# House Prices Prediction

Johnny Nguyen, An Le

Depauw University

**Abstract**

This paper outlines the fundamental steps and approaches that are used to forecast the price of residential housing in the United States. By removing/filling in gaps in the data, engineering characteristics, and detecting outliers, the data was prepared for further analysis. In the United States, machine learning models have been used to provide advice to appraisers, sellers, and buyers of residential real estate. The goal of this study was to evaluate various preprocessing techniques, algorithms, and parameterization methods in order to determine the most accurate home price estimate.

## I Data Description

The Kaggle data I utilized in this research included a variety of situations as well as home selling prices in Ames, Iowa. Each home featured 81 different features, including 43 different categories and 38 different numerical values. There were 1460 components in the training set, but only 1459 in the testing set. In this scenario, I want to utilize both category and numerical data to explain and

anticipate the sales prices of unknown dwellings based on the features provided. Knowing the data's specifics was beneficial, but a deeper awareness of the housing domains led to a better comprehension of the dataset. Using the.corr() function from the pandas library, Table 1 displays the top 10 numerical qualities that most closely correlate our selling price value. After gaining a thorough grasp of the data set, data cleaning was the next stage in preparing the data for algorithm fitting.

| Name | Description | Correlation |
|---|---|---|
| OverallQual | Rates the overall material and finish of the house | 0.790982 |
| GrLivArea | Above grade (ground) living area square feet | 0.708624 |
| GarageCars | Size of garage in car capacity | 0.640409 |
| GarageArea | Size of garage in square feet | 0.623431 |
| TotalBsmtSF | Total square feet of basement area | 0.613581 |
| 1stFlrSF | First Floor square feet | 0.605852 |
| FullBath | Full bathrooms above grade | 0.560664 |
| TotRmsAbvGrd | Total rooms above grade (excluding bathrooms) | 0.533723 |
| YearBuilt | Original construction date | 0.522897 |
| YearRemodAdd | Remodel date | 0.507101 |

Table 1: Top 10 Attributes Relating to Sale Prices

# II Pre-Processing

Both the testing and training sets had missing values. To begin, I deleted attributes with 95 percent of their data missing by using the thresh option from

pandas' drop() function. Additionally, we see that some factors (e.g., warmth or closeness to other situations) have no effect on the models. As a result, we decide to eliminate these superfluous variables. All of these processes simplified our analysis when it came to filling in the missing information. We will simply eliminate those characteristics from the dataset that have an excessive number of missing values. We evaluated the missing values in this section since they do not exist. As a result, we simply delete such traits since they have no influence on the forecast. We build a function called dropColumn that accepts the data frame as an argument. The purpose of this function is to remove any columns that are deemed unnecessary or if 95 percent of rows are missing. We want to delete this feature as a constant feature in order to display the same value for all data set observations. These characteristics have a little effect on the prediction process. They do not offer us with adequate information to discriminate or forecast a target using our algorithm model.

After experimenting with a variety of preprocessing approaches, I opted to solve the missing numerical and categorical data using a variety of different methods and techniques. For categorical variables, the mean may be used to fill in the gaps, and for numerical variables, the mode value can be used to fill in the gaps. This enables us to make use of the information included in an incomplete dataset. If you have a randomly picked observation from a normal distribution, the mean and mode are plausible estimates of its value. We may get a better anticipated model by removing the gathered values from the data set rather than just eliminating them.

We developed a function called missingValuesInfo, which takes a data frame as an input and returns a data frame containing a percentage of missing values in each column in the data frame. Then, we utilize a method called HandleMissingValues

to fill in the missing values with the new values that were created. This procedure is based on the description of the data set as well as our general understanding of the impacts of house prices. Consider a home with no garage: we can simply enter 0 in the missing value field since we believe that the absence of a garage indicates that the family does not have any automobiles.

Occasionally, in the most extreme instances, where we have no specific understanding about what value should be substituted for the missing values, we just write None in their stead. At the conclusion of this procedure, we will run the missingValuesInfo one more time to see if there are any leftover missing values that we haven't appropriately handled.

