Choi Contaldi Final Project

Yena Choi and Sabrina Contaldi

December 2022

1 Abstract

The purpose of this project was to create a model that predicts the sale price of houses based on data recorded about 78 different attributes. The model compares the attributes of the house to those of houses that have been previously sold to estimate the sale price. The data was pre-processed, some attributes were eliminated and a variety of models were tested to find the most accurate algorithm to predict the sale price of houses.

2 Data Description

A total of 78 attributes were included in the data set. These ranged from square footage of the yard to condition of the basement. The most important predictors were overall material and finish quality(OverallQual) and above grade (ground) living area square feet(GrLivArea). These were found in Experiment B where we ran a correlation analysis between the attributes and the Sale Price. Other attributes that were significantly important included the following: the year the house was built (YearBuilt), the total square footage of the basement (TotalBsmtSF) and the number of cars that fit in the garage (GarageCars). There were some data that worked like a twin (we will refer to it as twin data), like GarageCars or GarageArea. Some of them resulted in a slightly better outcome when using only one of them, so we decided to use only one attribute from twin data that gives a better accuracy.

3 Experiment

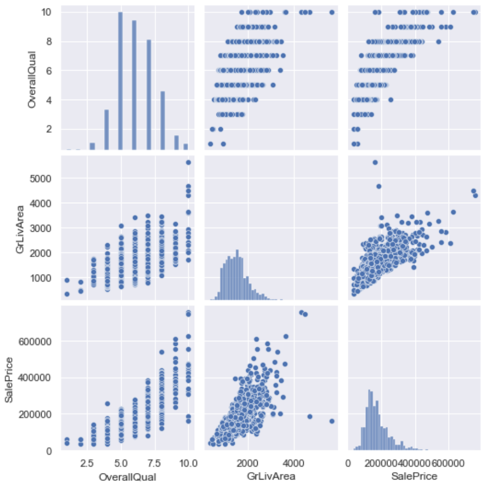
We pre-processed all of the attributes and isolated the most important ones by applying a correlation analysis of the attributes with the target attribute. We then ran the following 6 models on the data:

- Linear Regression
- Lasso
- Elastic-Net
- Bayesian Ridge Regression
- K Nearest Neighbor Regression
- Gradient Boosting Regression

3.1 Pre-Processing

For the first step, we decided to pre-process all of the attributes to make sure that we were dealing with only numeric attributes. For the NaN values, we replaced them with 0 or the mode of the training set. We then ran a correlation test between all of the attributes and the *SalePrice* attribute to help isolate the more important attributes. All attributes with a correlation coefficient lower than 0.3 or -0.3 were eliminated.

As we initially pre-processed the attributes, we made a note of whether the addition of the specified attribute increased or decreased the accuracy of the model. We then went through our notes and if an attribute had a correlation coefficient less than 0.3 but increased the accuracy of the model when it was added, we included it in our list of important attributes.



The correlation between the SalePrice and the 2 attributes with a correlation coefficient greater than 0.7

Additionally, we looked at the Quality and Condition attributes and ensured that they used the same scale. Originally, these attributes were assigned numbers at random to represent Excellent, Good, Average/Typical, Fair and Poor. We remedied this by letting 0 represent the best score and the highest number represent the worst score. We then flipped the scale and let 0 represent the worst score and let the highest number represent the best score. We found that the second scale had the greatest positive effect on the accuracy.

3.2 Algorithms and Hyperparameterization

In our experiment, we tested the following models.

3.2.1 Linear Regression

Linear Regression is one of the statistical ways to make relationships between an attribute set of independent variables and a dependent variable. Linear regression algorithms make use of gradient descent to optimize their predictions. The following function is a cost function used within gradient descent that predicts the error of the model:

$$j(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_{\theta_1}x^{(i)} - y^{(i)})^2$$

As for hyper-parameters, we tried setting the Normalize parameter to True but this had no affect on the mean accuracy of the model; it did not go higher than 80.877

3.2.2 Lasso

Lasso regression is a regularization technique. It is similar to Linear Regression but it can reduce features used in the model.

