| FCONPR (Final Contract Price) | Proxy for Sale Price |
| PVALU (Permit Value) | Proxy for Sale Price |
| SALE (Sale Date) | Low Predictive Power / Unimportant |
| COMP (Completion Date) | Low Predictive Power / Unimportant |
| FINC (Financing Type) | Low Predictive Power / Unimportant |
| AUTH (Permit Authorization Date) | Low Predictive Power / Unimportant |
| ID (Unique ID Code) | Low Predictive Power / Unimportant |
| STRT (Constr. Stat Date) | Low Predictive Power / Unimportant |
| $SLPR_F$ | Flag Variable |
| $FSLPR_F$ | Flag Variable |
| $FCONPR_F$ | Flag Variable |
| $LOTV_F$ | Flag Variable |
| $SQFS_F$ | Flag Variable |
| $FSQFS_F$ | Flag Variable |
| $PVALU_F$ | Flag Variable |
| $AREA_F$ | Flag Variable |
| $CONPR_F$ | Flag Variable |
| $FNSQ_F$ | Flag Variable |

Table 1. Reasons for dropping features from data set.

# 5. Results and Discussion

## 5.1. Overall Results

In this section, we discuss the results from our different models: ordinary linear regression, random forest, decision tree (SciKit Learn), decision tree(GUIDE), gradient boosting, stacking regression, and bagging regression. In order to evaluate our models, we used mean squared error (MSE) and the coefficient of determination $(R^2)$, which can be found in Table 2.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

Our simple linear regression model revealed a test accuracy of 47% $(R^2)$ and 2.28 $\times 10^{10}$ (MSE). We expected this performance score to be low, as this was our baseline algorithm for eventual model comparison. Despite this model being easily interpreted, its performance and accuracy are not strong enough, leading us to fit and assess the other models discussed.

We then fit a Decision Tree using both SciKit Learn and a GUIDE algorithm. When comparing the two, the Decision Tree from the SciKit learn package revealed a stronger model peformance. In fact, it had a test accuracy of 66% $(R^2)$ and MSE of 1.48 $\times 10^{10}$. This Decision Tree was tuned with a grid search to find the optimal parameters.

| Model | Performance (MSE) | Preformance ($R^2$) |
|---|---|---|
| Ordinary Linear Regression (SciKit Learn) | 2.28 $\times 10^{10}$ | 0.47 |
| Random Forest (SciKit Learn) | 9.19 $\times 10^{9}$ | 0.79 |
| Random Forest (GUIDE) | 1.60 $\times 10^{10}$ | 0.70 |
| Decision Tree (SciKit Learn) | 1.48 $\times 10^{10}$ | 0.66 |
| Decision Tree (GUIDE) | 1.12 $\times 10^{12}$ | 0.60 |
| Gradient Boosting (SciKit Learn) | 9.30 $\times 10^{9}$ | 0.78 |
| Stacking Regression (SciKit Learn) | 9.54 $\times 10^{9}$ | 0.78 |
| Bagging Regression (SciKit Learn) | 1.08 $\times 10^{10}$ | 0.75 |

Table 2. Model Performance on Test Set

It was revealed that criterion = friedman_mse, max_depth = 10, and min_samples_split = 10 resulted in the best model. This Decision Tree algorithm did give us a better performance score than a simple linear regression model and is easy to interpret, yet it has its disadvantages as well. This model can be easily susceptible to over-fitting and training it can be expensive. For these reasons, we decided to implement multiple ensemble methods in order to improve accuracy and performance. The GUIDE decision tree can be found on page 7.

We decided to implement four ensemble methods: Bagging, Gradient Boosting, Stacking, and Random Forest. The first method we tried was Bagging. After running hyper-parameter optimization, we found that base_estimator = tree, n_estimators = 500, and n_jobs = -1 gave the best performing model. It was found to have a test accuracy of 75% $(R^2)$ and a MSE of 1.08 $\times 10^{10}$. We did see great improvement in accuracy compared to linear regression and decision trees, as was expected due to its capability of improving unstable models such as a Decision Tree.

