House Price Prediction Using Regression Techniques

Final Course Project

DSO 530

Author: Shamim Samadi

# Introduction

Goal of this project is to predict the final sale price of houses for residential homes in Ames, Iowa. Given a dataset of houses including a variety of features, the goal is to build a model to predict the final sale price of a house. I picked this project as my course project mainly for the following reasons:

1. The project is trying to solve a problem that I consider interesting (and challenging) from a business perspective.
2. The dataset is a particularly challenging dataset. I am going to elaborate more on the main challenges I was faced with during the project.

The project is a competition on Kaggle (<https://www.kaggle.com/c/house-prices-advanced-regression-techniques>). My R markdown script for the work I have done for the project can be accessed on my Kaggle profile: <https://www.kaggle.com/shamoool/house-price-prediction>. Please note that, due to some constraints with respect to the packages I have been able to use on Kaggle, the Kaggle version is slightly different from the version that I have submitted as the course project. My plan is to further push the performance of my predictor even beyond the scope of the project and keep improving my ranking on the competition leaderboard.

# Business Perspective

The house sale price prediction is a very interesting problem from a business perspective. Being able to make an accurate prediction on the final sale price can help better evaluate the worth of the property and thus setting the price (e.g. by the agencies or sellers) such that the purchase package is going to attract more potential buyers and thus increasing the odds of a sale actually happening, which means increasing the ultimate profit made by the agency.

House price prediction is not a trivial problem to solve. There are a lot of different parameters that play a role in determining the value of a house. As borrowed from Kaggle:

“Ask a home buyer to describe their dream house, and they probably won’t begin with the height of the basement ceiling or the proximity to an east-west railroad. But this playground competition’s dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence.”

# Dataset

The dataset used for the purpose of this project is the Ames Housing Dataset which includes 2930 observations representing residential homes in Ames, Iowa where there are a total of 79 explanatory variables describing each observation. The target variable is the variable SalePrice, the final sale price of each house.

The predictor variables describe the various physical attributes of the property in both a qualitative and quantitative sense. Most of the variables capture information that a buyer would consider in practice when considering purchase of a residential property (e.g. when was it built? How big is the lot? Is the basement finished? How many bathrooms does it have?, …).