The numerical variables are the only variables that are accepted by the machine learning model. As a result, it is important to convert the category values to numerical values before continuing. All that the machine learning model will comprehend is the numerical value and the information that may be extracted from it. The purpose of this conversion variable is to improve the overall quality of the model. We convert the category variables into numerical variables by using the createDummy function, which accepts a dataframe as an input and returns a new dataframe. First and foremost, we will create a data frame that has categorical values. Using this information, we can generate a dummies dataset using the get dummies method in pandas. Following the conversion of the dummy features from categorical variables, we will connect the original dataset with the dummy dataset and remove the category variables columns from the resulting join table.

Non-numeric values may be converted into strings using a function called nonNumeric to String, which is included in the library. We build a function called ordinal converting to convert categorical data into ordinal values in order to pro-

vide a more accurate forecast. We transform the values ”Ex,” ”Gd,” ”TA,” ”Fa,” and ”Po” into the numbers 4, 3, 2, 1, 0 in a sequential manner.

# III Feature Engineering:

First, we figure out how strong the linear relationship is between every feature in the dataset and ‘SalePrice’, which is the target one, by calculating the correlation values between them and plotting the results on the n-dimensional arrays. Our results show a positive trend among values, ranging from 0 to 1. A high correlation score would indicate a strong linear relationship between the movement of the two features, and vice sera. In our case, we select features that have their correlation value exceeding or equal to 0.5 as predictor features, encompassing, while also dropping several helpless features that almost represent a non-linear relationship with the target ‘SalePrice’, such as ‘MoSold’ and ‘YrSold’.

Among the current predictor features that we obtain in the previous work, we realize some of them have meaningful connections with other features in the dataset. Therefore, we create some additional features that we believe could aid in predicting our target ‘SalePrice’.

After calculating the correlation values between each feature in the dataset and ’SalePrice’ (which is the target feature), we plot the results of our analysis on n-dimensional arrays to determine how strong the linear relationship is between each feature in the dataset and ’SalePrice’ (which is the target feature). Our findings reveal a favorable trend over a range of values ranging from 0 to 1. A high correlation value would imply that there is a strong linear link between the movement of the two characteristics, and the opposite would be true. In our case, we

select predictor features that have a correlation value greater than or equal to 0.5 as predictor features, encompassing, while also dropping several helpless features that almost represent a non-linear relationship with the target 'SalePrice,' such as 'MoSold' and 'YrSold', as well as several helpless features that almost represent a non-linear relationship with the target 'SalePrice,' such as 'MoS

We have discovered that several of the current predictor characteristics that we obtained in the prior study have important links with other features in the dataset. For this reason, we develop certain additional elements that we feel will assist in the prediction of our objective 'SalePrice' in the future.

To be more specific, we include the term 'TotalLot,' which sums both the LotFrontage and the LotArea. We just add up the entire lot area of the home and call it a day. totalSF = totalBsmtSF + totalBsmtSF + totalBsmtSF = totalSF According to our intuition and our study into square feet in the home, the overall size of a house in square feet is the sum of the basement area, first-floor area, second-floor area, and third-floor area... In this equation, I believe the solution is rather easy. TotalBath is equal to the total of FullBath and HalfBath on a given day. The total number of bathrooms is the sum of the total number of full bathrooms and half bathrooms. TotalPorch is the total of the OpenPorchSF, EnclosedPorch, and ScreenPorch variables in the previous paragraph. The total porch of the home is calculated by adding up the porches of each individual dwelling in the house. It is calculated as the sum of the BsmtFinSF1 and the BsmtFinSF2 variables. Total square feet of the basement on first and second floors is expressed as a number of square feet. Basement space is measured in square feet. The total of all of the house's characteristics, such as the kitchen, basement, basement attributes, and so on, is called qualitative features. In addition to the predictors variable, which is

constructed from the additional variables listed above.

Following the salePrice rows, as previously said, we will only choose the characteristics that have correlation scores larger than 0.5 based on the correlation scores. If there is more than one characteristic in the predictor that is multicollinear (that is, they are significantly connected to one another), we will choose the feature with the highest correlation score since it is more predictive. Overall Quality, GrLivArea, GarageArea, YearBuilt, and TotRmsAbvGrd were the criteria we used.

# IV Algorithm

.