We then tried a Gradient Boosting algorithm. Hyper-parameter optimization was run once again and it was determined that n_estimators = 500 and max_depth = 5 gave the strongest model. This regression ensemble method was assessed and had a test accuracy of 78% and a MSE of 9.30 $\times 10^{9}$.

A Stacking algorithm was then implemented. This model used Random Forest, Decision Tree, and Gradient Boosting as predictors. The test accuracy revealed an R-squared value that was the same as the Gradient Boosting algorithm, 78%, yet a slightly different MSE of 9.54 $\times 10^{9}$. Due to Gradient Boosting and Stacking seeming to have a very similar predictive power, we finally decided to try a Random Forest model.

Our best performing model was the Random Forest from the sci-kit learn package. The $R^2$ was 0.79, and had an MSE of 9.19$\times 10^{9}$. The Random Forest was tuned with a grid search to search for the optimal amount of estimators. After the search, it was found that the 500 estimators would

| Feature | Importance Score (rounded to three digits) |
|---|---|
| FSQFS (Final Sq Footage) | 0.397 |
| DIV (Region of US) | 0.166 |
| LOTV (Lot Value) | 0.101 |
| AREA (Lot Area in Sq. Ft.) | 0.053 |
| FULB (Number of Full Bathrooms) | 0.354 |
| STOR (Number of Stories) | 0.021 |
| WAL1 (Exterior Wall Type) | 0.019 |
| FUEL (Primary Fuel Used for Heating) | 0.016 |
| DECK (Indicator for Outdoor Deck) | 0.016 |
| GAR (Type of Garage / Indicator) | 0.016 |

Table 3. Top Ten Important Variables

yield the best preforming model.

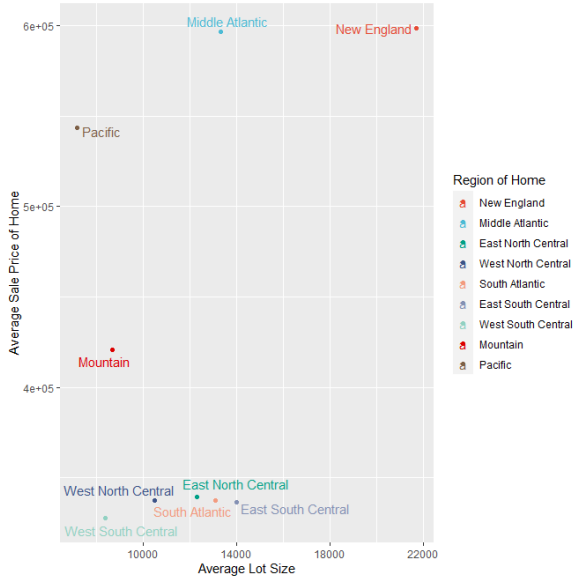
## 5.2. Features of Interest

As stated previously a priority of this report is to identify highly important features. These features relate directly to real world decisions that have to be made by individuals with a lot at stake. A useful attribute of the Sci-Kit Learn random forest regressor is that it scores features for their overall importance in the model. The top ten important features are summarized in table 5.2.

As expected the final square footage of the home is a highly important feature in determining the price of the home. It goes without saying the larger home are generally more expensive than smaller homes. The region (DIV) of the US in which the most is built is important, this is unsurprising and expected. Lot value also plays a critical role in the price of the home. Building a home on lots in higher value areas (for example urban vs. rural areas) can greatly affect the price of the home even when other factors are controlled for. Accordingly the area of the lot is also important. Although the relationship here is unclear. There is likely confounding variable at play here such the region the homes are in. This relationship can more clearly be seen in figure 2.