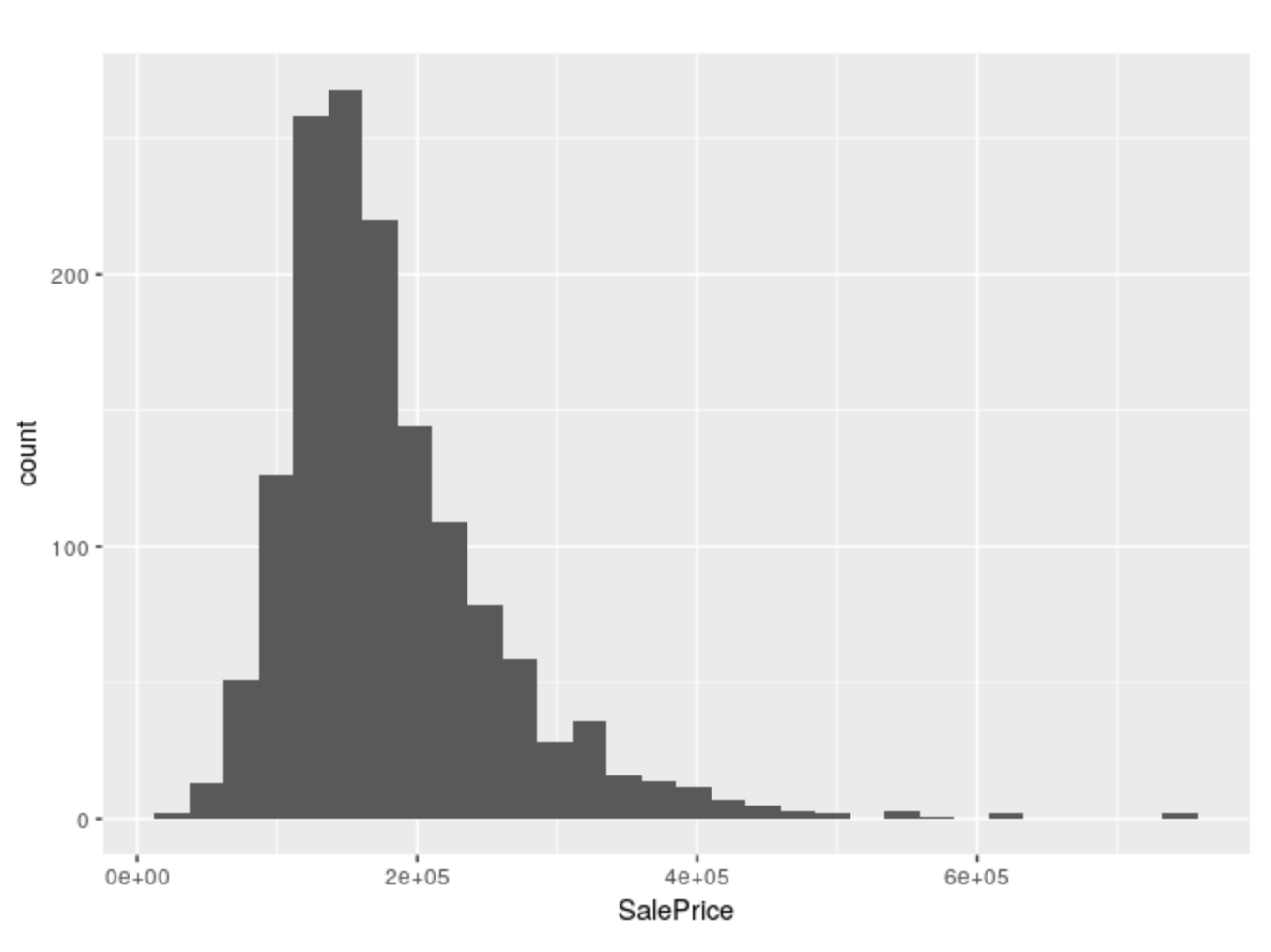
* **Continuous Variables:** The 20 continuous variables generally hold information about the area dimensions for different parts of the house for each observation. Examples of typical attributes are lot size and total dwelling square footage. More specific measurements such as the area measurements for the basement, main living area and even porches is provided for different categories of each living space.
* **Discrete Variables:** The 14 discrete variables typically quantify frequency of occurrence of items within the house. E.g. how many kitchens, bedrooms, and bathrooms on each floor level (basement level, ground level, etc). Other examples are garage capacity renovation/remodeling/construction dates.
* **Categorical Variables:** There are a total of 23 nominal and 23 ordinal variables on this dataset. While some of these variables are binary (i.e. they represent two categories), there are variables that represent as many as 28 classes. As an example, variable STREET has only two possible classes: gravel or paved; versus variable NEIGHBORHOOD captruring neighborhood areas within the Ames city limits where a property is located, which has 28 categories. As a general rule, the nominal variables describe a condition or a category for a specific house attribute while the ordinal variables rate such attributes.