For the purpose of forecasting home prices, many models such as linear regression, linear ridge, lasso, and gradient boosting were tested in order to identify which model would provide the best optimum outcome. Except for gradient boosting, all of the models belonged to linear regression models, with the exception of gradient boosting. Despite the fact that linear regression was straightforward, it served as a guiding light throughout the endeavor. It aids me in the optimization and fine-tuning of the linear ridge and lasso regression models, respectively. The preprocessing models that were utilized were primarily concerned with producing all numeric values in order for the linear models to be fitted. Linear ridge is a sort of linear regression that may be used on data sets that have sparse information because of its name. With the assumption that the data set was quite limited, ridge began working on hyperparameter optimization. Ridge regression incorporates a penalty term x that drives the development of a more straightforward model. The findings were just moderately better than those obtained using linear regression.

Cost Function for Ridge Regression:

$$\sum_{i=1}^{M}(y_i - y_i)^2 = \sum_{i=1}^{M}(y_i - \sum_{j=0}^{P}(w_j * x_{ij}))^2 + \lambda \sum_{j=0}^{P} w_j^2)$$

Cost function for Lasso Regression:

$$\sum_{i=1}^{M}(y_i - y_i)^2 = \sum_{i=1}^{M}(y_i - \sum_{j=0}^{P}(w_j * x_{ij}))^2 + \lambda \sum_{j=0}^{P} |w_j|)$$

Regularization is implemented in both the ridge and the lasso algorithms, which is a technique for avoiding specific coefficients from weighting more than others and therefore preventing overfitting. Lasso reduces the number of characteristics to a bare minimum in order to prevent complexity. The fact that the output of the lasso does not vary much throughout the fine-tuning process suggests to us that we have effectively identified the proper and unimportant components in our data.

Finally, gradient boosting was utilized to stray away from the traditional linear regression model. The gradient boosting regressor is an ensemble approach that generates many groups of models in order to converge to a single best-fitting model, as shown in Figure 1. Nevertheless, as a result of this greater power, there is an increased risk of overfitting.

The learning rate was critical in the development of a model that did not overfit the available data. It required a lot of trial and error to try to come up with a universal tuning strategy for hyperparameterizing the algorithms. Eventually, utilizing ranges of values and mapping each value to a new algorithm resulted in a simple method of evaluating tuning settings for each algorithm. The alpha value was used to alter the linear ridge and lasso shapes. It was decided to utilize 1000 alpha values, ranging from 0.01 to 1, with each step being 0.01 to plot different levels of ”accuracy.” The gradient boost regressor was tweaked using the identical approach as the gradient boost regressor, with the exception that the learning rate

was changed. Eventually, the most optimum values were discovered.