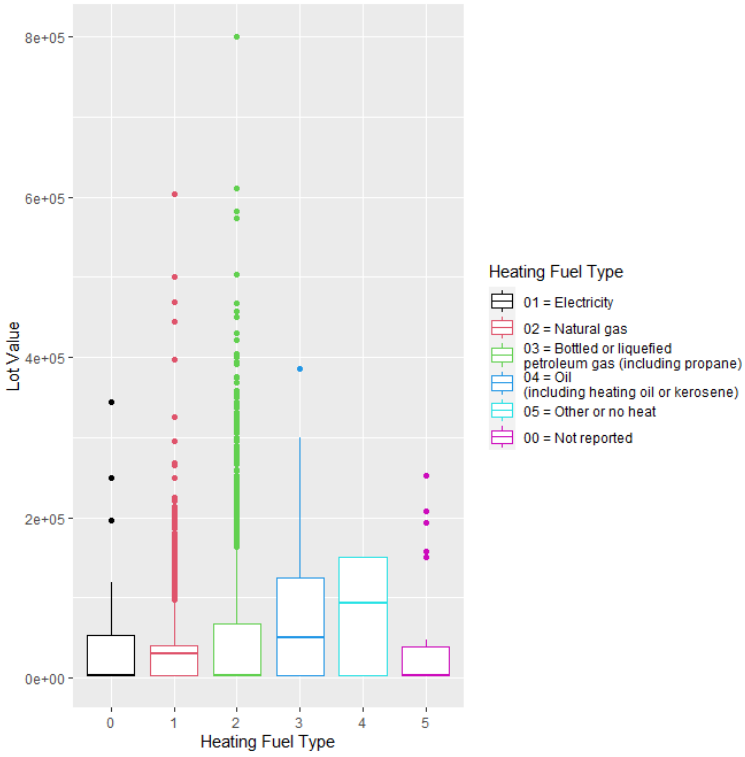
Interestingly the number of full bathrooms is quite important. It can be argued that the number of full bathrooms could be acting as a proxy for square footage of the home, however the same level of importance does not exist for the number bedrooms which would be expected to behave as similar proxy. The likely cause of the high importance of full bathrooms is the high marginal cost associated with fixtures and labor required to add a full bathroom. This high cost must be recuperated by the seller/builder so the sale price is increased significantly. The number of stories is important which is unsurprising because homes with additionally stories generally have more square footage and greater ”curb appeal”. Exterior wall type is important which is unsurprising because their can be a considerable difference in cost of wall materials as well as quality and longevity; all factors which must be represented in the final price of the home. The heating fuel used is important, the likely reason for this importance is the fuel is acting as a proxy for lot

Figure 2. Average Sale Price vs Average Lot size by Division




value. The location of the lot will play a large role in determining the value and the location will also determine the type of fuel available. For instance homes in rural areas often tend to use propane rather than natural gas, while homes in urban areas often have natural gas piped directly to them. This effect can be seen in figure 3.

Figure 3. Heating Fuel as Proxy for Lot Value



The presence or absence of an outdoor deck is important, the addition of a deck can add considerable value to a home. A garage is an important feature, it adds value (and cost) to a home. Attached garages increase the square footage of the house but at a lower cost than finished spaces.

## 6. Conclusions

In the introduction of this report four main goals were given for the chosen model. The Sci-Kit Learn Random forest model meets the four goals. It has a relatively high performance (using $R^2$ and MSE as metrics). It can be said that the model generalizes well if we take the test set performance to be representative of the generalization performance. It has a low computational cost. Fitting the model took less than three minutes and the time it takes to predict with the model is negligible. Thanks to the use to use Sci-Kit Learn frame work and well documented data processing steps the model should be easily reproducible for anyone with assess to the source code. However the one criterion in which the Random forest model is lacking in is ease of interpretation. Random forest models are notoriously a "black box". It can be difficult to make any kind of inferences about individual features from the random forest model alone. That said the feature importance scores can be pulled from the random forest model. These scores give a good indication for the features which should be explored in further analysis as was done in section 5.2. With this approach in mind, we say that the random forest model can be interpreted to a satisfactory level and therefore accomplishes the fourth goal.

Using machine learning to determine what factors contribute to a home's final sale price is important to both buyers, sellers, investors, and builders. Having informed market participants will help with more efficient price discovery, an important market factor.

In summary, we applied eight different machine learning models to survey data from the US Census's *Survey of Construction*. This data was engineered to properly fit models, applied to different model frameworks, and then analyzed to see which model best applied to our desired standards. Overall, the random forest was the best performing model because of the high performance, low computational cost, reproducible results, and clear interpretation.

