Capstone Project 1: Final Report

Chirag Limbachia

1. Introduction

Buying a house can be a very challenging process. It takes time, patience, and a lot of research to find a house you like and negotiate the right price for it. There are several features that influence the house price such as the neighborhood, square feet area, total number of rooms, garage size, masonary work, utilities, etc.

Can house prices be predicted with the help of such features? Can we identify features that influence house price the most? How important is neighborhood in determining the price? What influences the price more: location or size (sq.ft.)? How much value does remodeling add to the house? These are few of the many interesting questions I intend to explore and answer.

This analysis can be useful for a house buyer in terms of negotiating the right price for the house. It can also be useful for a real estate investor in terms of assessing the realistic value of the housing property.

2. Data Acquisition and Cleaning

Ames Housing dataset, compiled by Dean De Cock is used in this project. The dataset is available on kaggle. The target variable in the dataset is the house sale price. The dataset consists of 1460 rows and 79 columns. Each row corresponds to a house in Ames, Iowa, and each column is a feature that describes some aspects of the house. Description for the 79 features can be found at this link. The dataset is well structured but it required a some amount of wrangling/cleaning.

Distribution of sale price has a positive skew which is treated by applying logarithmic transformation (Figure 1).

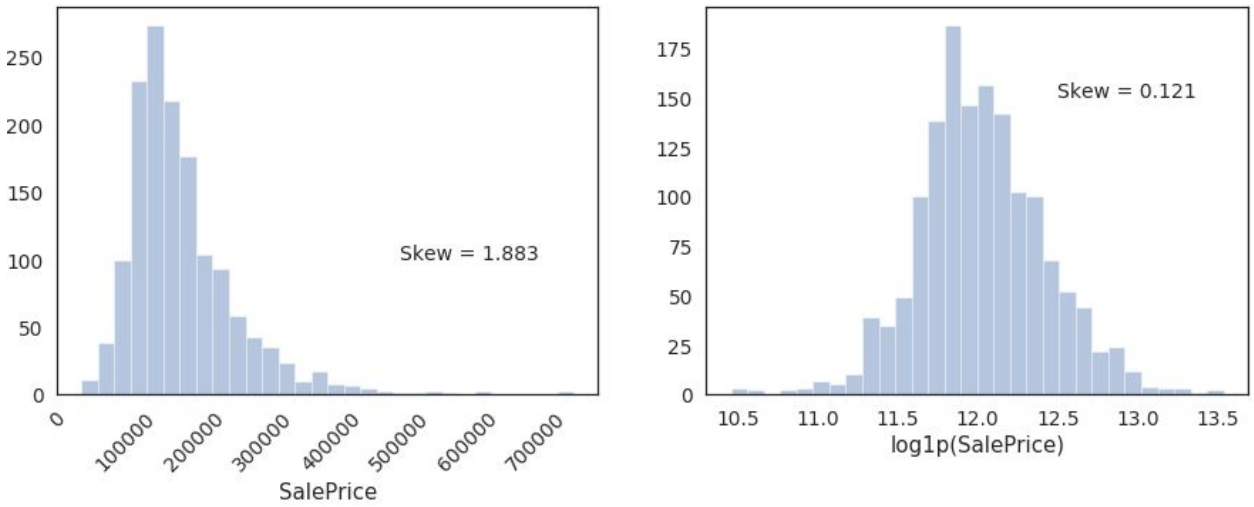

Figure 1. Distribution of sale price (left) and logarithm of sale price (right)

2.1. Outlier identification and removal

Distributions of some of the quantitative features like Above Ground Living Area (aka GrLivArea), and Lot Area are skewed. Skewness of GrLivArea (Figure 2) is mitigated by applying logarithm transform (Figure 3). Skewness in LotArea is mainly due to extreme values. Points beyond LotArea = 60000 sq.ft. seem like outliers and, hence, are removed (Figure 4). After removing outliers, the total number of examples came down to 1454 (from 1460).

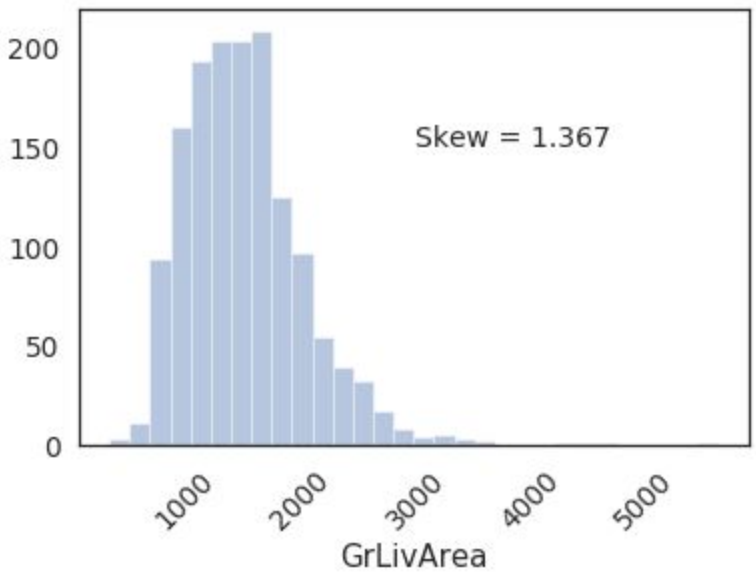

Figure 2. Distribution of Above Ground Living Area (GrLivArea) in sq.ft.

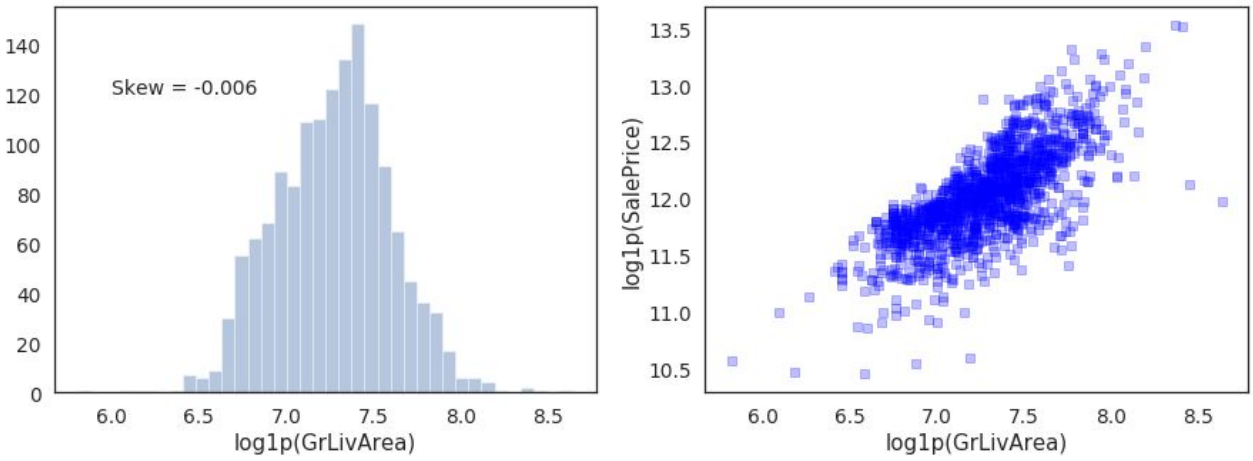

Figure 3. Logarithm of GrLivArea.

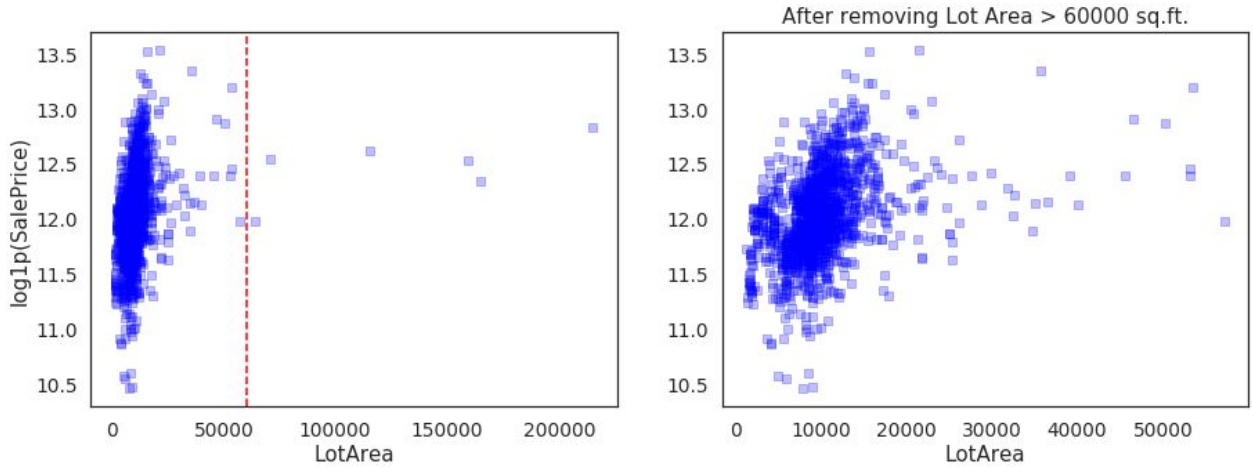

Figure 4. Lot Area vs. log of Sale Price before (left) and after (right) outlier removal.