3.2.3 Bayesian Ridge Regression

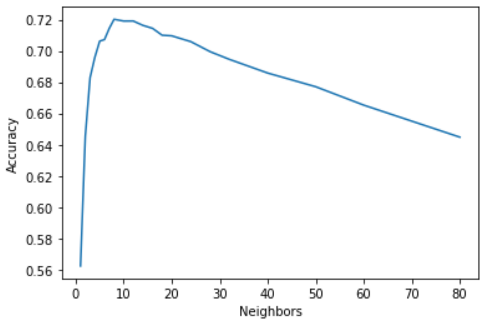
'BayesianRidge estimates a probabilistic model of the regression problem'. By performing linear regression using probability distributors instead of point estimates, the model is able to navigate poorly distributed data (*Scikit learn*).

3.2.4 Elastic-Net

Elastic-net is a linear regression model that is particularly useful when there are multiple features correlated with one another.

3.2.5 K-Nearest Neighbor Regressor

The k-nearest neighbor regressor is a model that makes predictions on the data by looking at a specified number of the nearest data points. In order to isolate which was the best value for K we implemented a parameter search plot to test which values of k had the highest accuracy. This is the only hyper-parameter that was implemented for this model: n_neighbors = 7.



The accuracy for different values of k

3.2.6 Gradient Boosting Regression

Gradient Boosting Regression is used to produce a predictive model from an ensemble of weak predictive models (Prettenhofer). We will plot deviance against boosting iterations. The hyper-parameters used for this model were the following:

- learning rate: 0.04
- max depth: 4
- number of estimators: 170
- subsample: 0.89

We implemented a method that, using a random number generator, found the hyper-parameters that allowed the model to be the most accurate. Since this model was the most accurate without hyper-parameters, we decided to focus on the hyper-parameters for this specific model.

3.3 Results

The Gradient Boosting Regressor was the most accurate model by a significant amount. The Average CV scores can be seen below for all the models:

Models Compared	
Model	**Accuracy**
KNN	0.714509562097632
Elastic Net	0.796829568732115
Linear Regression	0.808772591834881
Lasso	0.808783097304553
Bayesian Ridge	0.809102952050259
Gradient Boosting	0.884499335895707

3.4 Analysis

We hypothesized that eliminating some of the attributes would improve the accuracy of the model, however we thought that the accuracy would improve more significantly. There were also attributes that we intuitively thought would be more important that actually made little to no difference. Considering that our pre-processing approach was relatively basic, this could explain why these attributes, such as neighborhood, were not as significant as expected.

We also thought that the K-Nearest Neighbor Regressor would be more accurate than it was, but even after finding the most accurate value for k, it was approximately 8% less accurate than our next least accurate model and more than 15% less accurate than our most accurate model.

The Gradient Boosting Regressor was the most accurate model by approximately 8%. The focus that we put on the hyper-parameters definitely helped make this model more accurate but a large part of its accuracy has to do with the way the model works. The iterative nature of the model helps reduce the error.

4 Conclusion

In this study, we mainly focused on the Gradient Boosting Regressor model to produce a strong and accurate model among weak attributes, using ensemble method.

Compared to other models like Linear Regression, Lasso, Elastic Net, Bayesian Ridge Regression, or K Nearest Neighbor Regression, Gradient Boosting Regression gave a significantly higher accuracy, and in our opinion, this is because we did not focus on tuning or pre-processing the data. Gradient Boosting Regression is the one that deals the best with weak attributes and give the best output using the technique ensemble.

Therefore, the most accurate model that we ran was the Gradient Boosting Regressor, which was approximately 8% more accurate than our next best model, resulting in the accuracy of high 88

We focused on gradient boosting model this time, but we can tune the data using when we are using Linear Regression model and have a better result with the model in the future.

5 References

Prettenhofer, P., Telenczuk, M., amp; Ni, K. (n.d.). Gradient boosting regression. scikit. Retrieved December 15, 2022, from https://scikit-learn.org/stable/auto$_{e}xamples/ensemble/plot \quad -_{g}$ $radient_{b}oosting_{r}egression.html$

Scikit learn - Bayesian Ridge regression. Tutorials Point. (n.d.). Retrieved December 15, 2022, from https://www.tutorialspoint.com/scikit$_{l}earn/scikit-_{l}$ $earn_{b}ayesian_{r}idge_{r}egression.htm$