# Data Pre-processing

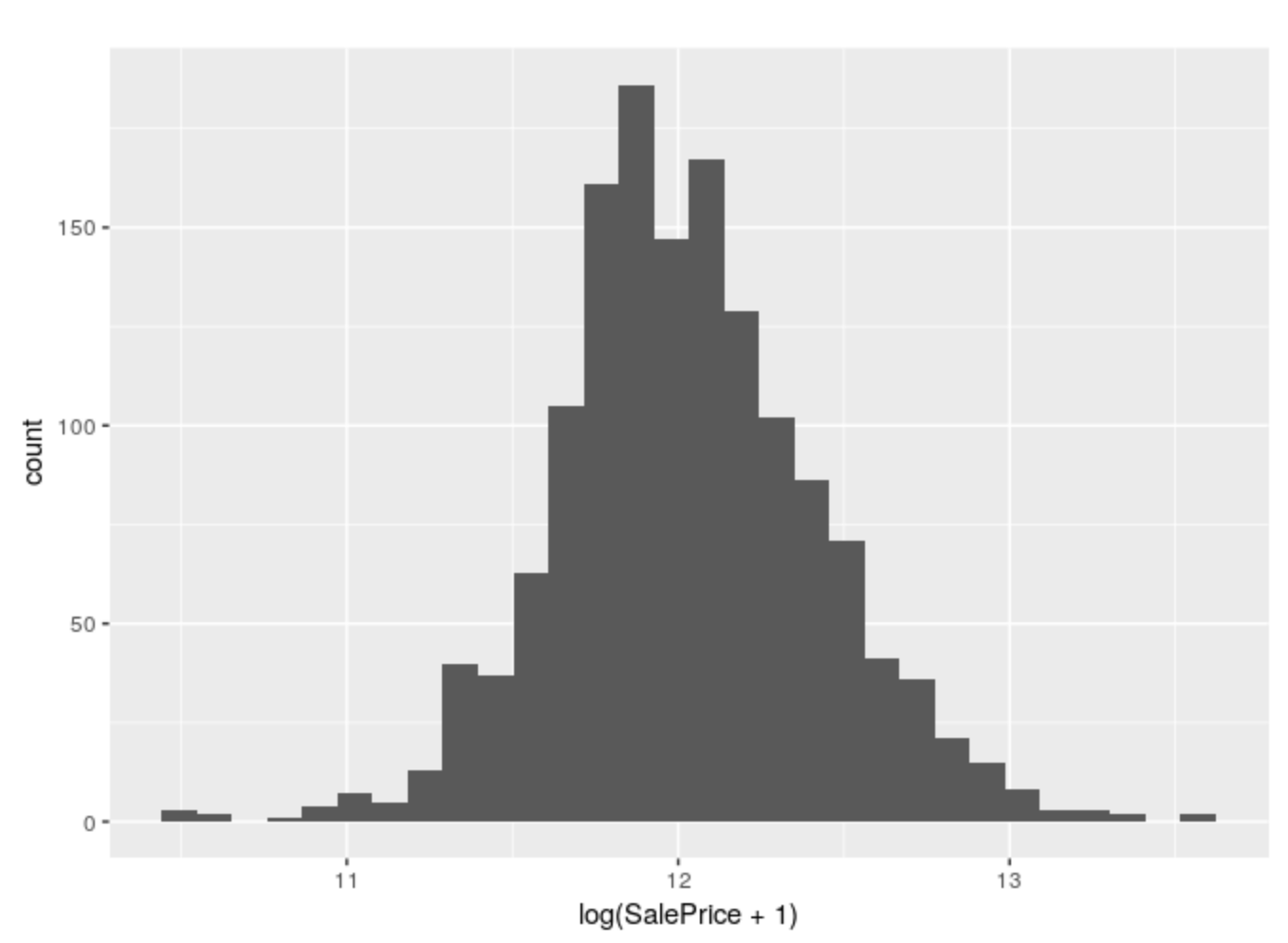
1. **Data Transformation**

Since the target variable represents financial data (i.e. price), it helps the regression model to transform the dependent variable using natural logarithm transformation. Taking a look at the histogram of the SalePrice variable reveals that the normality assumption is violated before the transformation and that the histogram looks much better after replacing the variable with its log.

* Histogram of the original dependent variable (skewed distribution)



* Histogram of the transformed dependent variable



1. **Handling Missing Values:**

There were several columns in the data that contained missing (NA) values. As part of the preprocessing step, I handled these missing entries using imputation technique where I used the most frequent method for imputation, i.e. for each variable containing missing values, the missing entries were replaced by the most frequent entry in the corresponding column. To avoid leakage of information from the test set, the most frequent entry was determined according to data in the training set only, to perform imputation on the train and test set.