## 7. Acknowledgements

The authors this report would like to acknowledge the following individuals:
Sabastian Raschaka, for his excellent instruction and code examples.
Wei-Yin Loh, for his GUIDE machine learning program.

## 8. Contributions

Nick Blanding: Contributed to creating tables, writing the report, feature visualization, and creating the presentation.

Kate Chesney: Contributed to data visualizations, writing the report (Related Work, Proposed Method, Results and Discussion), and creating the presentation.

Jonah Maroszek: Contributed to data cleaning, exploratory data analysis, model building, hyper-parameter optimization, writing/editing of the report, as well as plot and presentation creation.

Nolan Peterson: Contributed to data cleaning/feature selection, model building/tuning and creating visualizations. Contributed to writing report (Introduction, Data set, Features of Interest) and creating presentation.

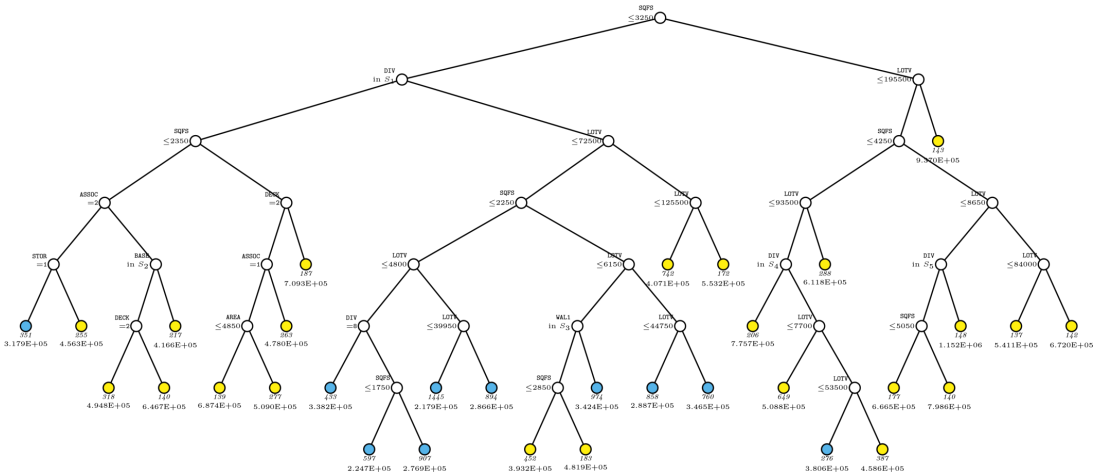
## References

[1] Manjula, R, et al. "Real Estate Value Prediction Using Multivariate Regression Models." IOP Conference Series: Materials Science and Engineering, vol. 263, 2017, p. 042098., https://doi.org/10.1088/1757-899x/263/4/042098.

[2] Oxenstierna, Johan. "Predicting House Prices Using Ensemble Learning with Cluster Aggregations." Department of Information Technology, December 2017.

[3] Baldominos, Alejandro, et al. "Identifying Real Estate Opportunities Using Machine Learning." Applied Sciences, vol. 8, November 2018.

[4] Clark, Patrick. "Zillow's Algorithm-Fueled Buying Spree Doomed Its Home-Flipping Experiment." Bloomberg.com, Bloomberg, 8 Nov. 2021.




GUIDE v.36.2 0.50-SE piecewise constant weighted least-squares regression tree for predicting FSLPR. Tree constructed with 13257 observations. Maximum number of split levels is 23 and minimum node sample size is 132. At each split, an observation goes to the left branch if and only if the condition is satisfied. Set $S_1$ = {1, 2, 9}. Set $S_2$ = {0, 3}. Set $S_3$ = {0, 4, 7, 8}. Set $S_4$ = {1, 2, 9}. Set $S_5$ = {3, 5, 7, 8}. Sample size (*in italics*) and mean of FSLPR printed below nodes. Terminal nodes with means above and below value of 3.842E+05 at root node are colored yellow and skyblue, respectively. Second best split variable at root node is FULB.