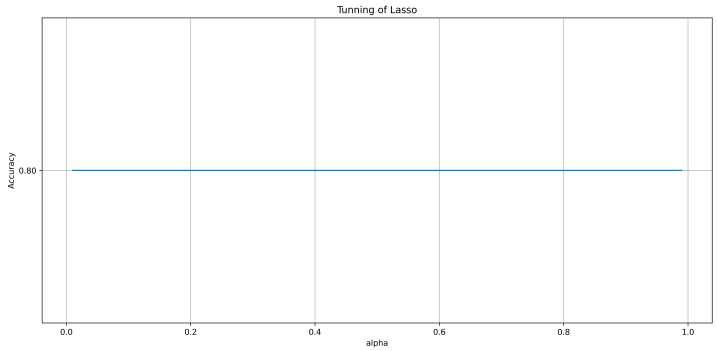


Figure 1: Relative Accuracies Corresponding to Alpha

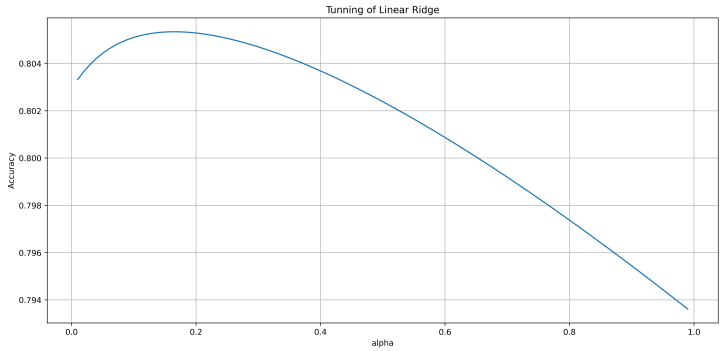


Figure 2: Relative Accuracies Corresponding to Alpha

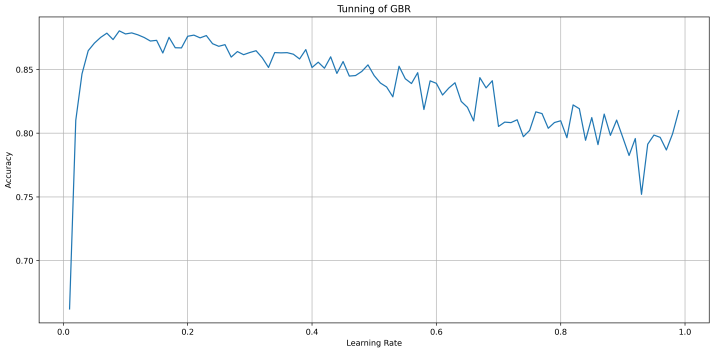


Figure 3: Relative Accuracies Corresponding to Learning Rate

# V Results

Predictions began to become more accurate once the models were fine-tuned to a higher degree. Because of the difference in scoring computations between the average variance and the mean squared error, the lasso outperformed the ridge, the lasso, and the linear regressionn. Gradient Boost outperformed the other two models in the kaggle contests, with the best results coming from the first model.

As a consequence of the findings, gradient boosting was shown to have a comparative advantage over the other models. Increased model complexity is a strong method that makes use of several models in order to improve upon the faults produced by the previous models. This is unquestionably a benefit over linear regression, despite the fact that it requires more processing capacity to be effective. Second, hyper tuning was critical in achieving precise results in the game. The graphs above demonstrate the significant reductions in accuracy that may result from either overstepping or understrapping the hyperparameter.

| Model | Tuning Parameter | Accuracy (r2) |
|---|---|---|
| Linear Ridge | Alpha: 0.16 | 0.803 |
| Lasso | Alpha: 0.99 | 0.805 |
| Gradient Boosting | Learning Rate: 0.09 | 0.88 |

Table 2: Comparation of CV scores between different models

Concluding from the results, gradient boosting had comparative edge on the other models. Boosting is a powerful technique that uses multiple models to improve upon the errors made by previous models. This is clearly an advantage over linear regression; Although, requiring more processing power to run. Secondly, hyper tuning was crucial in creating accurate scores. The graphs above illustrate the high drop offs in accuracy due to over or under stepping the hyper parameter.

# VI Conclusion

.

Preprocessing seems to be the most significant step in producing accurate models. Missing values and the predictors that were employed seem to be the most important elements influencing the models. The algorithms have enough information to provide semi-accurate findings when given the following predictors linked to the location, size, age, and quality of the home. Values that were substantially linked with price played a vital role in the efficient preparation of data. Linear regression performed well with numeric data; nevertheless, they did not have enough information on their own without the use of categorical inputs in the final analysis. The

model was able to provide the highest level of accuracy by using an ensemble of gradient boosting regressors. To summarize, understanding the domain of the data is the most significant factor in intuiting a visually appealing model.

# References

[1] Swalin, Alvira, ”How to Hanndle Missing Data”, Medium. Towards Data Science, 19 Mar. 2018, https://towardsdatascience.com/how-to-handlemissing-data-8646b18db0d4

[2] Hale, Jeff, ”Smarter Ways to Encode Categorial Data for Machine Learning”, Medium. Towards Data Science, 16 July. 2019, https://towardsdatascience.com/smarter-ways-to-encode-categoricaldata-for-machine-learning-part-1-of-3-6dca2f71b159

[3] Jain, Shubham, ”Linear, Ridge and Lasso Regression Comprehensive Guide for Beginners.”, Analytics Vidhya. 17 Sept. 2019, www.analyticsvidhya.com/blog/2017/06/a-comprehensive-guide-forlinear-ridge-and-lasso-regression/.