1. **Handling Categorical Variables:**

In handling the categorical variables, it is critically important to combine the train and test set beforehand since not all the categories of such variables may show up in the train or the test set and thus we may end up with inconsistencies in the variables (e.g. the same variable may end up having different number of factor levels in the training and test sets.

* + 1. **Conversion of Characters to Factors:** After building the combined set, I converted all the character variables to factor variables. A special scenario that was particularly challenging was that there were several features that represented numerical values, but in the chr format (e.g. a variable representing area as “800” – instead of 800). I manually detected such instances and converted these variables to numerics before the factor conversion.
    2. **One-hot Encoding of Factor Variables:** Factor variables were encoded by creating dummy variables which each dummy variable representing one factor level through binary entries. After the OHE process, the dimensions of the feature space went up to 295 from 79.

**4. Feature Scaling:**

Min-max scaling approach was used in feature standardization. Due to the presence of lots of binary features (as result of OHE), min-max scalar was favored over other approaches (e.g. standard feature scaling using feature mean and variance).

# Building the Regression Model & Evaluation

With the pre-processing steps finished, next step was to build the regression model. I used the following approaches:

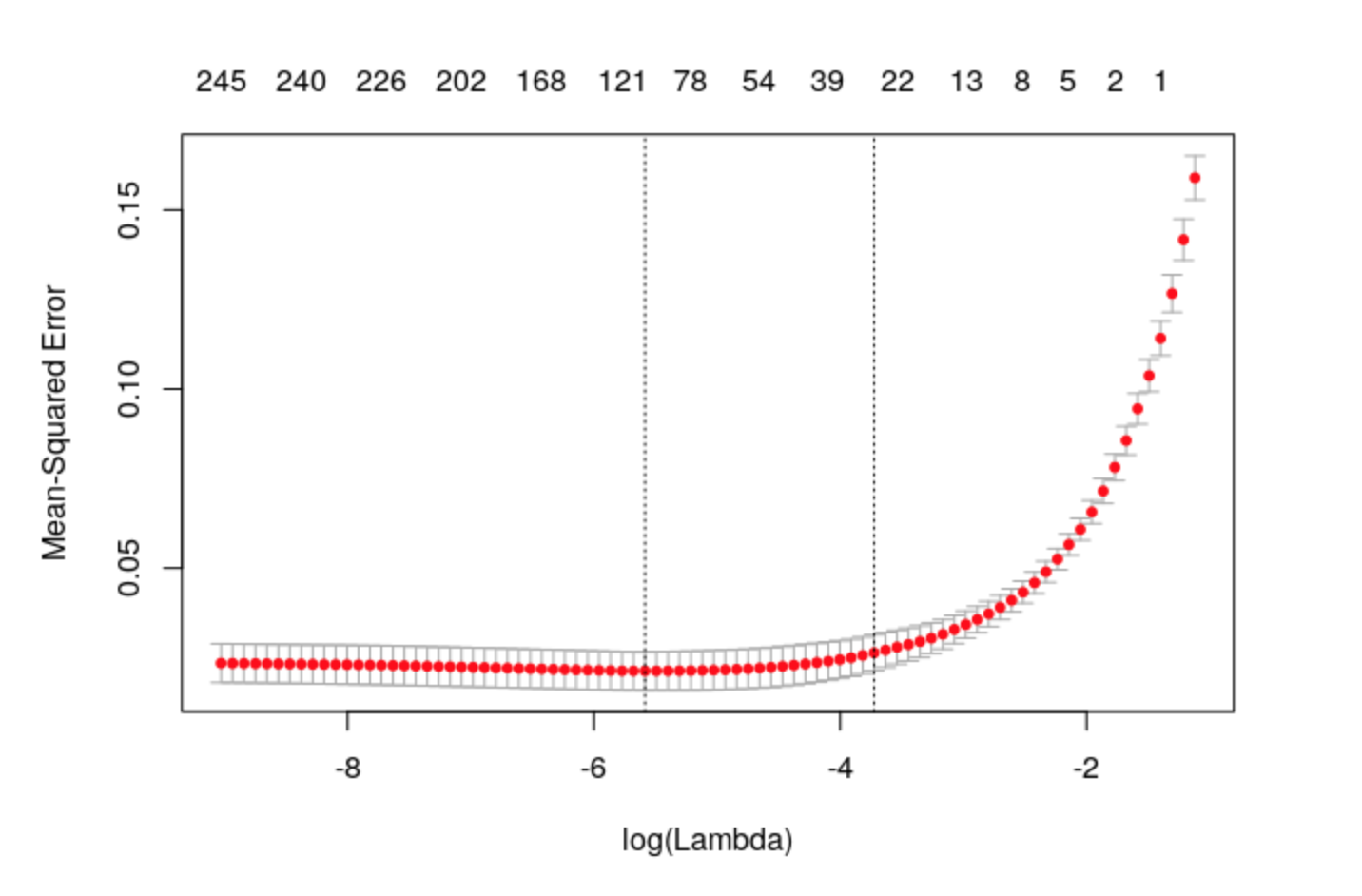
1. Lasso regularized regression model
2. Ridge regularized regression model
3. Gradient Boosting
4. Random Forest

Below are the details and results for each approach. Since the Kaggle competition does not provide the test labels, the cross-validation RMSE was evaluated across the different approach. Information on the test RMSE can be accessed on the Kaggle link provided in the introduction section.

* **Lasso regularized regression model**

Cross validation was used to tune the best lambda for the lasso model and the model resulting in the smallest cross-validation RMSE (lambda = 0.00375) was picked to train the regression model.

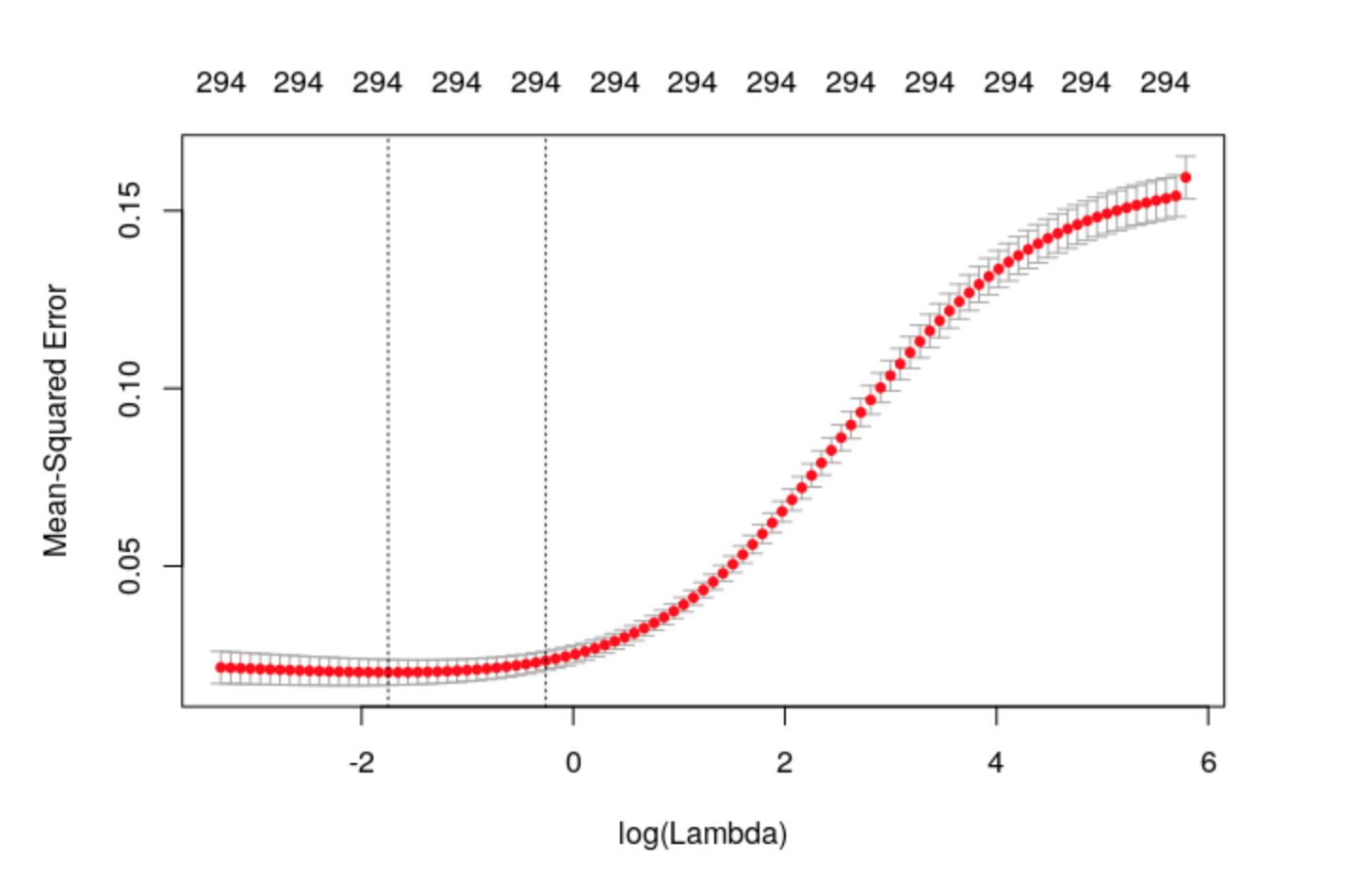
* + Cross-validation Plot



* **Ridge regularized regression model**

Cross validation was used to tune the best lambda for the ridge model and the model resulting in the smallest cross-validation RMSE (lambda = 0.1741) was picked to train the regression model.

* + Cross-validation Plot



* **Gradient Boosting model (GBM)**

K-fold Cross validation (5-folds, repeated 5 times) was used to tune the hyperparameters. The following hyperparameters were tuned:

* interaction.depth
* n.trees
* shrinkage

Parameter n.minobsinnode was set to 10 throughout the cross-validation and metric RMSE was used to determine the best model.

* **Random Forest (RF)**

K-fold Cross validation (5-folds, repeated 5 times) was used to tune hyperparameter mtry and RMSE was used to determine the best model.

# Model Comparison

The following table compare the performance of the four models. The GBM model showed the best performance on the dataset while lasso performed the weakest (even though lasso and ridge regression seem to be very similar in terms of performance – as expected).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **LASSO** | **Ridge** | **GBM** | **RF** |
| **RMSE** | 0.1447693 | 0.1416323 | **0.1243129** | 0.1385458 |

# Conclusion

While there are a lot of factors contributing to the value of a house, data science and its powerful tools make it possible to learn from the historical house purchase data and build models that can confidently predict the final sale price for unseen instances of properties for sale. In this project, we specifically built regression models that can predict the value of a possible future sale price of a house with very small prediction error. The model based on Gradient Boosting was shown to be the most powerful model resulting in the most accurate prediction with the lowest error. Gradient boosting is an ensemble regression technique in which sequential learners make predictions as each learner tries to learn from the mistakes of the previous learners and thus improve the prediction. The results we arrived at confirms that it is possible, and could be potentially very profitable, for the real agencies or private-party sellers to make reasonably accurate predictions on a final sale price of a property and thus help make strategic decisions regarding price throughout the advertising or selling process.