Ames Housing Prediction Model

Mona Elkholy

Chris Kennedy

Abdelrahman Gouda

Yuzhi Sheng

Jahanzaib Talpur

Tyson Van Patten

OR-568-001

Dr. Xu

George Mason University

Ames Housing Prediction Model

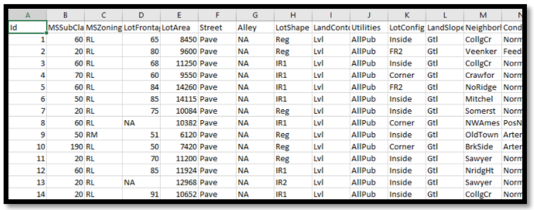
Buying a house is possibly one of the most significant decisions in a person's life. When a person enters into this process, they generally rely on a real estate professional to help assess the value of property. There are many different factors which go into the cost of a property. A savvy real estate professional is always attempting to see if a property is overpriced or underpriced. Any real estate professional who has a predictive model to help assess if a house is overpriced or underpriced would likely gain a competitive advantage over realtors that do not use a model or a model that has less predictive power.

The goal of the project will be to build a predictive model which can predict the potential sale price of a house based upon the provided information. The model needs to have strong predictive power, run quickly and efficiently, and be easy to interpret.

Data Sources, Data Preprocessing, and Software

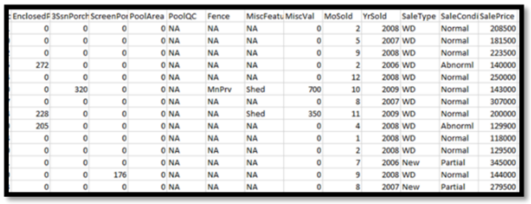
The Ames data set provided by Kaggle contains 2,920 observations (Kaggle, 2018). Kaggle has separated these observations into two separate datasets, a train set and a test set. Each set includes 1,460 observations. In the training dataset provided by Kaggle, there are 80 predictors and one response. The response variable is only present in the training data set and is the SalePrice.

The data is a collection of housing descriptions and sales prices for houses sold in Ames, Iowa from 2006 to 2010. From the 80 variables, 46 are categorical while 34 are numeric. There are 20 continuous variables which mostly describe the dimensions for the house. The remaining 14 numeric variables are discrete and are generally used to describe the number of rooms within the home, such as two bedrooms. The categorical variables are split evenly at 23 nominal and 23 ordinal. The categorical variable classes sizes range from 2 to 28 (De Cock, 2011). The data describes the house and includes variables such as the type of dwelling, physical location by neighbourhood, build date, last remodel data, exterior materials of the house, number of bedrooms, number of bathrooms, square footage of the home, and size of the lot. An example of the data can be seen in figure 1 and figure 2.



A	B	C	D	E	F	G	H	I	J	K	L	M	N
Id	MSSubCla	MSZoning	LotFronta	LotArea	Street	Alley	LotShape	LandConto	Utilities	LotConfig	LandSlope	Neighborl	Cond
1	60	RL	65	8450	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	Norm
2	20	RL	80	9600	Pave	NA	Reg	Lvl	AllPub	FR2	Gtl	Veenker	Feed
3	60	RL	68	11250	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	Norm
4	70	RL	60	9550	Pave	NA	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	Norm
5	60	RL	84	14260	Pave	NA	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	Norm
6	50	RL	85	14115	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	Mitchel	Norm
7	20	RL	75	10084	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	Somerst	Norm
8	60	RL	NA	10382	Pave	NA	IR1	Lvl	AllPub	Corner	Gtl	NWAmes	PosN
9	50	RM	51	6120	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	OldTown	Arter
10	190	RL	50	7420	Pave	NA	Reg	Lvl	AllPub	Corner	Gtl	BrkSide	Arter
11	20	RL	70	11200	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	Sawyer	Norm
12	60	RL	85	11924	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	NridgHt	Norm
13	20	RL	NA	12968	Pave	NA	IR2	Lvl	AllPub	Inside	Gtl	Sawyer	Norm
14	20	RL	91	10652	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	Norm

Figure 1. Example of Data from Ames Dataset



EnclosedF	3SsnPorch	ScreenPor	PoolArea	PoolQC	Fence	MiscFeatu	MiscVal	MoSold	YrSold	SaleType	SaleCondi	SalePrice
0	0	0	0	NA	NA	NA	0	2	2008	WD	Normal	208500
0	0	0	0	NA	NA	NA	0	5	2007	WD	Normal	181500
0	0	0	0	NA	NA	NA	0	9	2008	WD	Normal	223500
272	0	0	0	NA	NA	NA	0	2	2006	WD	Abnorml	140000
0	0	0	0	NA	NA	NA	0	12	2008	WD	Normal	250000
0	320	0	0	NA	MnPrv	Shed	700	10	2009	WD	Normal	143000
0	0	0	0	NA	NA	NA	0	8	2007	WD	Normal	307000
228	0	0	0	NA	NA	Shed	350	11	2009	WD	Normal	200000
205	0	0	0	NA	NA	NA	0	4	2008	WD	Abnorml	129900
0	0	0	0	NA	NA	NA	0	1	2008	WD	Normal	118000
0	0	0	0	NA	NA	NA	0	2	2008	WD	Normal	129500
0	0	0	0	NA	NA	NA	0	7	2006	New	Partial	345000
0	0	176	0	NA	NA	NA	0	9	2008	WD	Normal	144000
0	0	0	0	NA	NA	NA	0	8	2007	New	Partial	279500

Figure 2. Example of Data from Ames Dataset

Once the dataset was imported into R, an exploratory data analysis was conducted. The variables in the dataset are used to describe the properties of an individual house. Within the dataset, there are some columns which have a large amount of NA values. Within most datasets when dealing with a large amount of NA values, the standard practice is to either remove the variable or impute a value into the dataset. However, in the Ames dataset the NA generally represents the element the variable being described does not exist for that observation. For example, the variable Fence has five classes which are Good Privacy (GdPrv), Minimum Privacy (MnPrv), Good Wood (GdWo), Minimum Wood/Wire (MnWw), and No Fence (NA). Figure 3 is a bar chart demonstrating the number of each class in the dataset. In this example, if the NA value was imputed with either most common value the house with no fence would have been labelled with a minimum privacy fence which the observation does not have. If the Fence variable was just removed from the

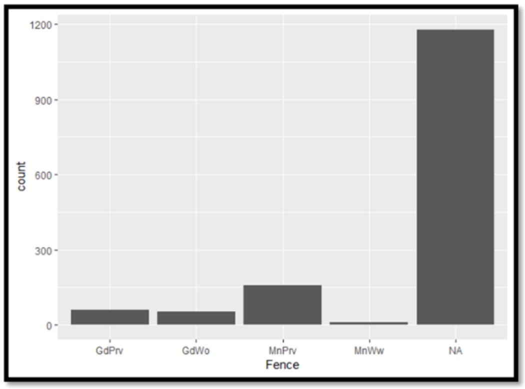


Figure 3. Class breakdown of Fence within the dataset

dataset, the model could potentially be missing an important parameter. By examining the dataset variables and their meaning within the dataset, it was concluded that most variables should be removed or imputed. In some cases, the NA value was replaced with a None or zero to represent the absence of the item being described.

The software used to conduct the analysis was R version 3.5.0 (R Core Team, 2018). R Studio version 1.1447 was used as the R integrated development environment (IDE) (RStudio, 2018). The following packages were used to process, transform, or model the data for the ordinary least square regression, partial least squares and principal component regression were rsq (Zhang, 2018), plyr (Wickham, 2011), tidyverse (Wickham, tidyverse: Easily Install and Load the 'Tidyverse', 2017), caret (Kuhn, 2018), ggplot2 (Wickham, ggplot2: Elegant Graphics for Data Analysis, 2016), earth (Milborrow, 2018), pls (Mevik, 2018), gridExtra (Auguie, 2017), and e1071 (Meyer, 2018). The following packages were used for the Ridge and Lasso regressions elasticnet (Zou, 2018), pls (Mevik, 2018), mlbench (Leisch & Dimitriadou, 2012), caret (Kuhn, 2018), corrplot (Wei & Simko, 2017), AppliedPredictiveModeling (Kuhn, AppliedPredictiveModeling: Functions and Data Sets for 'Applied Predictive Modeling', 2018), and e1071 (Meyer, 2018). The robust linear regression was completed using the caret (Kuhn, caret: Classification and Regression Training, 2018) and fastDummies (Kaplan, 2018) packages for R. The following packages were used for the random forest model dplyr (Wickham, dplyr: A Grammar of Data Manipulation, 2017), GGally (Schloerke, 2018), caret (Kuhn, caret: Classification and Regression Training, 2018), ggplot2 (Wickham, ggplot2: Elegant Graphics for Data Analysis, 2016), and randomForest (Liaw, 2002). To build the support vector machine model the following packages were used tidyverse (Wickham, tidyverse: Easily Install and Load the 'Tidyverse', 2017), corrplot (Wei & Simko, 2017), grid (R Core Team, 2018), AppliedPredictiveModeling (Kuhn, AppliedPredictiveModeling: Functions and Data Sets for 'Applied Predictive Modeling', 2018), caret, e1071 (Meyer, 2018), and lattice (Sarkar, 2008).

Ordinary Least Square Regression

The first model generated with the dataset was the ordinary least square (OLS) regression. This model was chosen due to its lack of complexity and ease of interpretation. The first step was to prepare the data for the model. The data was examined for variables which had a near zero variance. To accomplish this, the NearZeroVar function was used. The NearZeroVar function identified the Street, LandCountour, Utilities, LotConfig, Condition2, RoofMatl, BsmtCond, BsmtFinType2, BsmtFinSF2, Heating, LowQaulFinSF, KitchenAbfGr, GarageQual, GargageCond, OpenProchSF, EnclosedPorch, X3SsnPorch, ScreenPorch, PoolArea, and MiscVal. Predictors with over 90% NA or null features where removed from the dataset. This included PoolQC, MiscFeatrues, Alley and Fence. For the remaining categorical variables missing or NA values were imputed to None. For the numerical values, the missing values or NAs were imputed as a zero.

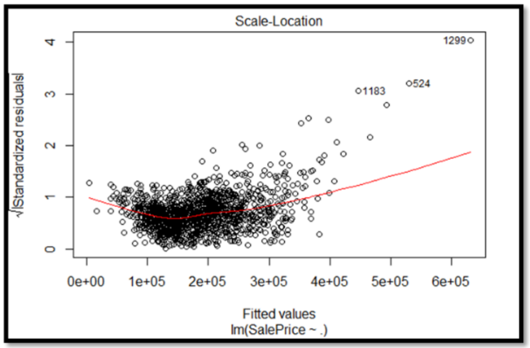


Figure 4. Plot of OLS model with all predictors

Using the processed dataset, a standard OLS linear model was constructed in R using the lm function. This model returned a root mean squared error (RMSE) of 36025.38, R^2 of .800896, RMSE SD of 13888.9, and a R^2 SD of .1218249. This model can be visualized in figure 4. This model identified the MSSubClass, LotArea, OverallQual, OverallCond, YearBuilt, MasVnrArea, BasmtFinSF1, X1stFlrSF, X2ndFlrSF, BedroomabvGr, KitchenAbvGr, and TotRmsAnvGr as predictors with high levels of impact on the model.

A plot was generated showing all selected predictors correlation with the response. From the graph, these predictors were chosen to produce a filtered OLS regression Full Bath, Year Built, year RemodAdd, TotalBsmtSF, X1stFirSF, GarageCars, GarageArea, OverallQual, GrLivArea, TotRmsAbvGrd with the response of SalePrice. This model produced RMSE of 39051.74, R^2 of .7778653, and RMSE SD of 131.27.4 and a R^2 SD of .1285138. From the filtered predictor set the YearRemodAdd, TotalBsmtSF, GarageCars, OverallQual, and GrLivArea were identified as significant predictors for the model.

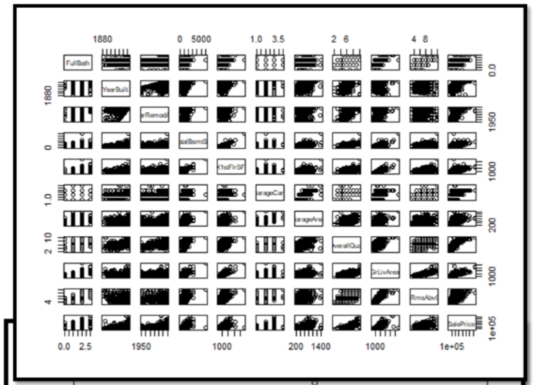


Figure 5 Predictors Correlation with Response

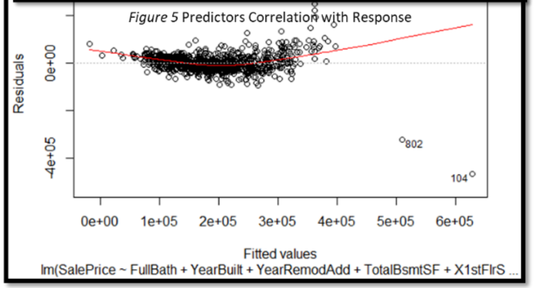


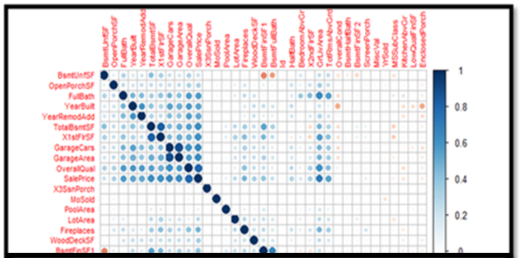
Figure 6. Plot of OLS model with filtered predictors

While the filtered OLS model was a simpler model with reduced predictors it lost predictive power. The RMSE of the OLS model with all predictors was 36025.38 compared to an RMSE of 39051.75 for the filtered predictors. While the OLS model with the all predictors is a more complicated model than the filtered OLS model it is still a reasonably simple model and overall performs well for the dataset.

Partial Least Squares and Principal Component Regression

When running an OLS model one issue that can affect the predictive power of the model is the correlation of predictors within the dataset. As demonstrated in Figure 7 there is are several predictors which are correlated. To address this multicollinearity within the dataset, a partial least squares (PLS) model will be conducted. This dataset was preprocessing in the same manner as the OLS model.

As part of the data preprocessing it is essential to center and scale the predictors. Once the data was processed, the PLS model was fitted using 10-fold cross-validation. The results of the components on the RMSE of the model can be seen in Figure 8. With two components there is a significant drop in RMSE; however, looking at all 15 components it shows another decline in RMSE at seven components. The components 2, 7, and 12 were further inspected. 2 Components had an RMSE of 38139.75, R^2 of .786481, RMSE SD of 14353.57, and a R^2 SD of .1324041. For 7 Components there was an RMSE of 38232.37, R^2 of .785965, and RMSE SD of 15040.75 and a R^2 SD of



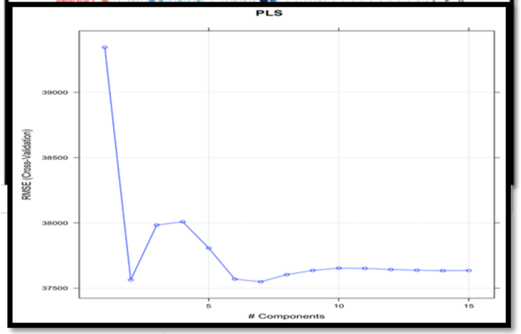


Figure 8 PLS Components

.1348921. For 12 components there was an RMSE of 38357.54, R^2 of .784778, and RMSE SD of 15187.74 and a R^2 SD of .1360046. While the ncomp of 2 resulted in a slightly lower RMSE than then the ncomp of 7 the R^2 was almost the same. In this case, the model appears to run best with a ncomp of 7.

The dataset was then fitted to a principal component regression model. The data was preprocessed the same as for PLS and then centered and scaled. The results of the PCR can be viewed in figure 9. Figure 9 shows the RMSE slowly drop from 2 components to 7 components and continue to decline untill 12 components. 2 Components had an RMSE of 49777.54, R^2 of .6544161, an RMSE SD of 13650.12, and a R^2 SD of .1424364. The seven components had an RMSE of 49189.32, R^2 of .6726159, an RMSE SD of 16906.00, and a R^2 SD of .1530155. The 12 components had an RMSE of 46355.20, R^2 of .6992587, an RMSE SD of 15711.34, and a R^2 SD of .1414779. The best number of components to use in the PCR model is 12.

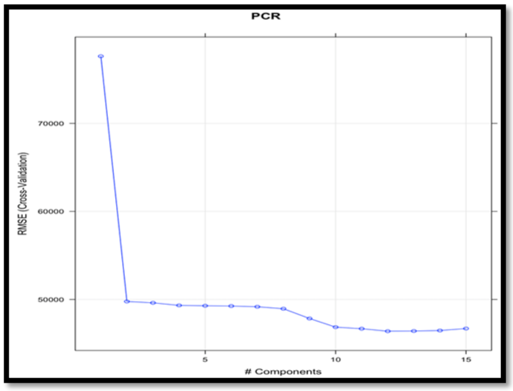


Figure 9 PCR Components

The best model to use for this dataset between the PLS and PCR is the PLS model with seven components. The PLS model with seven components had an RMSE of 38232.37, and the best PCR model with 12 components has an RMSE of 46355.20. This is a significant reduction in RMSE. While the PLS with seven components model is the best fit between the PLS and PCR the OLS model with all the variables still performs better than a PLS model. The OLS model with all variables RMSE is 36025.36 compared to the RMSE of PLS with seven components at 38232.37.

Top 5 the most important predictors for sale price determined by PLS and PCR are OveralQual, GrLivArea, TotalBsmtSF, GarageCars and GarageArea. Therefore, overall quality, above ground living area square feet, total basement square feet, size of garage capacity and garage square feet have a significant impact of the sale price.

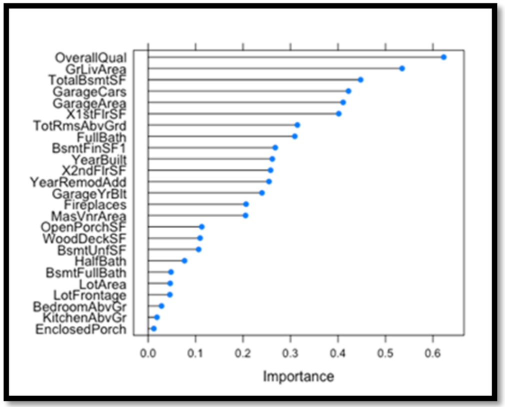


Figure 10 Top 5 Most Important Predictors

Ridge and Lasso Regression

One issue that might arise from OLS, PCR, or PLS models is the tendency to overfit a model. To avoid overfitting, a model Ridge or Lasso regression methods can be used. To see if the housing dataset would produce lower RMSE with Ridge or Lasso the data had to be preprocessed. The first step in the preprocessing was to conduct a near zero variance check and remove those variables. The near zero variance check was performed the same as in previous models. A Box-Cox transformation was then conducted on the CenteralAir, MAsVnRArea, BsmtFinSF2, and BSmtHalfBath.

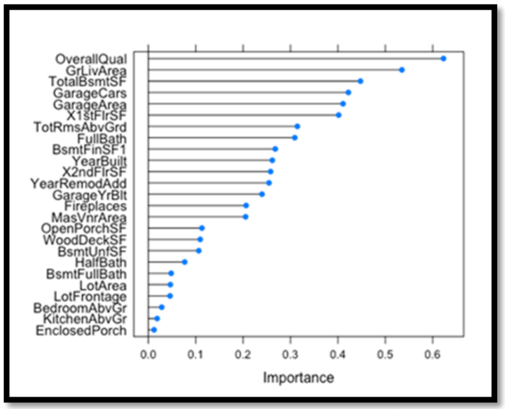


Figure 11 Top 5 Most Important Predictors

As part of the processing, categorical variables were transformed into ordinal variables. For example in the variable quality of basement or garage, the original classification was Ex, Gd, TA, FA, and

Po, these variables were transformed to 5,4,3,2, and 1. In this dataset, if the NA represented a feature not present in the dataset the variable was transformed from an NA to a 0 value. This transformation affected the ExterQual, ExterCond, BSMTQual, BSMTcond, BSMTExposire, BSMTFinType1, BSMTFinType2, HeatingQC, CenteralAir, KitchenQual, FireplaceQu, GarageFinish, GarageQual, GarageCond, PoolQuality, Street, LotShape, LandSlope, and Functional. For the numerical features that had missing values the median was imputed for the missing values. For example, LotFrontage had 259 missing values. These missing values were imputed with the median value of 68.

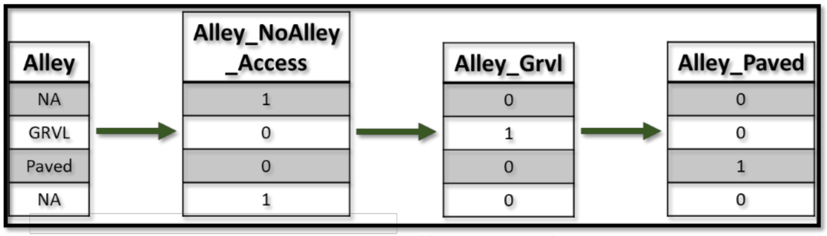
Once the data was processed it was fitted to the ridge regression model. The ridge regression applied a penalty to control for overfitting but would result in a reduced R^2. As a result of the penalty some coefficient would be reduced to near zero, but ridge will not reduce a coefficient to zero. The RMSE of 48521.84, R^2 of .6468478, an RMSE SD of 4313.133, and a R^2 SD of .02881520. A second Ridge model was fitted removing several highly correlated predictors. The variables removed were GrLivArea, GarageCars, TotalBsmtSf, YearBuilt, and FireplaceQu. These variables were removed due to their collinearity above .75. This resulted in a model with an RMSE of 48228.85, R^2 of .65193, an RMSE SD of 4796.79, and a R^2 SD of .02803259.

The filtered ridge model appears to perform slightly better than the complete ridge model. Both ridge models have lower R^2 which is to be expected as the penalty term introduced will reduce the RMSE at the expense of the R^2. If the model is producing better RMSE than the OLS the impact to R^2 should not be considered a negative issue. However, both models have a higher RMSE than PLS, PCR or OLS. The OLS model is still performing better than any of the models thus far.

The next step was to fit the data using a Lasso regression. The critical difference between the ridge and lasso is that ridge can reduce a coefficient to nearly zero while lasso can reduce a coefficient to zero. By reducing a coefficient to zero, the lasso regression can effectively act as a variable selection method. Fitting a lasso model to all variables resulted in RMSE of 48619.85, R^2 of .645442, an RMSE SD of 4468.969, and a R^2 SD of .02842689. The highly correlated predictors of GrLivArea, GarageCars, TotalBsmtSf, YearBuilt, and FireplaceQu were removed from the original dataset to recreate a filtered dataset for another lasso model. This filtered lasso model had an RMSE of 30768.21, R^2 of .6509745, an RMSE SD of 5107.29, and a R^2 SD of .02638728. To this point, this is the best result of any model.

Robust Linear Regression

In an effort to control the effect of outliers on the predictive power of the model a robust linear regression (RLM) was the next model evaluated. Before the model could be fitted the dataset had to be preprocessed first. The first set was to remove variables with a near zero variance. The highly correlated predictors of GrLivArea, GarageCars, TotalBsmtSf, YearBuilt, and FireplaceQu were also removed from the database. In addition to the previously mentioned variables being removed the additional variables which had over ten levels were excluded. This means the Neighborhood, Exterior 1, and Exterior 2 were removed. In addition to the standard data preprocessing of the near-zero variance check and highly correlated variable check binary variables were created from nominal predictors. For example, the alley variable has three classes of NA, Gravel, and Paved. To generate the dummy variable three additional predictors were created, the Alley_NoAlley_Access, Alley_Gravel, and Alley_Paved. From the original alley predictor if GRVL was listed, then there would be a zero for Alley_NoAlley_Access and Alley_Paved. For the Alley_Grvl predictor will be given a one value of that observation. The binary



Alley	Alley_NoAlley_Access	Alley_Grvl	Alley_Paved
NA	1	0	0
GRVL	0	1	0
Paved	0	0	1
NA	1	0	0

Figure 12 Dummy Variable Creation Example

transformation can be seen in Figure 10.

Once the data for the model was preprocessed, it was fit to the robust linear model. This robust linear regression model had an RMSE of 39713.41, R^2 of .72, an RMSE SD of 9153.579, and a R^2 SD of .08155716. Since the model included the creation of additional dummy variables, the analysis significantly increased the number of predictors. However, as a near zero variance was conducted the only variables which remained were variables with high levels of variance.

In an attempt to further reduce the effect of outliers on the predictive power of the model a spatial sign processing was conducted and applied to both an OLS model and another RLM model. The spatial sign procedure projects the predictor values onto a multidimensional sphere. This has the effect of making all the sample the same distance from the center of the sphere. Mathematically, each sample is divided by its squared norm. On the linear model with the spatial sing applied the RMSE of .0034, R^2 of .84, an RMSE SD of .0040796021, and a R^2 SD of .08793517. The results for the RLM with spatial sign were RMSE of .0061, R^2 of .8, an RMSE SD of .0048728716, and a R^2 SD of .04077799. This RMSE appears to be significantly lower, but the RMSE and R^2 have been scaled due to the use of the spatial sign. As a result, it is difficult to compare the results of the model preprocessed with the spatial sign to the models that were not preprocessed using the spatial sign.

Random Forest

The best thing about Random Forest is that it is not necessary to do preprocessing for it. The decreases the computational cost and the analysis can be performed quickly. Although, there was missing data which was to be dealt. So, removed all the predictors (Lot Frontage, Alley, FireplaceQu, PoolQC, Fence, MiscFeature) with missing observations more than 259. The good thing about Random Forest Regression is that it can deal with different types of data. So, there is no need to create dummy variables or convert them into any specific type to perform Random Forest on it.

After doing the preprocessing of missing data, the data was fit to Random Forest Regression. The model had a mean of squared residuals 795849829 and percentage of Var explained was 87.21. However, there were few predictors with negative %incMSE so had to remove those variables to improve the model. Those predictors were Id, PoolArea, GarageQual, GarageCond, LowQualFinSF, MiscVal, YrSold, RoofMatl, Electrical, Heating, Condition2, Street, MoSold, ExterCond. The model was fit again after removing all those variables and resulted in a mean of squared residuals 76199040 and % of Var explained 87.75 which was actually a bit better than the previous one. The top predictors were GrLiving Area, Neighborhood and OverallQual.

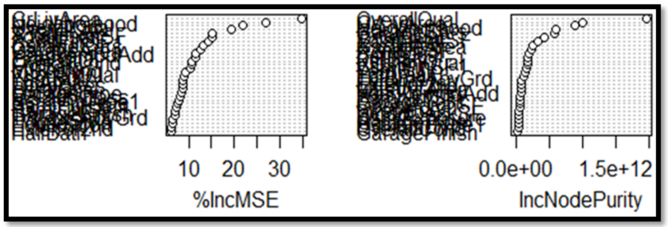


Figure 13 Random Forest Results

Support Vector Machine

The last model is a machine learning model known as a support vector machine. To run this model, it has to processed. The first step of data preprocessing was to recode the ordered factors associated with the condition of a feature with a number. Missing values within predictors associated with the existence of an alleyway, basement, garage, and the related quality/features were replaced with “None”. In these cases, the house feature was assumed to be absent; however, variables such as electrical type or roof material, missing values were replaced with the most common value, since the house was expected to have electricity and a roof. The missing values in lot frontage area were imputed with the median of the non-missing values. Predictors which exhibited high skewness were transformed by taking the log of the predictor. The following near zero variance predictors were removed from the data set: Street, Alley, LandContour, Utilities, LandSlope, Condition2, RoofMatl, BsmtCond, BsmtFinType2, BsmtFinSF2, Heating, LowQualFinSF, KitchenAbvGr, