Springboard Data Science

Capstone 2

Advanced Regression to Predict Housing Sale Prices

Final Report

Table of Contents

01

Executive Summary

Springboard Data Science

May 2018

01 Executive Summary

This project is based on an active competition on Kaggle, where housing characteristics are used to predict housing sale prices in Ames, Iowa. The dataset contains 79 characteristics for almost 3,000 homes, half of which are used for training a regression model, while the other half is used to test the model and score it.

Predicting home prices is a classic regression problem and the underpinning of many real estate-based tech companies, as well as an essential part of business strategy for developers and realtors. Although regression was first introduced in the early 1800s, it still remains one of the most popular predictive algorithms for its efficiency and efficacy. This project used some advanced regression techniques that have built upon the original OLS regression, such as ridge and lasso regression, as well as some modern techniques such as gradient boosting.

In the end, OLS linear regression ended up outperforming both ridge and LASSO regression, as well as the ensemble methods, though almost all of the models performed similarly. The worst performing models were the linear regression run with PCA, and the LASSO regression model.

Finally, the best performing model was used to predict housing sale prices on the test data set from Kaggle, adn submitted. As of now, its RMSLE score is 0.145 and is in the top 45% of all models submitted.

02
Background

Springboard Data Science

May 2018

02 Background

02.01 Purpose

Predicting home sale prices is a classic regression problem that remains crucial in the real estate industry and continues to be refined as new regression methods are developed.

The intent of this project is to explore different regression techniques and examine ways to improve the accuracy of regression models used to predict housinng prices.

Regression models used in this study include: linear regression, Lasso regression, random forest regression, and xgboost.

02.02 Intended Audience

There are several groups who would be able to leverage this data to make informed decisions:

Realtors
Professionals who sell homes need to be strategic about how much they list homes for--too high and they may get few or no offers, too low and the home may sell for less than it could have. Having as accurate a sale price as possible for a home would be extremely beneficial for realtors.

Economists
The housing market is an extensive area of study for economics, and has many implications cities and the overall economic well being of the market. Understanding housing price trends is an area of interest for many economists and economic development professionals.

Real Estate Tech Companies
There are several real estate tech companies with existing price prediction models, such as Zillow's "Zestimate" and Redfin's "Redfin Estimate." These price prediction algorithms are used to predict the prices of homes that are both on and off the market, and are also used to predict whether a home will go above or below its listing price. While those price prediction models are far more complex and accurate, this project will provide a basic understanding of some techniques that can be used to predict home prices.

02.03 Data Source

Data Source
The data set is part of an active Kaggle competition. The data is split into two files, one for training a model and one that will be used to score the model. The contest information and data files can be found here: https://www.kaggle.com/c/house-prices-advanced-regression-techniques.

Data Files
For this project, there are two main data files:

train.csv
This file holds data, including sale prices, for 1,460 homes sold in Ames, IA. The data is comprised of 79 features used to describe every home, such as the number of bedrooms, the neighborhood the home was located in, and they type of roofing each home has.

test.csv
This file has data on 1,459 homes sold in Ames, IA, excluding the sale prices. This dataset is used to test and score the model on Kaggle.

Data Dictionary
In addition to the two data tables, there is also a data dictionary which explains what every variable is, the different possible categories (if it is a categorical variable), and what null values indicate.

These features include values such as the total number of rooms in each home, the square footage of different room types, and the type of garages each home has.

There are null values within a number of the features, but they typically indicate that the home referenced does not have the feature, for instance, homes without basements have null values for "Basement Finish Type."

The following is a table of every feature and a description of it.

Feature Name	Feature Description
SalePrice	the property's sale price in dollars. This is the target variable that you're trying to predict.
MSSubClass	The building class
MSZoning	The general zoning classification
LotFrontage	Linear feet of street connected to property
LotArea	Lot size in square feet
Street	Type of road access
Alley	Type of alley access
LotShape	General shape of property
LandContour	Flatness of the property
Utilities	Type of utilities available
LotConfig	Lot configuration
LandSlope	Slope of property
Neighborhood	Physical locations within Ames city limits
Condition1	Proximity to main road or railroad
Condition2	Proximity to main road or railroad (if a second is present)
BldgType	Type of dwelling
HouseStyle	Style of dwelling
OverallQual	Overall material and finish quality
OverallCond	Overall condition rating
YearBuilt	Original construction date
YearRemodAdd	Remodel date
RoofStyle	Type of roof
RoofMatl	Roof material
Exterior1st	Exterior covering on house
Exterior2nd	Exterior covering on house (if more than one material)
MasVnrType	Masonry veneer type

MasVnrArea	Masonry veneer area in square feet
ExterQual	Exterior material quality
ExterCond	Present condition of the material on the exterior
Foundation	Type of foundation
BsmtQual	Height of the basement
BsmtCond	General condition of the basement
BsmtExposure	Walkout or garden level basement walls
BsmtFinType1	Quality of basement finished area
BsmtFinSF1	Type 1 finished square feet
BsmtFinType2	Quality of second finished area (if present)
BsmtFinSF2	Type 2 finished square feet
BsmtUnfSF	Unfinished square feet of basement area
TotalBsmtSF	Total square feet of basement area
Heating	Type of heating
HeatingQC	Heating quality and condition
CentralAir	Central air conditioning
Electrical	Electrical system
1stFlrSF	First Floor square feet
2ndFlrSF	Second floor square feet
LowQualFinSF	Low quality finished square feet (all floors)
GrLivArea	Above grade (ground) living area square feet
BsmtFullBath	Basement full bathrooms
BsmtHalfBath	Basement half bathrooms
FullBath	Full bathrooms above grade
HalfBath	Half baths above grade
Bedroom	Number of bedrooms above basement level
Kitchen	Number of kitchens
KitchenQual	Kitchen quality
TotRmsAbvGrd	Total rooms above grade (does not include bathrooms)
Functional	Home functionality rating
Fireplaces	Number of fireplaces
FireplaceQu	Fireplace quality
GarageType	Garage location
GarageYrBlt	Year garage was built
GarageFinish	Interior finish of the garage
GarageCars	Size of garage in car capacity
GarageArea	Size of garage in square feet
GarageQual	Garage quality
GarageCond	Garage condition
PavedDrive	Paved driveway
WoodDeckSF	Wood deck area in square feet
OpenPorchSF	Open porch area in square feet
EnclosedPorch	Enclosed porch area in square feet
3SsnPorch	Three season porch area in square feet
ScreenPorch	Screen porch area in square feet
PoolArea	Pool area in square feet
PoolQC	Pool quality

Fence	Fence quality
MiscFeature	Miscellaneous feature not covered in other categories
MiscVal	$Value of miscellaneous feature
MoSold	Month Sold
YrSold	Year Sold
SaleType	Type of sale
SaleCondition	Condition of sale

03

Exploratory Data Analysis

Springboard Data Science

May 2018

03 Exploratory Data Analysis

In this section, the training data set is examined and several visualizations are created to understand the distribution of home prices in Ames, Iowa, and different relationships to it.

First, the distribution of the home sale prices is examined in Figure 1 by plotting a histogram of the sale prices in the training data set. It seems the majority of homes fall between the $100,000 - $200,000 range, but the distribution is skewed to the right, with some homes significantly more expensive than the average home

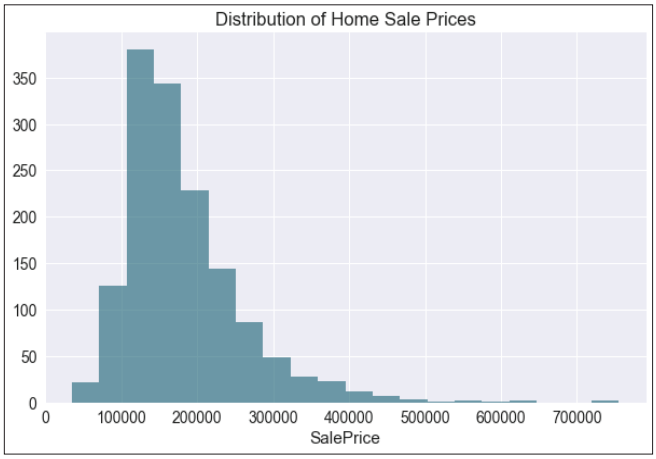


Figure 1

The average home price is about $180,000, with a standard deviation of $79,442. The cheapest home is about $35,000, and the most expensive home is $755,000.

The mean, standard deviation, and other general statistics for the Sale Price column are shown in the table below.

Statistic	Value
count	1460
mean	180921.1959
std	79442.50288
min	34900
25%	129975
50%	163000
75%	214000
max	755000

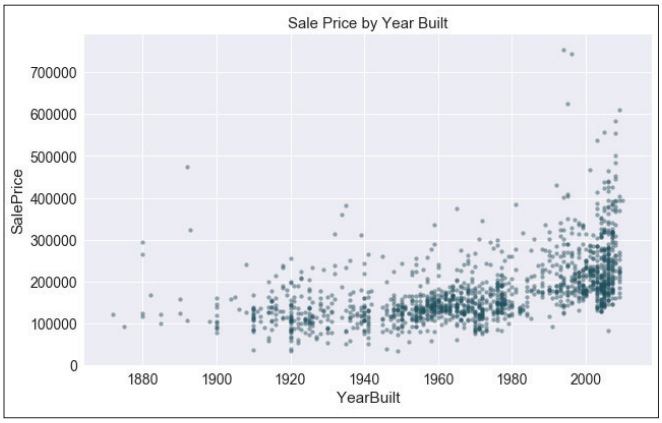


Figure 2

Examining the relationship between the sale price and the year the home was constructed in Figure 2, it appears that newer homes - constructed after the 1990s - had noticeably higher prices, but homes older than that tended to not have a notable difference in sale price. This is unsurprising as homes that have not been lived in tend to sell for more

In Figure 3 below, the sale price is plotted against the above grade and below grade living areas. As expected, there is a correlation between the living area square footage and the sale price. Because the data is split into above grade living area and basement area, a third feature is also created, called "Total SF," to examine the relationship between total square footage and sale price

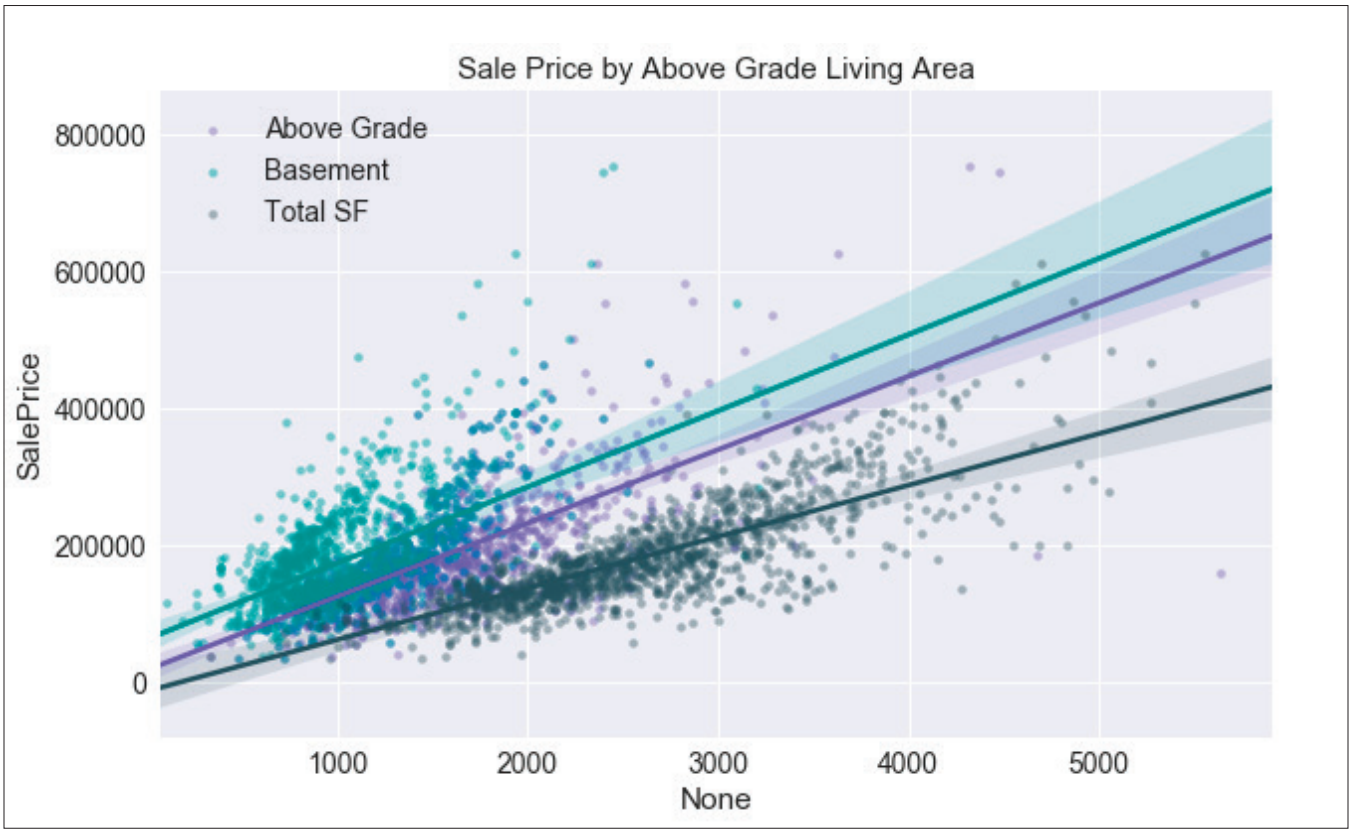


Figure 3

Figure 4

In Figure 4 above, the sale price is plotted against the overall quality rating of the home (1 through 10) in a swarm plot. There is a very noticeable relationship between quality and sale price, with the homes rated highly in quality having significantly higher prices.

By condition, homes that were rated highly tended to have very slightly higher sale prices, but not by too much. The majority of homes received a rating of 5 for condition. (Figure 5)

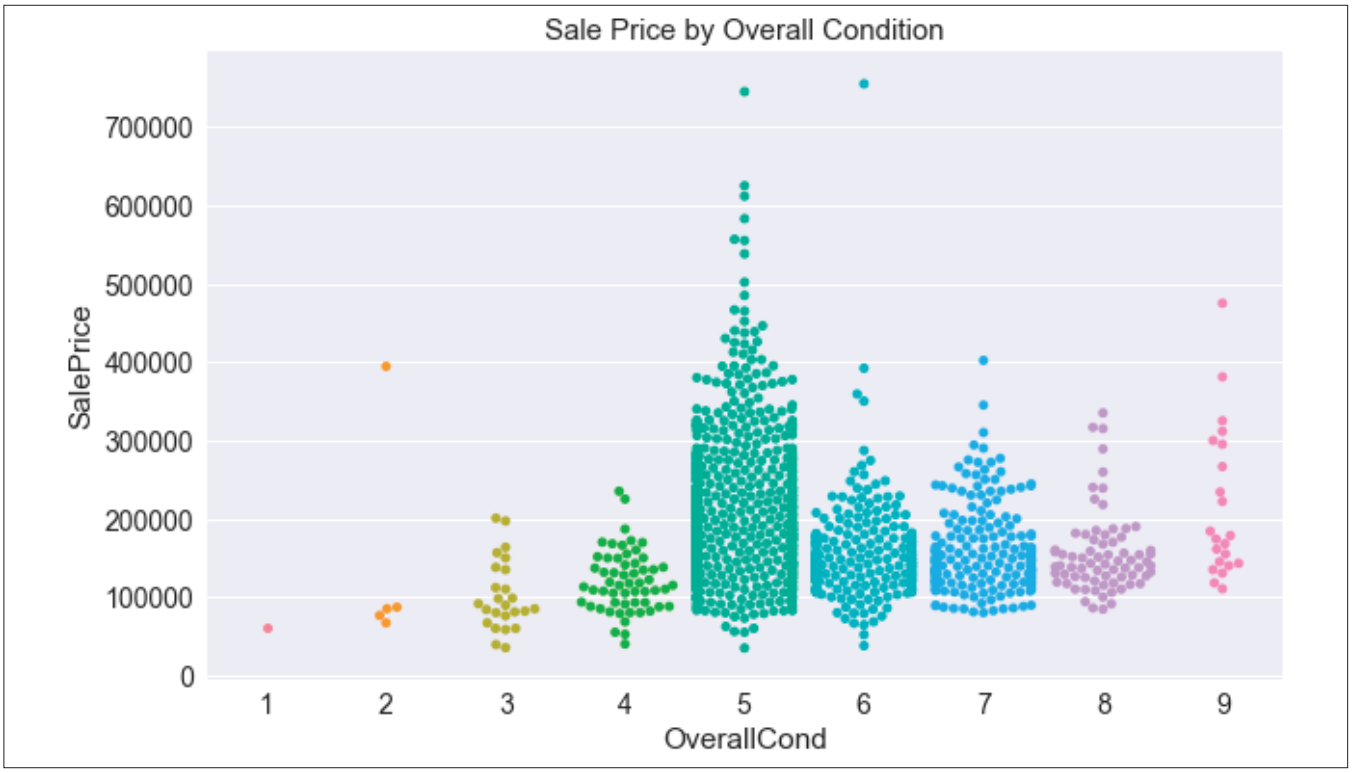


Figure 5

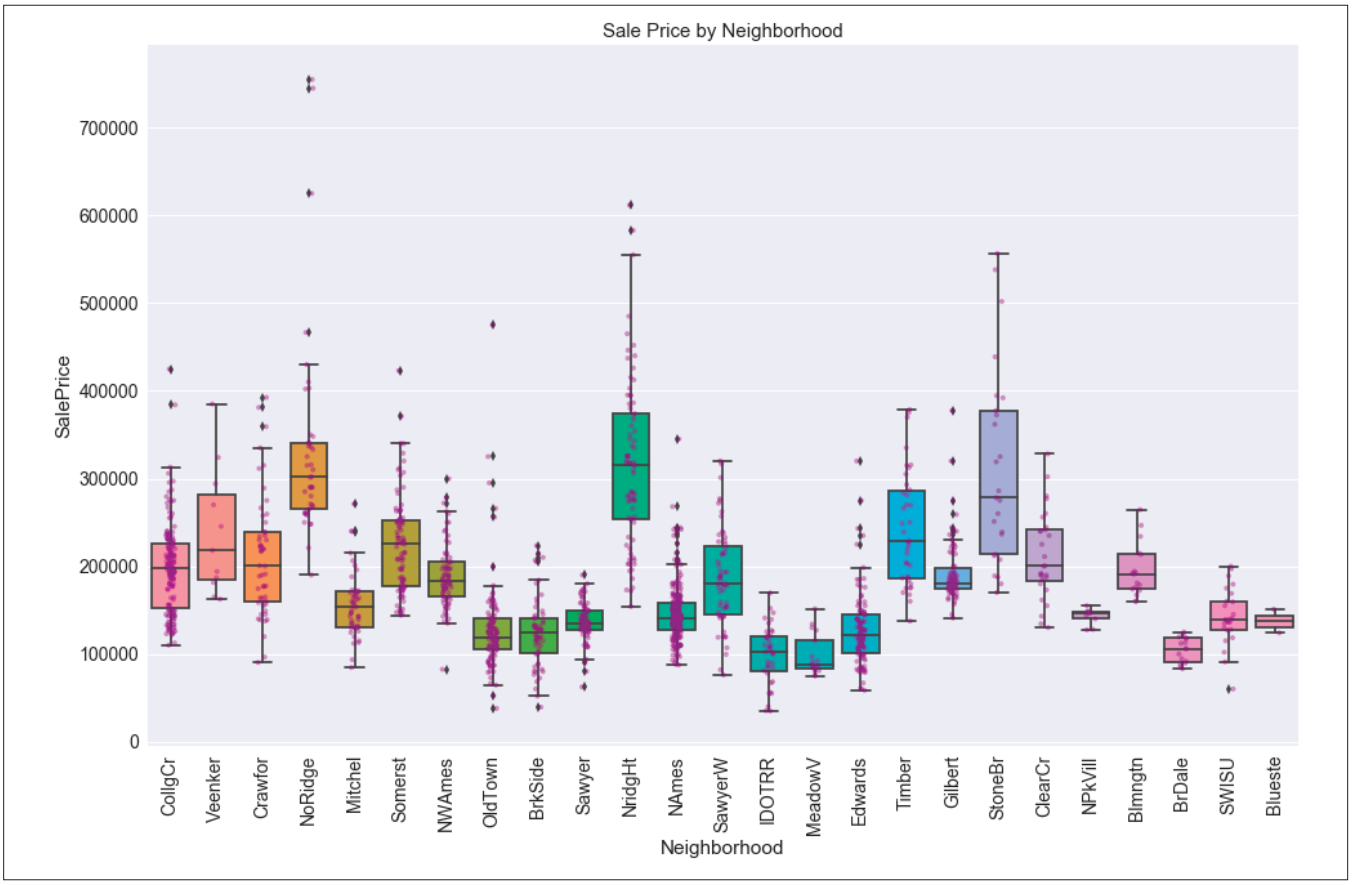


Figure 6

The boxplot above in Figure 6 depicts the spread of sale prices by neighborhood. Northridge Heights appears to have homes with the highest sale prices.

Examining the sale prices by the slope of the land shows that most homes were located on a plot with a gentle slope. The range of prices tended to be similar no matter the slope of the plot (Figure 7).

Figure 7

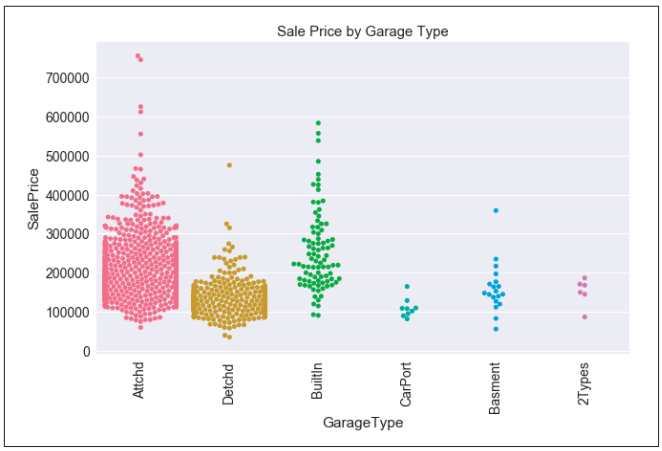


Figure 8

In Figure 8, the sale prices are plotted against their garage types. Most of the homes in the dataset had attached garages. Homes with attached garages tended to have higher home values than homes that had detached garages.

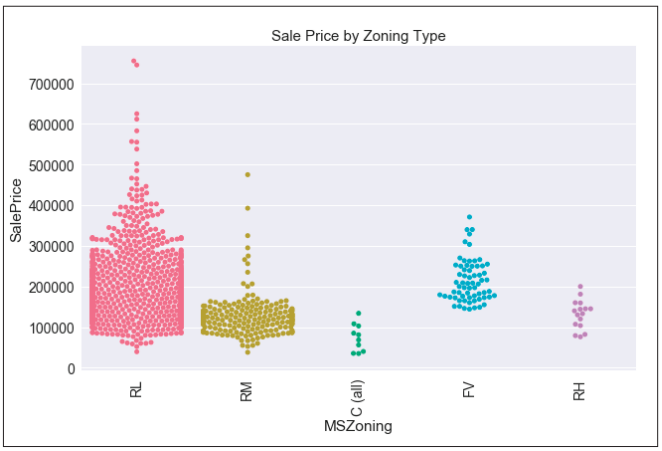


Figure 9

Finally, the above swarm plot depicts the relationship between sale price and zoning for the home. The majority of homes were located in a low-density residential zone, and those homes also tended to have the highest prices. There were only a handful of homes located in a commercial zone, and those had lower than average prices.

04

Statistical Analysis

Springboard Data Science

May 2018

04 Statistical Analysis

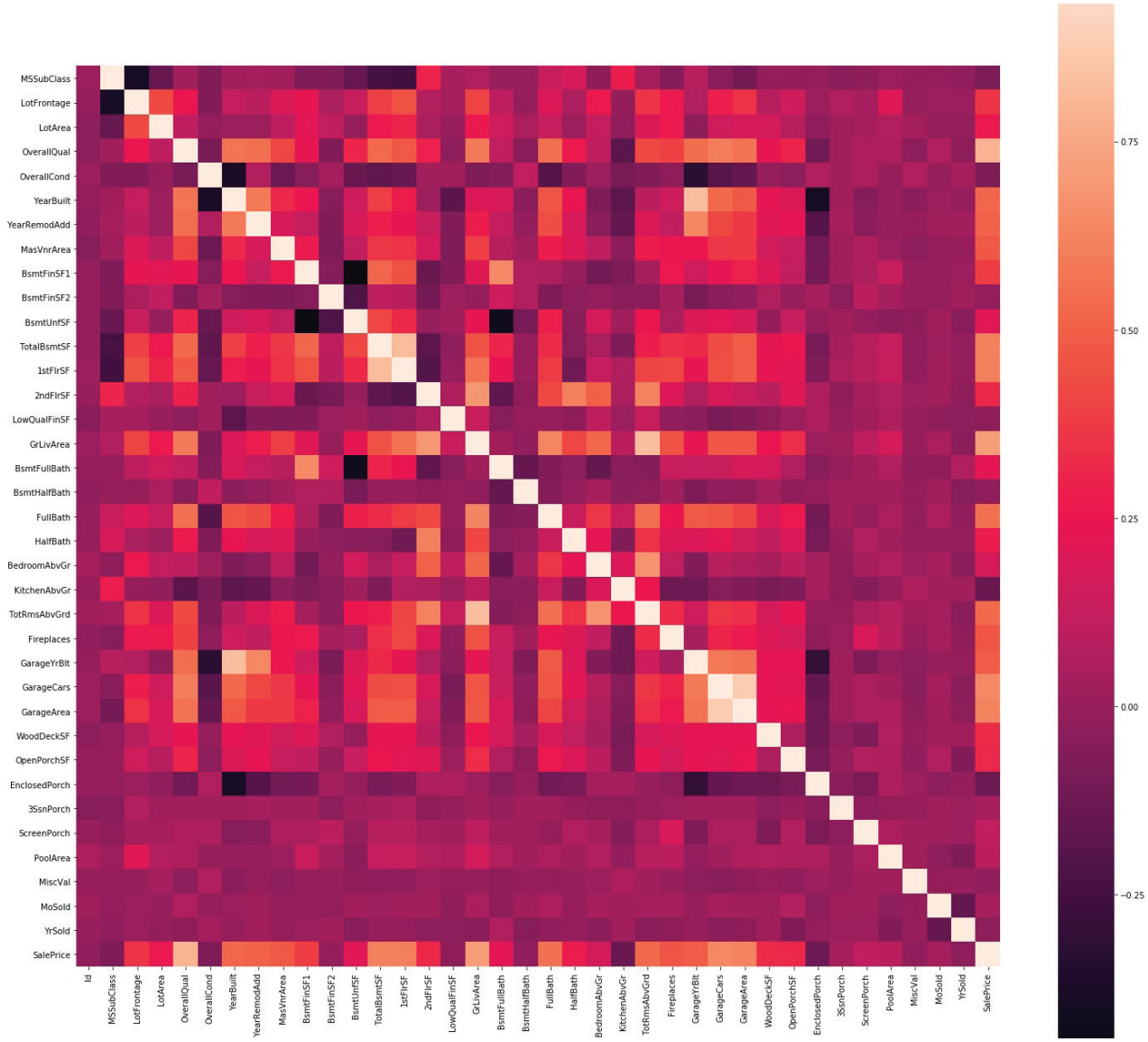


Figure 10

In this section, statistical analysis is performed to determine which variables have the greatest correlation with sale prices, as well as examine whether there is a substantial amount of collinearity within the features, which will have to be addressed before any regression models can be fitted.

04.01 Correlation Matrix

First, a heatmap of the correlation values is created to get a general sense of which variables are most highly correlated to sale prices.

In general, there seems to be a positive correlation between sale prices and most of the other numerical features, with the top five correlations listed below:

Feature Name	correlation	p-value
OverallQual	0.79	2.19E-313
GrLivArea	0.71	4.52E-223
GarageCars	0.64	2.50E-169
GarageArea	0.62	5.27E-158
TotalBsmtSF	0.61	9.48E-152

The overall quality of the home and it living area have the strongest correlations to sale price. While there are some negative correlations, the magnitude of those values is relatively small.

The entire table of correlation values can be viewed in Appendix A.

04.02 ANOVA

To analyze the relationship between sale prices with the categorical variables, an analysis of variance (ANOVA) will be performed. The ANOVA analysis will help determine whether the differences in the sale prices are significant or not for the categorical variables.

The following table shows the features with the largest 20 f-statistics from the ANOVA and their corresponding p-values.

Feature	f-value	p-value
ExterQual	443.3348	1.44E-204
KitchenQual	407.8064	3.03E-192
BsmtQual	392.9135	9.61E-186
GarageCars	351.2075	1.12E-211
OverallQual	349.0268	0.00E+00
GarageFinish	250.9625	1.20E-93
FullBath	246.067	3.80E-129
Fireplaces	146.7412	4.42E-83
MasVnrType	111.6724	4.79E-65
Foundation	100.2539	5.79E-91
CentralAir	98.30534	1.81E-22
HeatingQC	88.39446	2.67E-67
HalfBath	74.4697	1.61E-31
Neighborhood	71.78487	1.56E-225
GarageType	71.52212	1.25E-66
BsmtExposure	70.88798	1.02E-42
BsmtFinType1	67.60218	1.81E-63
TotRmsAbvGrd	56.16424	1.19E-103
SaleCondition	45.57843	7.99E-44
MSZoning	43.84028	8.82E-35

The complete table of ANOVA values can be found in the appendix.

The code used to perform this analysis can be found here: https://github.com/kellywong1314/Springboard-Data-Science/blob/master/Capstone%202%20-%20Advanced%20Regression%20to%20Predict%20Housing%20Prices/Statistical%20Inference%20-%20Advanced%20Regression.ipynb

05

Data Wrangling

Springboard Data Science

May 2018

05 Data Wrangling

In order to prepare the data for a machine learning model, all of the null values need to be imputed or dropped, categorical values need to be encoded, and the skew of the target variable has to be addressed.

First, the dataset is examined to see which features contain null values:

Feature	Percent Null
PoolQC	99.52
MiscFeature	96.30
Alley	93.76
Fence	80.75
FireplaceQu	47.26
LotFrontage	17.73
GarageYrBlt	5.54
GarageType	5.54
GarageFinish	5.54
GarageQual	5.54
GarageCond	5.54
BsmtFinType2	2.60
BsmtExposure	2.60
BsmtFinType1	2.53
BsmtCond	2.53
BsmtQual	2.53
MasVnrArea	0.55
MasVnrType	0.55
Electrical	0.07

Imputing Null Values

The feature for PoolQC has the highest number of nulls. As there are relatively few features with null values in this dataset, the majority of them are able to be imputed.

For many of the categorical features, the null values indicate that the home simply does not have that feature. For instance, the feature for PoolQC is blank if the home does not have a pool. Those values were filled with a category called "None" and later imputed.

The numerical features with null values were dealt with individually, but most were imputed using the mean value within each column.

Finally, for categorical features that were truly missing data, the null values were imputed with the mode of the feature. For instance, almost every home had "Public" electricity, so the Electrical feature was filled with that same value.

For a detailed explanation of how every feature with null values was addressed, refer to the jupyter notebook for data wrangling: https://github.com/kellywong1314/Springboard-Data-Science/blob/master/Capstone%202%20-%20Advanced%20Regression%20to%20Predict%20Housing%20Prices/Data%20Wrangling%20For%20Regression.ipynb

Encoding Categorical Variables

The categorical variables within the dataset needed to be encoded before a machine learning model could be run. While it would have been quick and simple to immediately use pandas' get_dummies function to encode all of the variables, the dataset would have lost the relationships between the categorical variables which were ordinal.

For instance, features that are condition-based and have categories such as "Good," "Fair," and "Poor" are ordinal, and preserving their relative distance from each other may increase the model's accuracy. To address this, features with ordinal values were mapped with a scale of numbers, as shown below.

After this step, the remaining categorical variables were encoded using get_dummies.

```
# create dictionary of values for ordinal categorical variables

cond_nums = {'LotShape': {'IR3':0, 'IR2':1, 'IR1':2, 'Reg':3},
             'LandSlope': {'Gtl':0, 'Mod':1, 'Sev':2},
             'ExterQual': {'Po':1, 'Fa':2, 'TA':3, 'Gd':4, 'Ex':5},
             'ExterCond': {'Po':1, 'Fa':2, 'TA':3, 'Gd':4, 'Ex':5},
             'BsmtQual': {'None':0, 'Po':1, 'Fa':2, 'TA':3, 'Gd':4, 'Ex':5},
             'BsmtCond': {'None':0, 'Po':1, 'Fa':2, 'TA':3, 'Gd':4, 'Ex':5},
             'BsmtExposure': {'None':0, 'No':1, 'Mn':2, 'Av':3, 'Gd':4},
             'BsmtFinType1': {'None':0, 'Unf':1, 'LwQ':2, 'Rec':3, 'BLQ':4, 'ALQ':5, 'GLQ':6},
             'BsmtFinType2': {'None':0, 'Unf':1, 'LwQ':2, 'Rec':3, 'BLQ':4, 'ALQ':5, 'GLQ':6},
             'HeatingQC': {'Po':1, 'Fa':2, 'TA':3, 'Gd':4, 'Ex':5},
             'CentralAir': {'N':0, 'Y':1},
             'KitchenQual': {'Po':1, 'Fa':2, 'TA':3, 'Gd':4, 'Ex':5},
             'Functional': {'Sal':0, 'Sev':1, 'Maj2':2, 'Maj1':3, 'Mod':4, 'Min2':5, 'Min1':6, 'Typ':7},
             'FireplaceQu': {'None':0, 'Po':1, 'Fa':2, 'TA':3, 'Gd':4, 'Ex':5},
             'GarageFinish': {'None':0, 'Unf':1, 'RFn':2, 'Fin':3},
             'GarageQual': {'None':0, 'Po':1, 'Fa':2, 'TA':3, 'Gd':4, 'Ex':5},
             'GarageCond': {'None':0, 'Po':1, 'Fa':2, 'TA':3, 'Gd':4, 'Ex':5},
             'PavedDrive': {'N':0, 'P':1, 'Y':2},
             'PoolQC': {'None':0, 'Fa':2, 'TA':3, 'Gd':4, 'Ex':5}}
```

```
# replace ordinal categorical variables using above dictionary

df_imputed.replace(cond_nums, inplace=True)
```

Scaling the Numerical Values

To prevent the features with large values from being weighted more heavily within the model, RobustScaler was used to scale the numerical values.

After scaling, the features are all brought into the same scale.

Logging the Target Feature

Finally, because the target variable, Sale Price, has a significant right skew, the log of every sale price was computed and used to train the model. Log transformations can reduce the skew of distributions.

Dropping Outliers

Although the scaler should create features that are robust to outliers, there were very significant outliers noticed during the EDA portion of the project which had very low prices considering the square footage of the homes. These were dropped from the dataset

06

Machine Learning

Springboard Data Science

May 2018

06 Machine Learning

This project included seven regression models and an analysis of which model performed best in predicting home sale prices in Ames, IA. Data was wrangled in the previous section, which included imputing null values in the features, encoding the categorical variables, scaling all of the features, and taking the log of the target variable.The models include five variations of linear regression and two ensemble methods.

The Kaggle competition will be scored based on the Root Mean Squared Logarithmic Error (rmsle), so the models will be optimized for this value. However, the Root Mean Squared Error was still be computed for every model.

Linear Regression with PCA

This first model used principal component analysis (PCA) to generate new components to be used as features in a linear regression. As a PCA generates new features, multicollinearity will not be a concern. Therefore, the PCA was run with all of the available features in the dataset.

First, the PCA was run and the number of features plotted against their explained variance values to determine a suitable number of components to fit to the PCA, as shown in Figure 11.

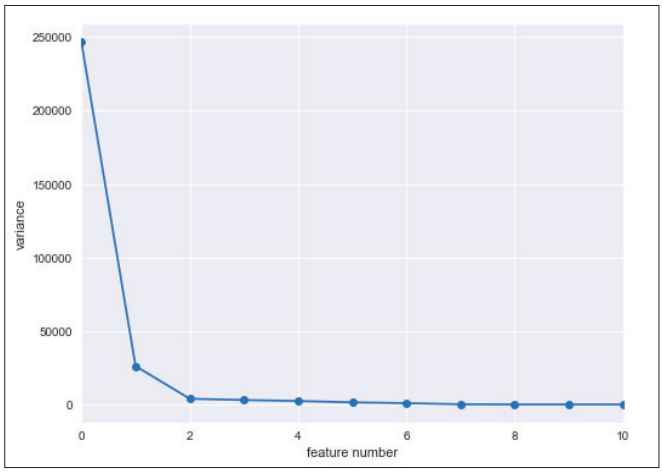


Figure 11

An elbow point can be observed at 2 components, so the PCA was performed with 2 components and fit to the features from the dataset.

The model's resulting RMSLE and RMSE were:

Linear Regression with PCA	
RMSLE	0.03087
RMSE	0.40118

Out-of-Box Linear Regression

The next model was a simple out-of-box linear regression performed with all of the available features. As there was multicollinearity observed within the dataset, this model was to serve as the benchmark for the other linear regression models, which were run after removing features with high variance inflation factors (VIFs).

This model's resuting RMSLE and RMSE were:

Out-of-box Linear Regression	
RMSLE	0.00932
RMSE	0.11961

Linear Regression with VIF Analysis

To address the issue of features that are both correlated with the target variable and each other, VIF values were computed and features that had high VIFs were dropped from the dataset.

The original dataset for the machine learning model had 229 columns, but after dropping features that had a VIF of 5 or higher, the dataset had 162 features remaining.

After running the linear regression model on this new set of features, the resulting RMSLE and RMSE values were:

Linear Regression with VIF	
RMSLE	0.01095
RMSE	0.14075

Ridge Regression

The next model was a ridge regression model, which uses L2 regularization, meaning that a penalty of the squares of the linear coefficients is added to the loss function.

Tuning the hyperparameters showed that the model seems to perform best when alpha=10.

The model's resuting RMSLE and RMSE were:

Ridge Regression	
RMSLE	0.01024
RMSE	0.13178

This model performed very closely to the linear regression model.

LASSO Regression

The LASSO model adds a penalty of the absolute value of the coefficients to the loss function, increasing sparsity in a model, which can intrinsically perform feature selection.

The model performed best with an alpha value of 0.1, and the resulting RMSLE and RMSE were:

LASSO Regression	
RMSLE	0.01904
RMSE	0.24653

This model did not perform as well as the ridge regression model or the linear regression models,

possibly because the dataset has already had feature selection performed through VIF analysis and additional sparsity was not needed.

Random Forest

Next, a random forest regressor was created and its performance evaluated. There are more hyperparameters, so the tuning step would take more time than it did for linear regression models if every hyperparameter was being tuned. Therefore, RandomizedSearchCV was used instead of GridSearchCV.

A total of 250 iterations were tested over five folds, totalling 1,250 fits. The best hyperparameters found during the tuning step were:

- 'n_estimators': 1200,
- 'min_samples_split': 5,
- 'min_samples_leaf': 1,
- 'max_features': 'sqrt',
- 'max_depth': 25,
- 'bootstrap': False

The model's resuting RMSLE and RMSE were:

Random Forest	
RMSLE	0.01130
RMSE	0.14548

XGBoost

Finally, an xgboost model was fit and its performance analyzed. Similar to the random forest, tuning the numerous hyperparameters takes quite a bit of time and computing power, so RandomizedSearchCV was used instead of GridSearchCV.

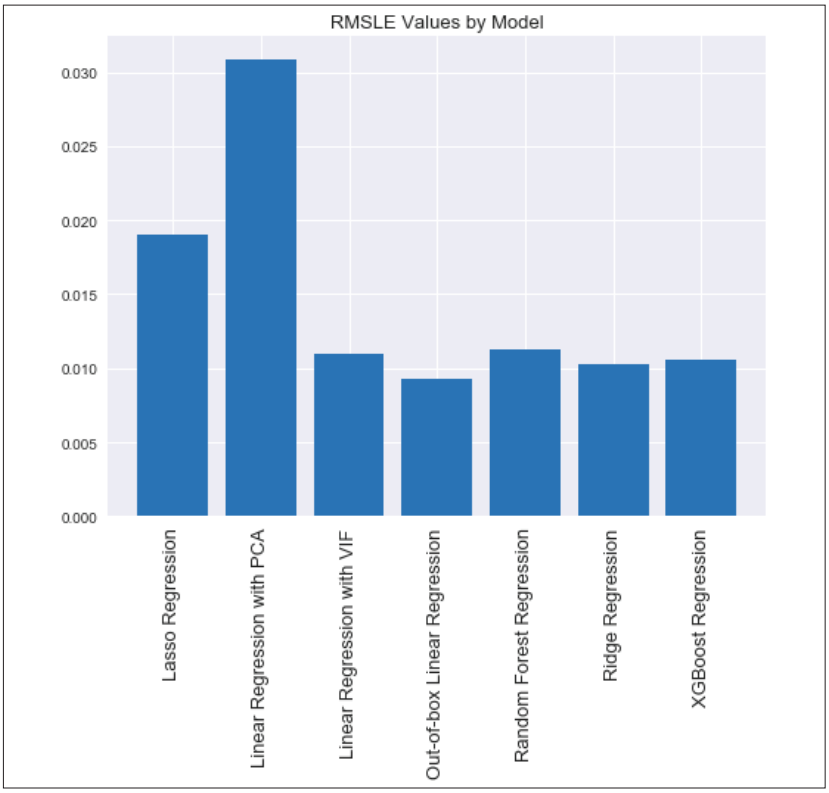
RandomizedSearchCV was performed with 1,000 iterations over 5 folds, totalling 5,000 fits. The best hyperparameters found were:

- 'subsample': 0.7
- 'reg_lambda': 1.0
- 'reg_alpha': 0.1
- 'min_child_weight': 3
- 'max_depth': 15
- 'learning_rate': 0.1
- 'gamma': 0.05
- 'colsample_bytree': 0.8

The model's resuting RMSLE and RMSE were:

XGBoost	
RMSLE	0.01058
RMSE	0.13604

The final RMSLE values for every model are shown below in Figure 12. The linear regression model performed best, though most of the models performed similarly. The worst performing models were the linear regression performed with a PCA and the LASSO regression.



07 Conclusion

Springboard Data Science

May 2018

07 Conclusion

The majority of the models scored similarly, with the linear regression just performing slightly better than ridge regression. The worst-performing model was the linear regression fit with PCA. While tuning the linear models was relatively simple and took very little time, the ensemble methods took significantly longer.

As the xgboost model and the random forest model were tuned with RandomizedSearchCV, their performance is likely to improve slightly more if more tuning is done. Still, the best performing model out of the seven was a simple linear regression.

The top five models were used to generate submissions into the Kaggle competition, and there RMSLE values are shown in the following table.

Model	RMSLE on Kaggle
Linear Regression	0.14552
Linear Regression after VIF	0.15965
Ridge Regression	0.14646
Random Forest	0.16447
XGBoostt	0.14822

The best scoring model resulted in an RMSLE of 0.14552, and is in the top 45% of all submissions.

As the best scoring model was a linear regression, future models may benefit from more focus on techniques that may improve linear model performance. Although two outliers were removed from the training data set, further focusing on removing influential points or leverage points may significantly improve the linear regression model's performance. These methods should be pursued in future models.

In addition, many of the top performing models in the Kaggle competition seem to have used blended models. Another possible path of exploration for further improving model performance would be to use a blended model approach and examine which combinations result in improvement.

The entire github repository and files can be found here: https://github.com/kellywong1314/Springboard-Data-Science/tree/master/Capstone%202%20-%20Advanced%20Regression%20to%20Predict%20Housing%20Prices

Appendices

Springboard Data Science

May 2018

Appendix A: Correlation Values

The following table contains all of the correlation values betweeen the numerical features and the home sale prices, along with their p-values.

Feature	Pearson correlation	p-value
OverallQual	0.791	2.19E-313
GrLivArea	0.709	4.52E-223
GarageCars	0.640	2.50E-169
GarageArea	0.623	5.27E-158
TotalBsmtSF	0.614	9.48E-152
1stFlrSF	0.606	5.39E-147
FullBath	0.561	1.24E-121
TotRmsAbvGrd	0.534	2.77E-108
YearBuilt	0.523	2.99E-103
YearRemodAdd	0.507	3.16E-96
Fireplaces	0.467	6.14E-80
BsmtFinSF1	0.386	3.39E-53
WoodDeckSF	0.324	3.97E-37
2ndFlrSF	0.319	5.76E-36
OpenPorchSF	0.316	3.49E-35
HalfBath	0.284	1.65E-28
LotArea	0.264	1.12E-24
BsmtFullBath	0.227	1.55E-18
BsmtUnfSF	0.214	1.18E-16
BedroomAbvGr	0.168	9.93E-11
ScreenPorch	0.111	1.97E-05
PoolArea	0.092	4.07E-04
MoSold	0.046	7.61E-02
3SsnPorch	0.045	8.86E-02
BsmtFinSF2	-0.011	6.64E-01
BsmtHalfBath	-0.017	5.20E-01
MiscVal	-0.021	4.18E-01
LowQualFinSF	-0.026	3.28E-01
YrSold	-0.029	2.69E-01
OverallCond	-0.078	2.91E-03
MSSubClass	-0.084	1.27E-03
EnclosedPorch	-0.129	8.26E-07
KitchenAbvGr	-0.136	1.86E-07
GarageYrBlt	0	1.00E+00
LotFrontage	0	1.00E+00
MasVnrArea	0	1.00E+00

Appendix B: ANOVA Table

The following table contains all of the f-values and their corresponding p-values for an ANOVA analysis of every categorical feature against the home sale price.

Feature	f-value	p-value
ExterQual	443.33	1.44E-204
KitchenQual	407.81	3.03E-192
BsmtQual	392.91	9.61E-186
GarageCars	351.21	1.12E-211
OverallQual	349.03	0.00E+00
GarageFinish	250.96	1.20E-93
FullBath	246.07	3.80E-129
Fireplaces	146.74	4.42E-83
MasVnrType	111.67	4.79E-65
Foundation	100.25	5.79E-91
CentralAir	98.31	1.81E-22
HeatingQC	88.39	2.67E-67
HalfBath	74.47	1.61E-31
Neighborhood	71.78	1.56E-225
GarageType	71.52	1.25E-66
BsmtExposure	70.89	1.02E-42
BsmtFinType1	67.60	1.81E-63
TotRmsAbvGrd	56.16	1.19E-103
SaleCondition	45.58	7.99E-44
MSZoning	43.84	8.82E-35
PavedDrive	42.02	1.80E-18
LotShape	40.13	6.45E-25
Alley	35.56	4.90E-08
MSSubClass	33.73	8.66E-79
SaleType	28.86	5.04E-42
BsmtFullBath	27.45	2.88E-17
OverallCond	26.00	7.96E-38
FireplaceQu	24.40	5.02E-19
Electrical	23.07	1.66E-18
HouseStyle	19.60	3.38E-25
Exterior1st	18.61	2.59E-43
RoofStyle	17.81	3.65E-17
Exterior2nd	17.50	4.84E-43
BsmtCond	14.03	5.14E-09
BldgType	13.01	2.06E-10
BedroomAbvGr	12.93	3.30E-16
LandContour	12.85	2.74E-08
YearRemodAdd	10.48	2.35E-76
KitchenAbvGr	9.84	2.01E-06
GarageQual	9.57	1.24E-07

Feature	f-value	p-value
GarageCond	9.54	1.31E-07
YearBuilt	9.53	9.13E-107
ExterCond	8.80	5.11E-07
PoolArea	7.85	2.34E-09
LotConfig	7.81	3.16E-06
GarageArea	7.75	1.07E-158
GarageYrBlt	7.53	9.06E-73
RoofMatl	6.73	7.23E-08
Condition1	6.12	8.90E-08
LotFrontage	6.06	1.92E-58
GrLivArea	5.29	1.16E-90
Fence	4.95	2.31E-03
TotalBsmtSF	4.66	1.27E-88
Heating	4.26	7.53E-04
LotArea	4.10	6.59E-50
Functional	4.06	4.84E-04
OpenPorchSF	3.88	5.06E-49
MasVnrArea	3.51	2.02E-54
1stFlrSF	3.46	5.61E-59
WoodDeckSF	2.89	2.91E-35
BsmtFinType2	2.70	1.94E-02
Street	2.46	1.17E-01
BsmtFinSF1	2.20	1.48E-26
MiscFeature	2.16	1.05E-01
Condition2	2.07	4.34E-02
2ndFlrSF	2.05	2.22E-20
LandSlope	1.96	1.41E-01
PoolQC	1.63	3.04E-01
BsmtUnfSF	1.51	1.64E-08
ScreenPorch	1.32	3.68E-02
LowQualFinSF	1.18	2.56E-01
EnclosedPorch	1.17	1.14E-01
3SsnPorch	1.04	4.10E-01
MiscVal	0.99	4.74E-01
MoSold	0.96	4.83E-01
YrSold	0.65	6.30E-01
BsmtFinSF2	0.54	1.00E+00
Utilities	0.30	5.85E-01
BsmtHalfBath	0.22	8.01E-01