2.2. Imputation of missing values

Qualitative Features

There are several qualitative features with missing values (see Table 1). All except two features have structurally missing values due to non-existence of the said (or related) feature. For example, if a house doesn't have a Fireplace, it has a missing value for fireplace-quality (FireplaceQu). Therefore, a new category, “Missing”, was imputed at missing instances of such qualitative features.

Other two features: Electrical system (Electrical) and masonry veneer type (MasVnrType) have random missing values. These missing instances are imputed with the most frequently occurring (mode) category.

Table 1. Quantitative features with total number and percentage of missing values.

	Feature	# of missing vals	% of missing vals
0	PoolQC	1378	99.7106
1	MiscFeature	1331	96.3097
2	Alley	1293	93.5601
3	Fence	1109	80.246
4	FireplaceQu	659	47.6845
5	GarageType	73	5.2822
6	GarageFinish	73	5.2822
7	GarageQual	73	5.2822
8	GarageCond	73	5.2822
9	BsmtExposure	38	2.74964
10	BsmtFinType2	38	2.74964
11	BsmtQual	37	2.67728
12	BsmtCond	37	2.67728
13	BsmtFinType1	37	2.67728
14	MasVnrType	7	0.506512
15	Electrical	1	0.0723589

Quantitative Features

There are three features with missing values (Table 2): garage year built (GarageYrBlt), masonry veneer area (MasVnrArea), and length of the street connected to the property (LotFrontage in feet),

GarageYrBlt: Values are missing for those houses that do not have a garage. Median of GarageYrBlt is imputed for missing GarageYrBlt instances.

MasVnrArea: Related to the qualitative feature MasVnrType. Therefore, using MasVnrType as the grouping variable, group specific MasVnrArea medians are imputed at missing instances of MasVnrArea.

LotForntage: Related to two qualitative features: Lot Configuration (LotConfig) and shape (LotShape). Again, group specific medians are computed and imputed for missing instances of LotFrantage.

Table 2. Numerical features with total number and percentage of missing values.

	Feature	# of missing values	% of missing values
0	LotFrontage	253	17.5329
2	GarageYrBlt	81	5.61331
1	MasVnrArea	8	0.554401

2.3. Data Augmentation

The Neighborhood feature of the dataset provides the name of the neighborhood in which the house is located. Using this information, geo coordinates and zip codes are extracted from Google's Geocoding API. Furthermore, using the zip codes, demographic information such as median household income and median home value is also gathered using the *uszipcode* library.

For a detailed process of data wrangling, check this notebook.

3. Data Insights

This section highlights some interesting trends in the data. It will also ask and address some interesting questions that are not centric to the problem at hand (i.e., prediction of sale price), but would be useful in developing an understanding about housing in Ames, Iowa.

House sale price is highly correlated with the overall house quality (OverallQual); better quality houses have a higher sale price (Figure 5).

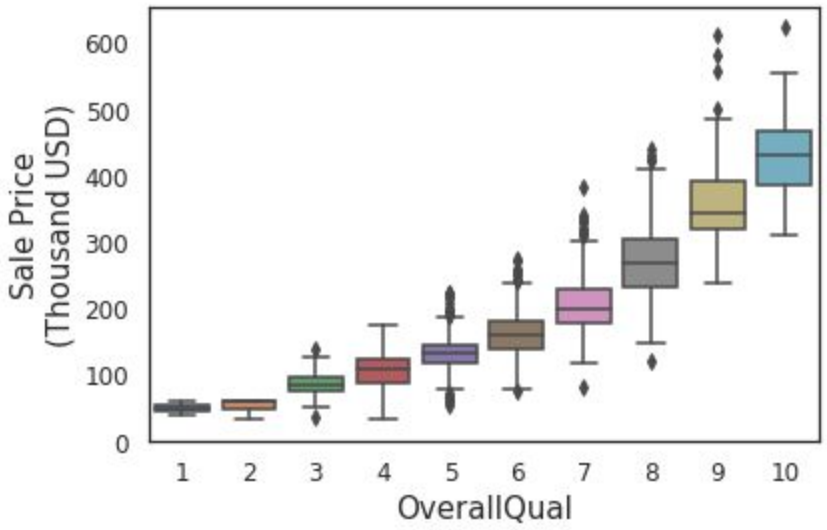

Figure 5. Influence of house's overall quality on sale price. Better quality houses cost more.

Sale price is also correlated with features that reflect size of the housing property such as, size of the living area above ground, garage area, total basement area, area of the first floor, lot area, total number of rooms, number of full bathrooms, etc (Figure 6).

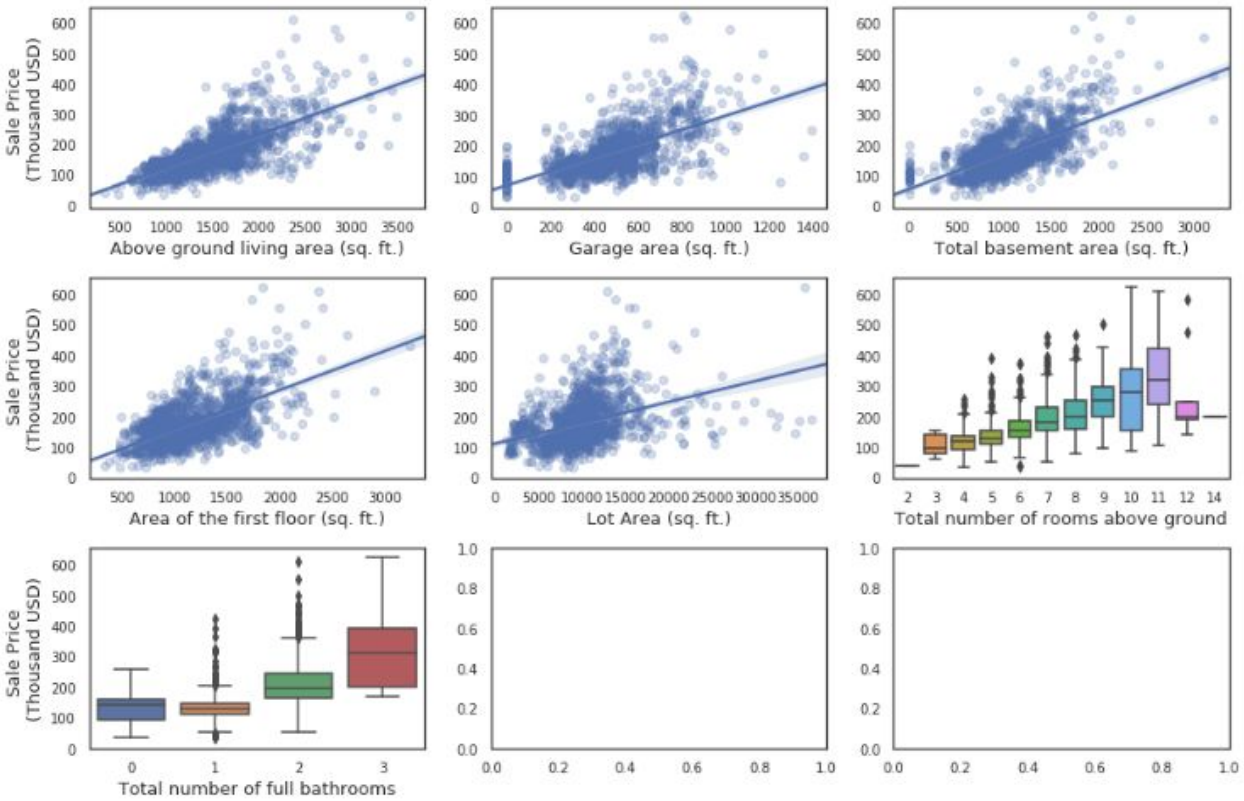

Figure 6. Influence of variables that reflect size of the housing property on sale price.

Neighborhood has an influence on the sale price. Houses in Northridge (NoRidge), Northridge Heights (NridgeHt), and StoneBrook (StoneBr) have highest average sale prices. Whereas, houses in Meadow Village (MeadowV), Iowa DOT and Rail Road (IDOTRR), and Briardale (BrDale) have lowest average sale prices (Figure 7).

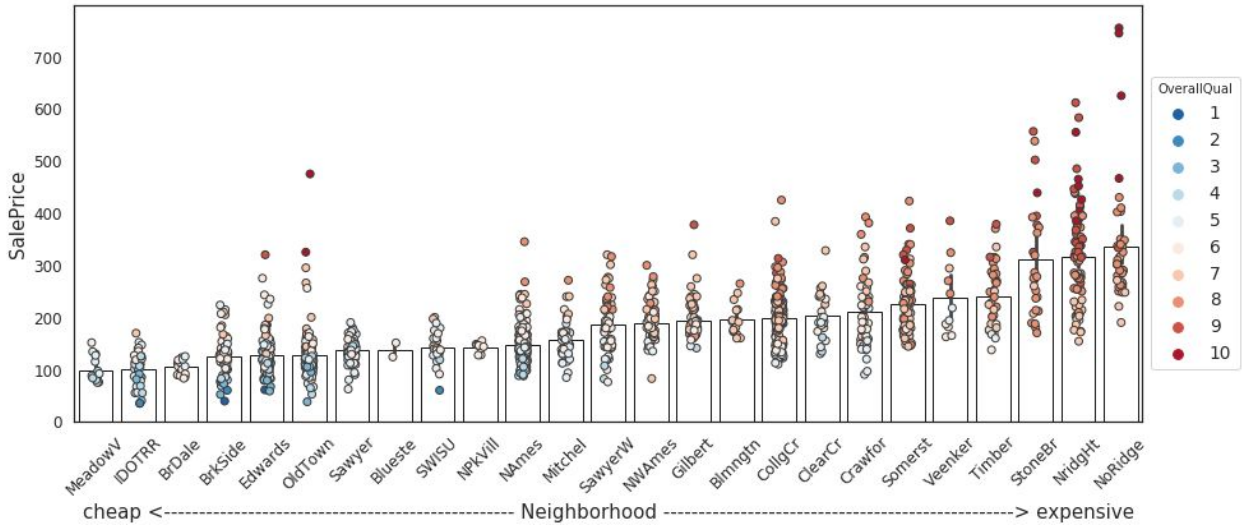

Figure 7. Average sale price of houses by neighborhood.

Northridge Heights, Northridge, and Stone Brook also have the highest concentration of better quality houses. This explains why houses in these neighborhoods are expensive (Figure 8).

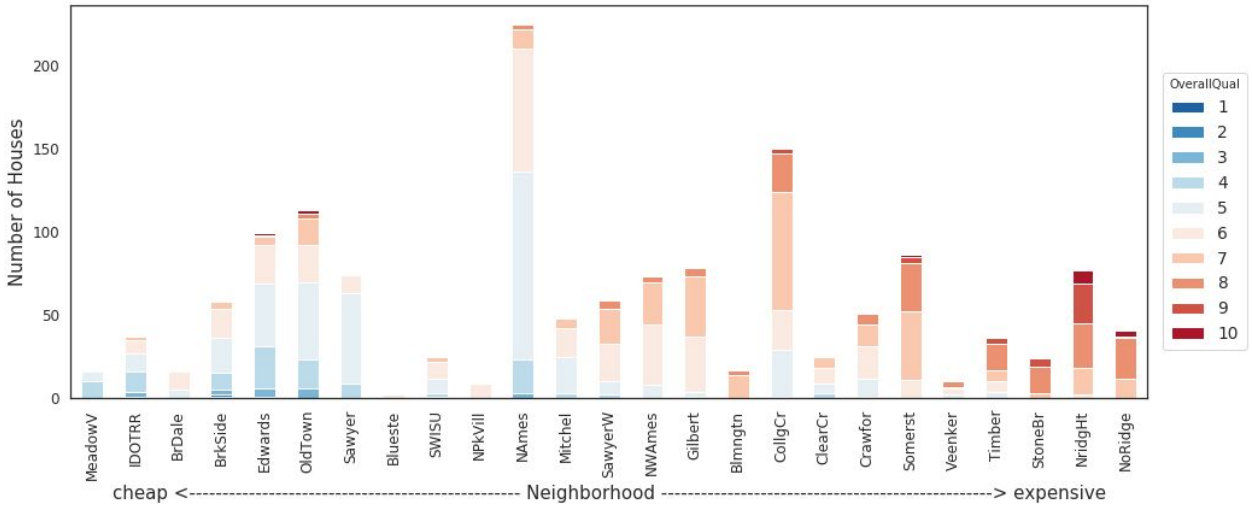

Figure 8. Total number of houses in each neighborhood. Colormap reflects the number of houses of a particular quality.

Expensive neighborhoods like NoRidge, NridgeHt, and StoneBr tend to have houses that either have a finished or a roughly finished garage. Rarely do they have houses with no garage. Whereas, cheaper neighborhoods like MeadowV, IDOTRR, and BrDale mostly have houses with unfinished or no garage (Figure 9).

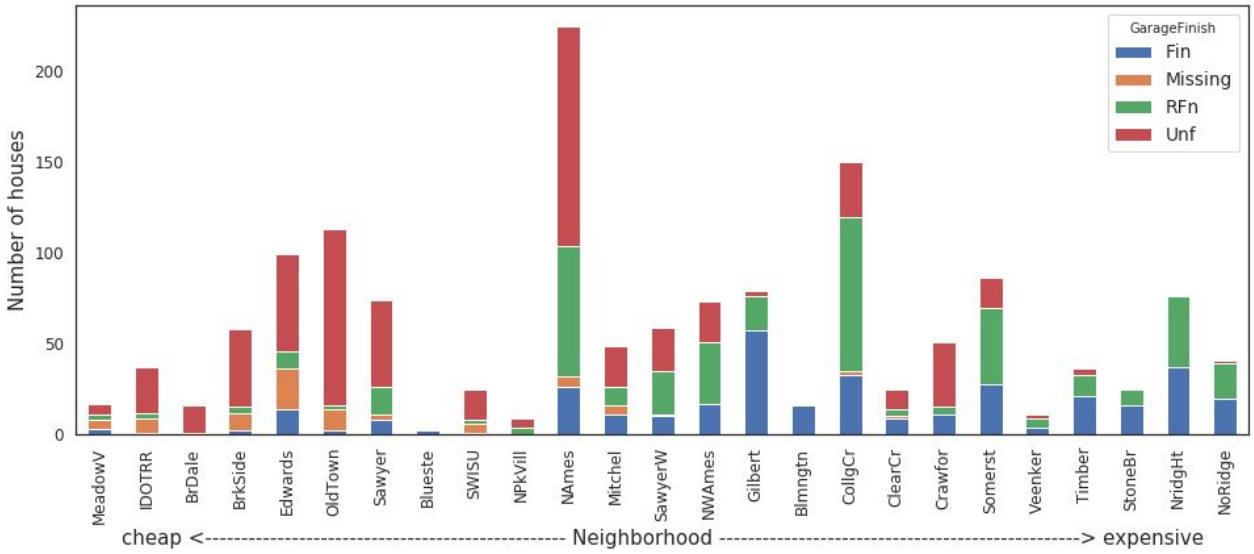

Figure 9. Total number of houses in each neighborhood. Colormap reflects the number of houses of a particular kind of garage finish.

Attached garages are common across all except a few cheaper neighborhoods. Built-In garages are more common amongst houses in expensive neighborhoods. Whereas, detached (Detchd) or no garage are most common in cheaper neighborhoods (Figure 10).

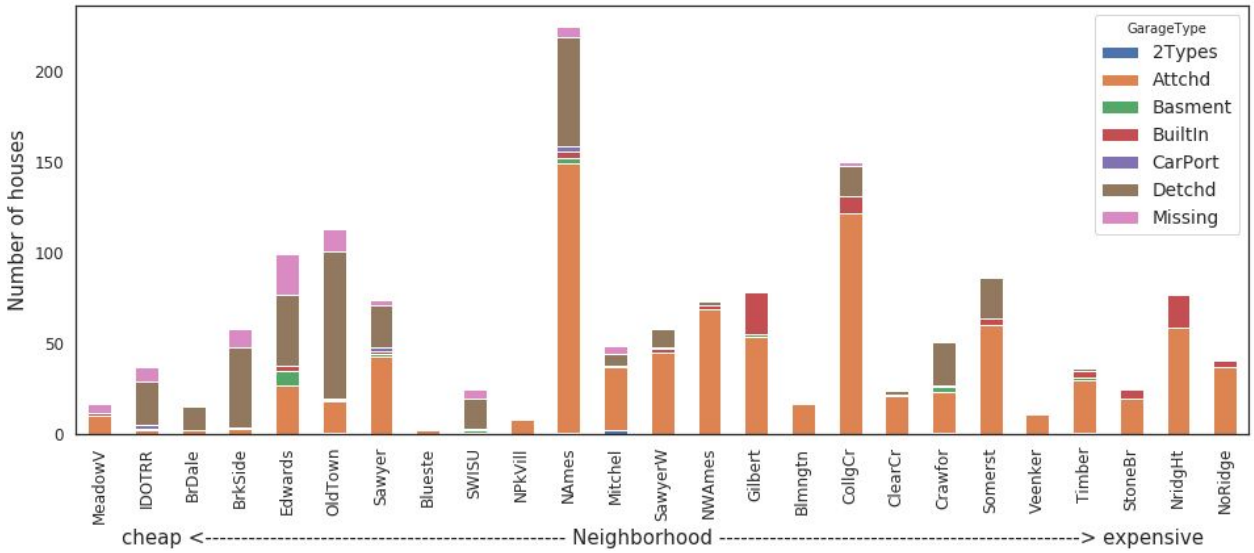

Figure 10. Total number of houses in each neighborhood. Colormap reflects the number of houses of a particular type of garage.

Most houses are remodeled within a couple years after they were built (Figure 11).

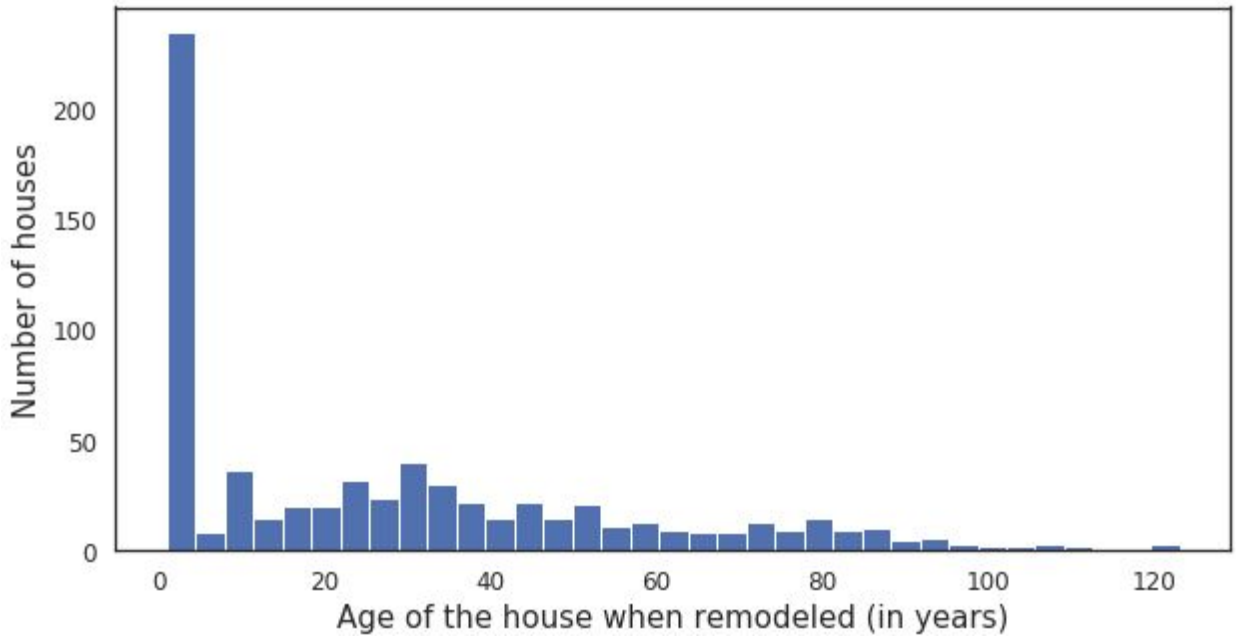

Figure 11. Number of houses remodeled at a particular age.

All houses were sold between 2006-2010 (Figure 12).

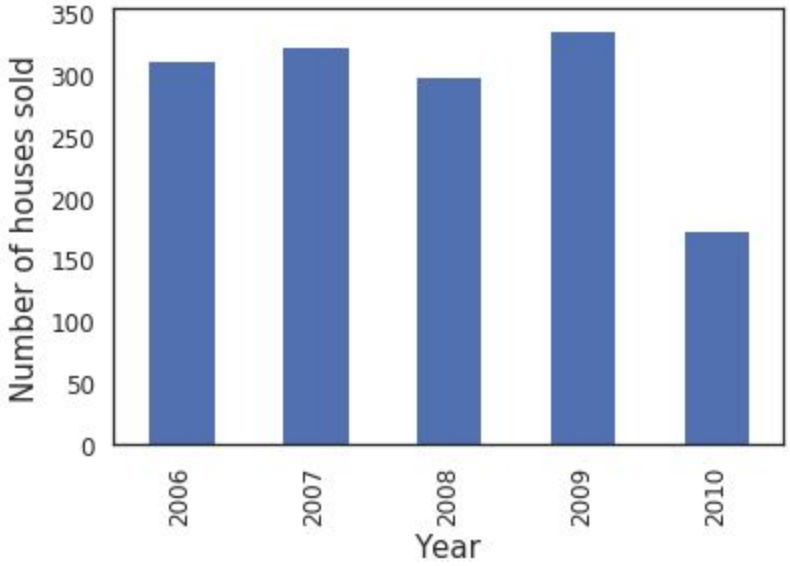

Figure 12. Number of houses sold each year.

Most houses were sold during summer; highest in the month of June (Figure 13).

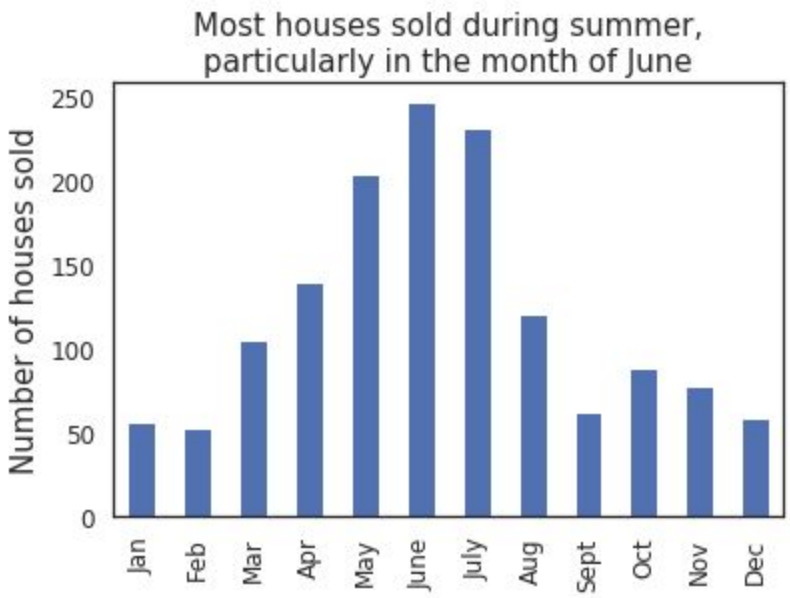

Figure 13. Number of houses sold in a particular month.

Mean sale price seems to fluctuate across the year. It falls by a few thousand dollars (roughly $10,000) by the end of winter (in April) and starts rising again by May-June, peaking in September. This explains why most houses are sold during summer (Figure 14).

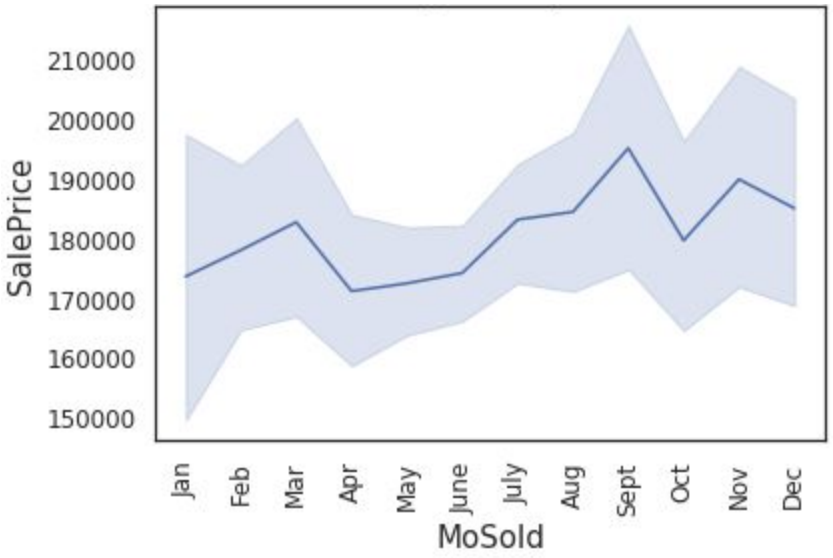

Figure 14. House price fluctuation across a year.

Houses located in floating village residential (FV) zones are most expensive, whereas those located in the commercial (C (all)) zones are the cheapest (Figure 15).

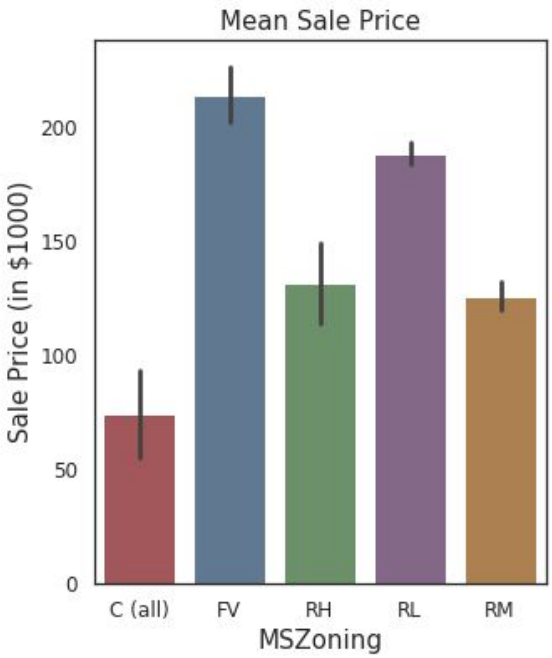

Figure 15. Average Sale Price by Zoning.

Most houses are located in low density residential (RL) zones. In Somerset, most houses are located in the floating village (FV) residential zone (Note: houses in this zone also have the highest average sale price). Only in the Iowa DOT and Rail Road (IDOTRR) neighborhood do we find houses located in commercial (C all) zones (Note: houses in this zone have the lowest average sale price).

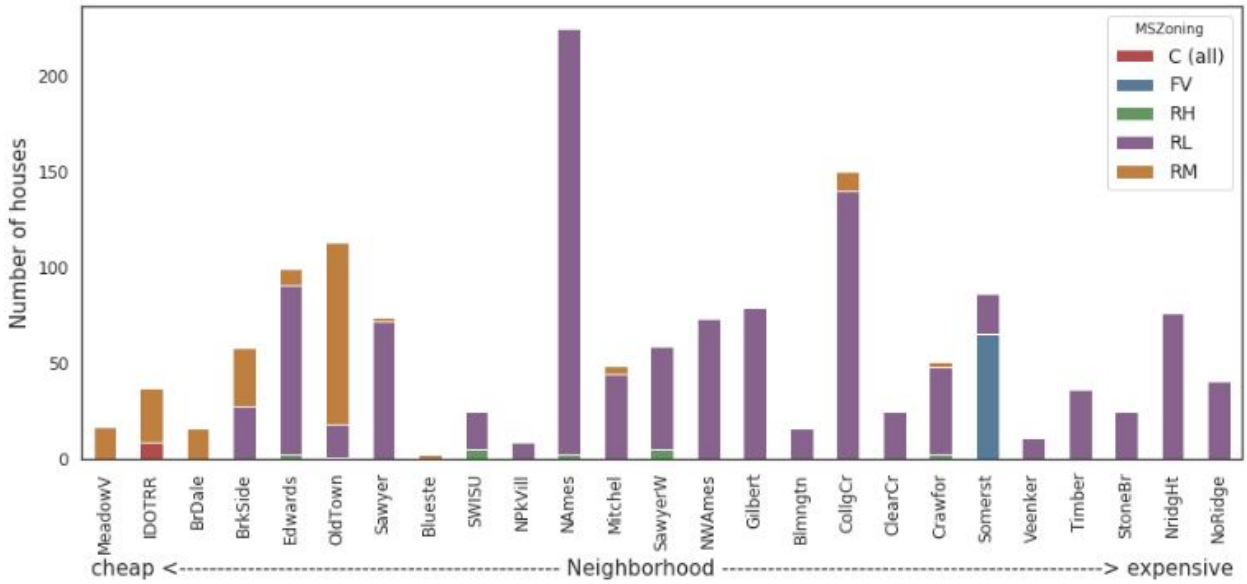

Figure 16. Average Sale Price by Neighborhood and Zoning.

For more exploratory data analysis, check this notebook.

4. Feature Engineering

4.1. Quantitative Features

Feature correlation matrix

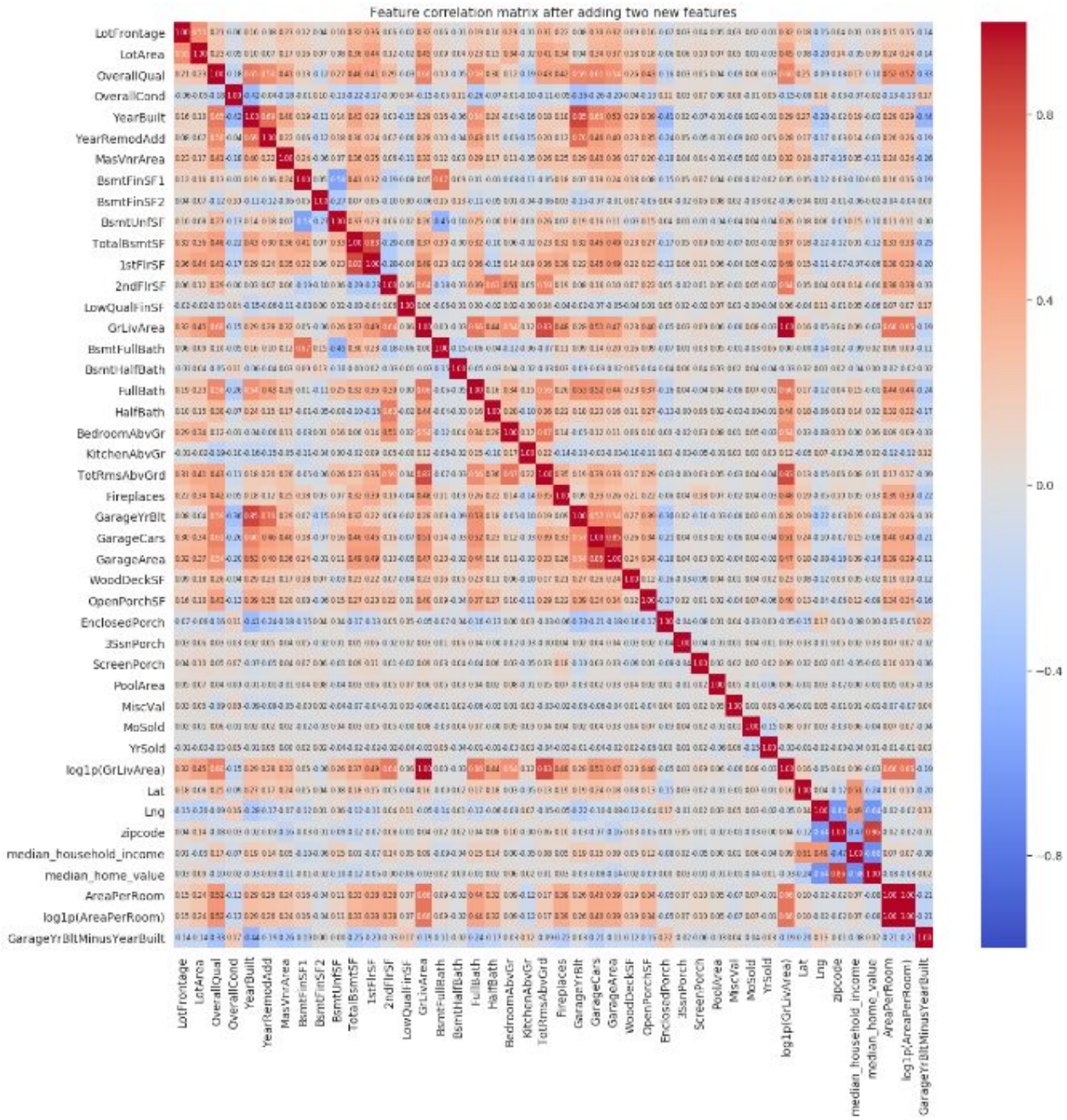

Figure 17. Pairwise Spearman correlations of all numerical features.

Several features are highly correlated with one another. Highest positive correlation is between house and garage year built (YearBuilt and GarageYrBlt). This is not unexpected, as most house garages are built along with the house. A new feature, GarageYrBltMinusYearBuilt, is added which models the difference between the year in which the garage and the house was built.

Second highest positive correlation is between the total number of rooms (TotRmsAbvGrd) and the logarithm of GrLivArea. This is also expected as houses with greater area tend to have more number of rooms. Area per room (AreaPerRoom) is computed by dividing GrLivArea by TotRmsAbvGrd and added as a new feature.

Correlation of the newly added features with the rest of the features.

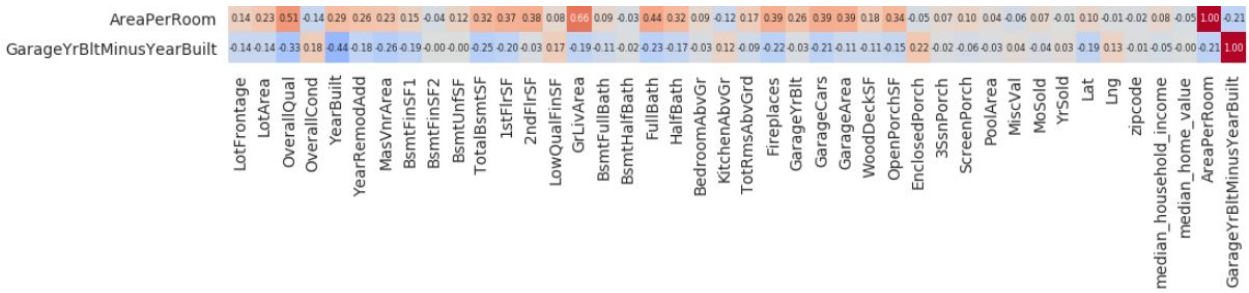

Figure 18. Spearman correlation of newly added features with the rest of the features.

Unlike the parent features, the newly added features are not that highly correlated with the remaining features. Neither are they highly correlated with the parent features.

Correlation with the target (sale price)

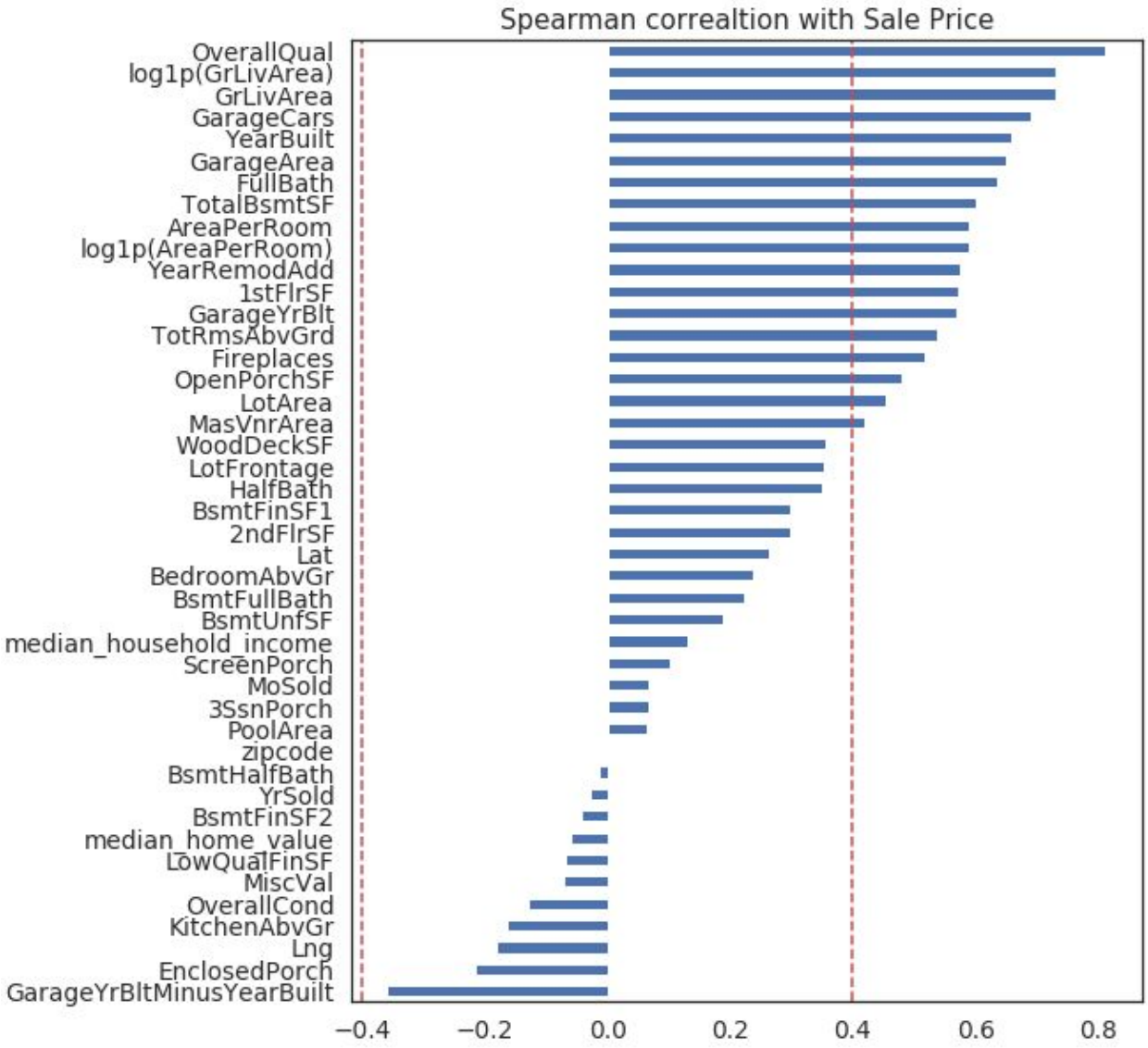

Figure 19. Feature correlations with the sale price.

Several features are highly correlated (> 0.4) with the sale price. One of the new features, AreaPerRoom, is also highly correlated (~0.6) with the sale price. In fact, it is more correlated than one of its parent features from which it is derived (TotRmsAbvGrd). GarageYrBlitMinusYearBuilt is another new feature. It has the highest negative correlation with the sale price. Both new features are goods candidates for predicting the sale price.

4.2. Qualitative Features

Kruskal (non-parametric) ANOVA

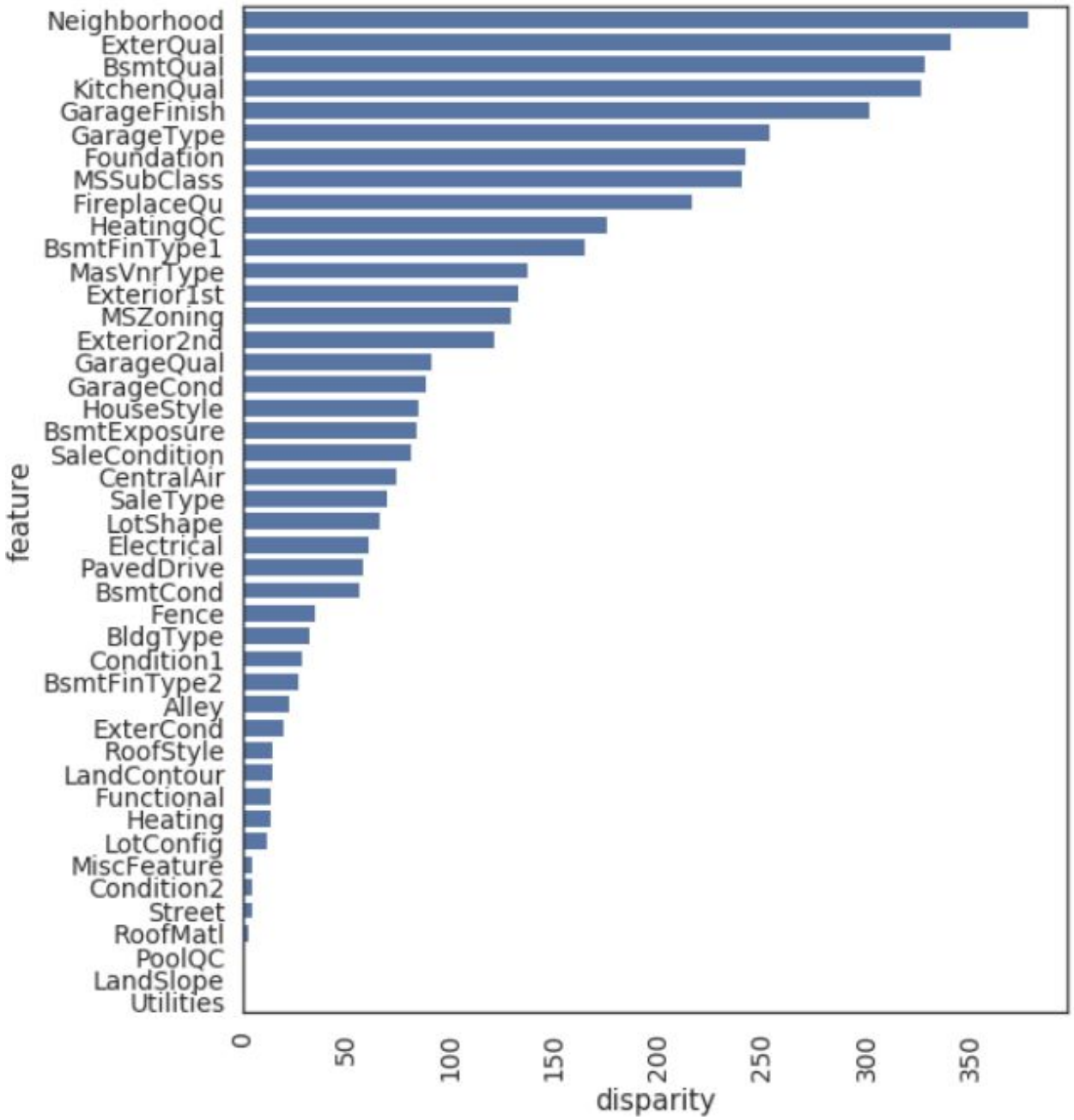

Figure 20. Non-parametric ANOVA disparity measure for all categorical features with respect to sale price.

The above bar plot shows the influence of categorical features on the sale price. Features with higher disparity have greater influence on the sale price. Disparity is derived from the p-value

obtained from ANOVA analysis of a particular feature. Features like Neighborhood, exterior quality (ExterQual), basement quality (BsmtQual), kitchen quality (KitchenQual), garage finish (GarageFinish) seem to have high influence on the sale price.

Sixteen out of the 44 qualitative features are actually ordinal features. Based on the provided feature descriptions (can be found at this link), appropriate ordinal labels were assigned to the levels of the ordinal features. Remaining 28 qualitative features were one-hot encoded.

Feature pairs that showed a correlation of 0.4 or greater were multiplied to create interaction features. The final training dataset consists of 1454 examples and 437 features. For details on feature engineering, check this notebook.

5. Model Building

5.1. Pre-processing

All features in the dataset are now in numeric form. Few more preprocessing steps need to be undertaken before the dataset can be used to train a model.

5.1.1. Data splitting

The dataset is splitted into 70-30% ratios. Train set retains the 70% of the examples and the test set retains the remaining 30% of the examples.

5.1.2. Standardization

All features except those features that are one-hot encoded (and their interactions) are standardized. Standardization involved subtracting the mean and dividing by the standard deviation. This ensures that the freaters have a mean of 0 and a standard deviation of 1. Standardization was carried out only on the train set. The learned standardization parameters were applied to the test set to standardize test set features.

5.2. Evaluation Metric

Since this is a regression problem, Root-Mean-Squared-Error (RMSE) between the logarithm of predicted and observed sale price is used as the evaluation metric. Using the logarithm of the sale price as the target variable ensures that the errors in predicting expensive and cheap houses will affect the results equally.

For a more general assessment of performance, a five-fold cross-validation (CV) approach was adopted. In a five-fold CV, the training data is divided into 5 equal length folds. The model is trained on 4 out of 5 folds and tested on the 5th held-out fold. This process is repeated until

every fold has been treated as a held-out fold. RMSE is computed on each held-out fold. This gives five RMSE values. Mean of the five RMSE values is used as the model performance metric. Lower the mean RMSE value, better the model performance.

5.3. Linear Regression

Scikit-learn's LinearRegression class is used to train a linear model on the training set. Performance of the model on the training data is fair (mean RMSE = 1.89), but extremely poor on the validation set (mean RMSE = 32.9). Observed vs. Predicted values in Figure 21 indicates that the model fits well on the training data, but not on the validation data. This is a textbook case of overfitting. A solution to the problem of overfitting is regularization. Regularization reduces the wigglyness of the fitted curve and makes it smoother by shrinking the contribution of every feature. Next three subsections present different kinds of regularizations.

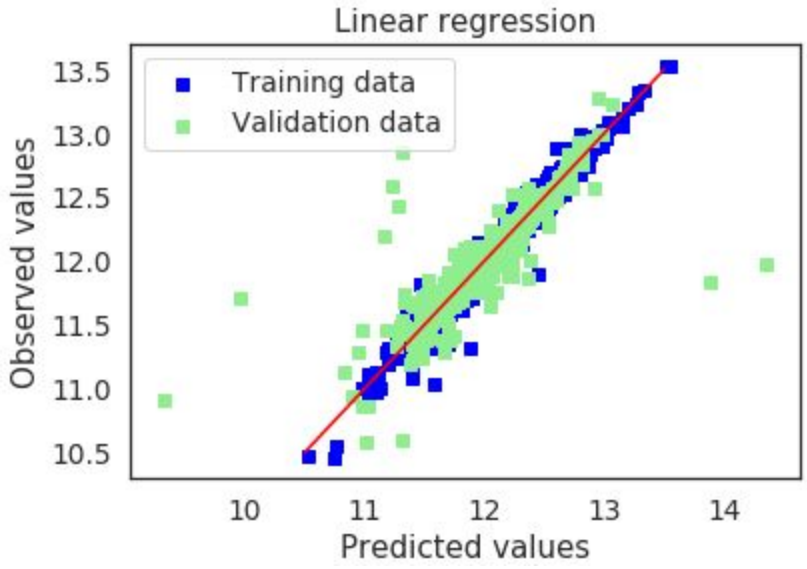

Figure 21. Linear Regression: Observed vs. Predicted

5.4. Ridge Regression

Ridge regression is a variant of a regularized linear regression. It shrinks contribution of the features by minimizing the L2-norm of the feature parameters. Ridge class of scikit-learn library is used to train a ridge regression model. Performance of ridge regression is much better on both training and validation data as compared to linear regression. The model is further fine tuned by finding an optimal value for the learning rate (alpha) via a grid search algorithm provided by the scikit-learn library. Alpha = 30 gives the best performance with mean training RMSE = 0.127 and mean validation RSME = 0.120. Note that both training and validation RSMEs are very close, and the model seems to perform better in the validation data as compared to the training data. This indicates that the model is not overfitting anymore and is

generalizable to out-of-sample data. Figure 22 also shows that the ideal fit (the red line) aligns well with both training and validation data. However, there are models to be tried.

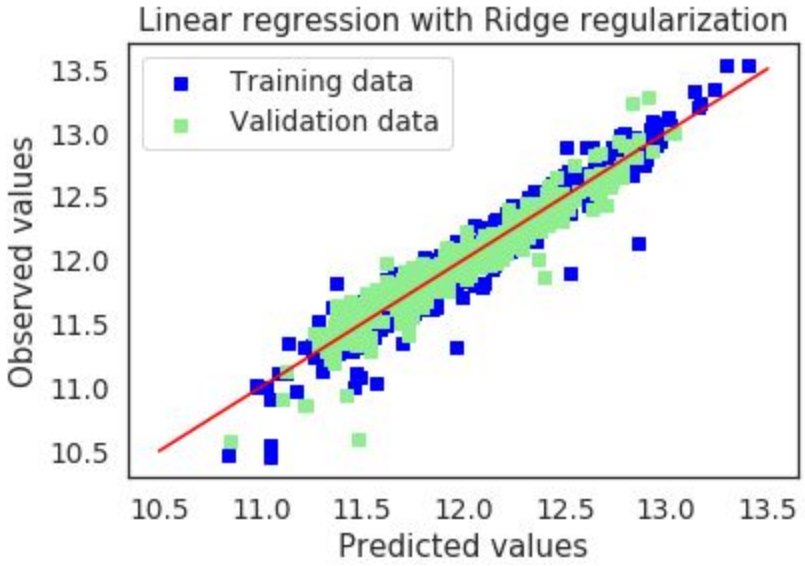

Figure 22. Ridge Regression: Observed vs. Predicted

5.5. Lasso Regression

Lasso regression is another variant of regularized linear regression. It shrinks contribution of the features by minimizing the L1-norm of the feature parameters. Lasso class of scikit-learn library is used to train a lasso regression model. The model is fine tuned by finding an optimal value of alpha via a grid search algorithm. Alpha = 0.0003 gives the best performance with a mean training RMSE = 0.126 and validation RMSE = 0.124. Interestingly while its performance on the training data is better than that of ridge regression, its performance on the validation is rather poorer as compared to that of ridge regression (Figure 23). However, the gap between training and validation RMSE is much less for lasso as compared to ridge. This suggests that lasso is more generalizable than ridge. Hence lasso regression is the best performing model thus far. Figure 24 shows observed vs. predicted value scatter plot.

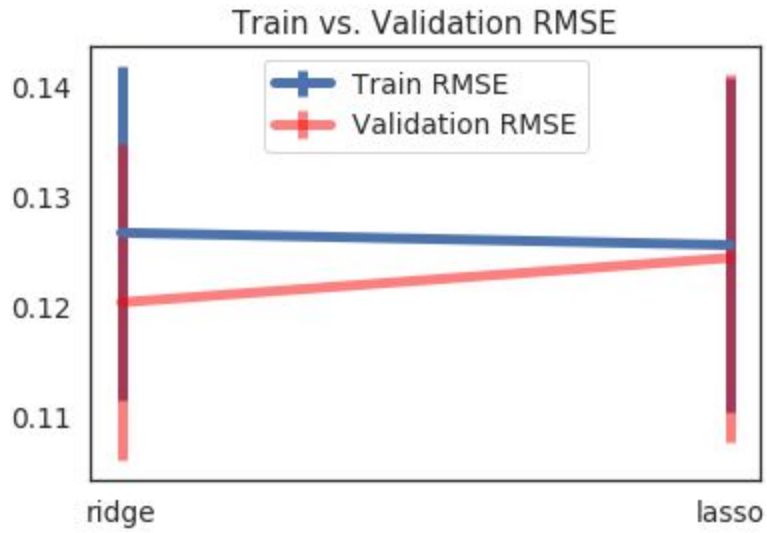

Figure 23. Train and validation RMSEs for Ridge and Lasso Regression Models.

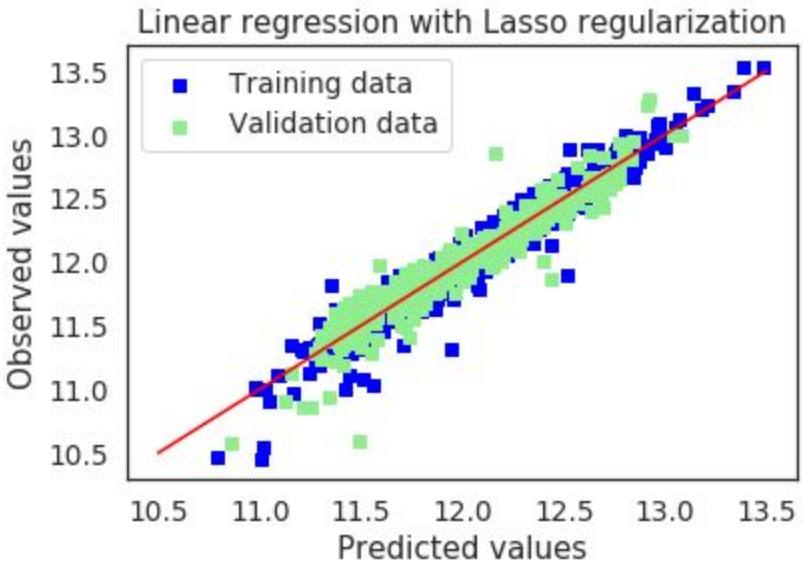

Figure 23. Lasso Regression: Observed vs. Predicted

5.6. Support Vector Machine

A support vector machine regression plots the data in an n-dimensional space (where n is the number of features) and finds a hyperplane that contains the majority of the data points. SVR class from the scikit-learn library is used to train a SVM regression model. Both training and validation RMSE values are high as compared to those of the best performing model thus far, lasso regression (Figure 24). Hence SVM is out of consideration.

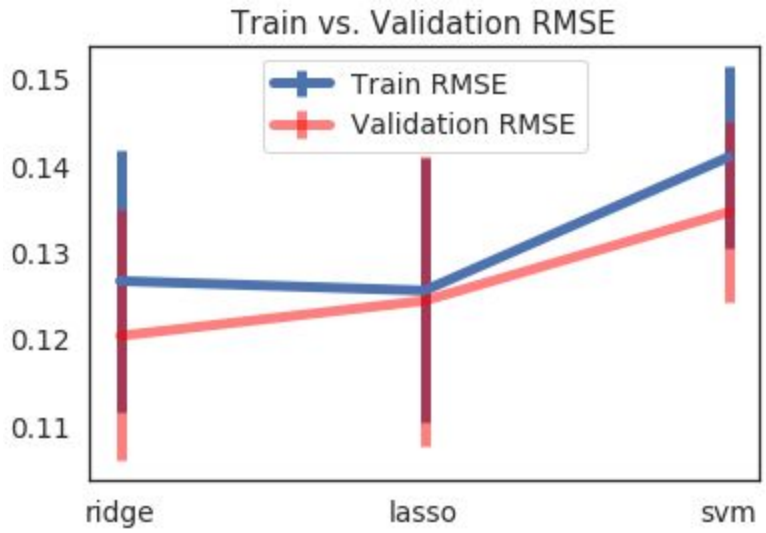

Figure 24. Training and Validation root-mean-squared-errors (RMSE) for different models.

5.7. Random Forest

The RandomForestRegressor class from the scikit-learn library is used to train a random forest regression model. The model was further fine-tuned by finding optimal parameters such as number of trees and maximum number features to consider when performing a split. Performance of the model is even poorer than SVM (Figure 25). Hence random forest is also out of consideration.

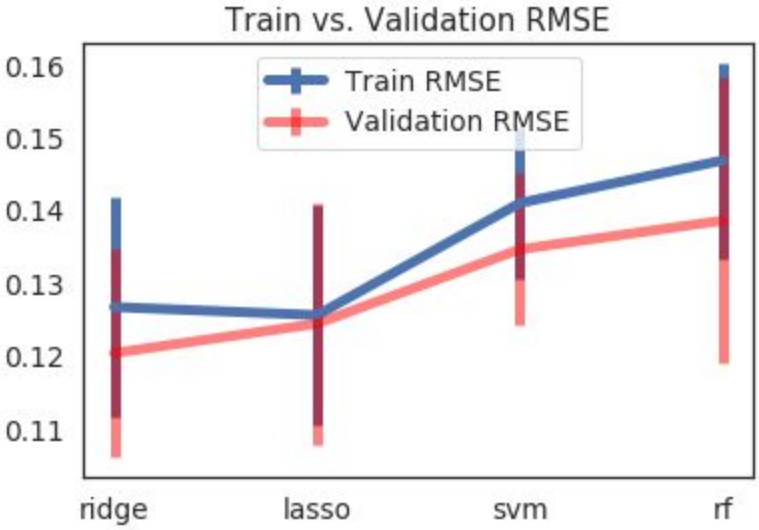

Figure 25. Training and Validation root-mean-squared-errors (RMSE) for different models.

5.8. AdaBoost & Gradient Boost Regression

Adaptive and gradient boost models are also trained and tested. However they did not perform any better than the best performing model, lasso regression (Figure 26). Hence both adaboost and gradient boost are also out of consideration.

6. Final Model

Lasso regression yields the best performance. Figure 25 also mentions performance of an elastic net regression model (enet) which looks equal to lasso. This is because, during fine-tuning of the enet model, the best parameters found essentially made it a lasso regression model (l1_ratio = 1). Out of 437 features, 271 features are eliminated due to l1 penalty, and only 166 features turned out to be useful in driving predictions. Top 20 features that have the highest influence on the sale price are mentioned in Figure 26. Condition2_PosN seems to have the highest influence. Condition2_PosN is a one-hot encoded categorical variable which flags houses that are near a park, greenbelt, in Ames, Iowa. Table 3 presents performance measures (mean and standard deviations for RMSE) of every model tried.

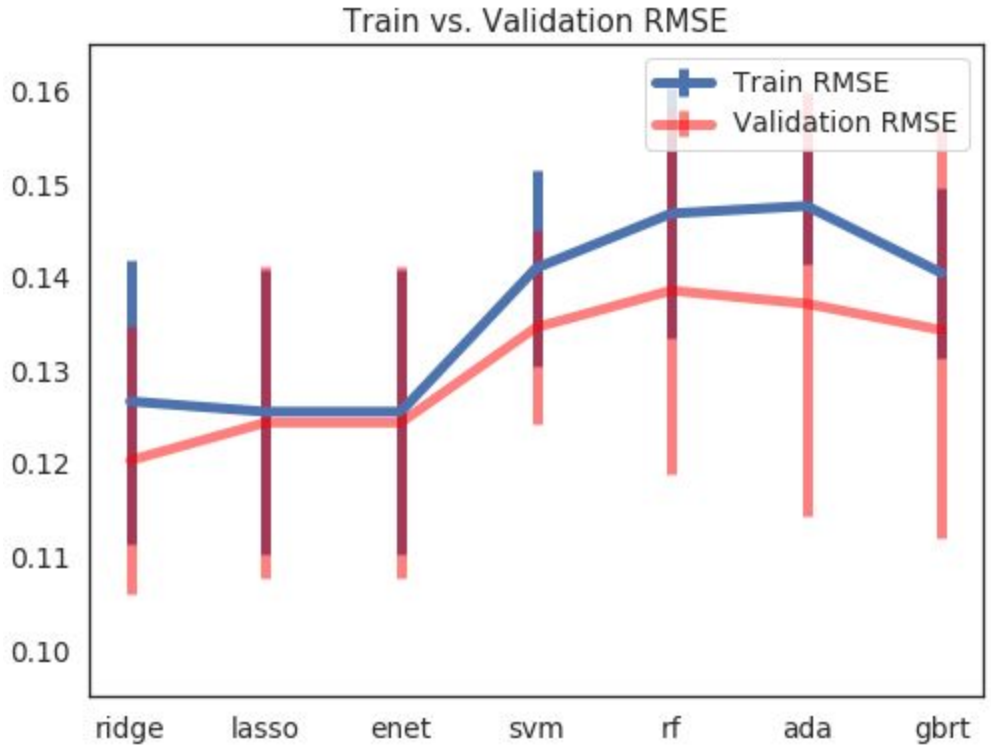

Figure 25. Training and Validation root-mean-squared-errors (RMSE) for different models.

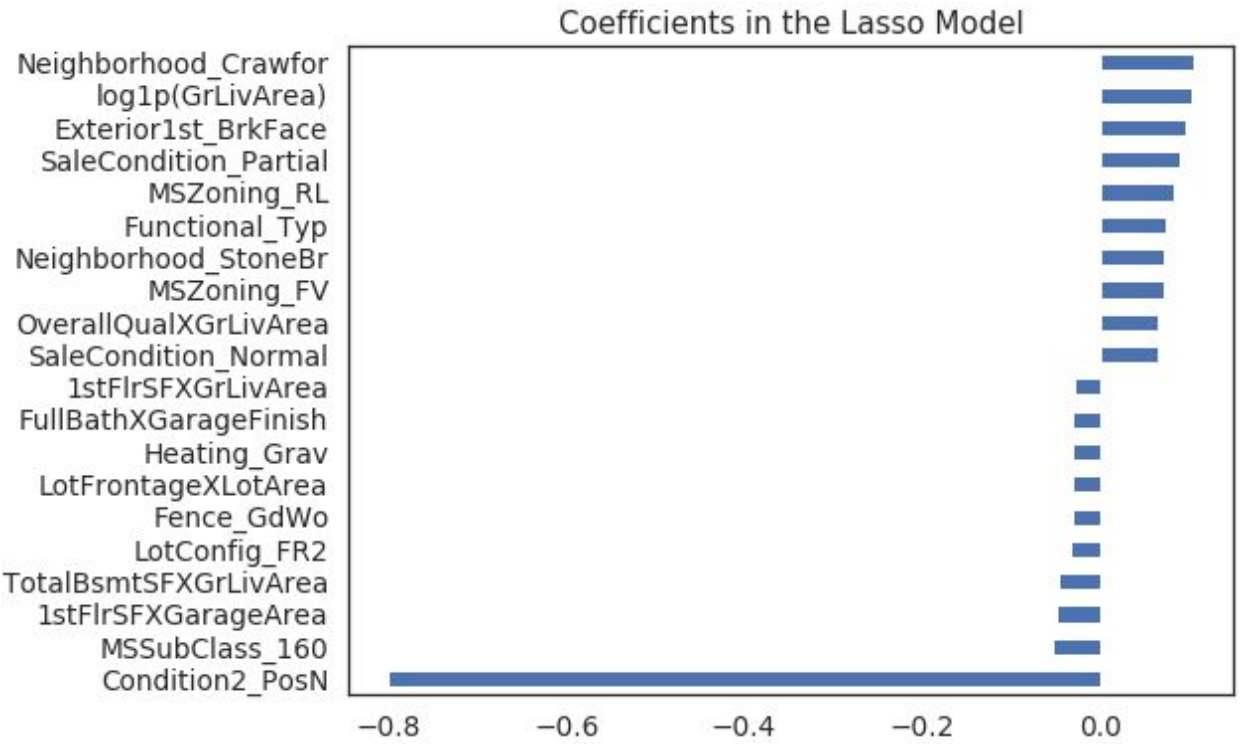

Figure 26. Coefficient of Lasso Regression Model.

Table 3. Mean and standard deviation of the RMSE on train and validation datasets.

	model	train_mean_rmse	train_std_rmse	val_mean_rmse	val_std_rmse
0	LR	1.886960	1.854015	32.976174	21.253336
1	ridge	0.126681	0.017003	0.120349	0.016109
2	lasso	0.125569	0.017000	0.124396	0.018678
3	enet	0.125569	0.017000	0.124396	0.018678
4	svm	0.140985	0.011784	0.134616	0.011595
5	rf	0.146845	0.015059	0.138575	0.021999
6	ada	0.147650	0.007012	0.137160	0.025465
7	gbrt	0.140443	0.010199	0.134328	0.024832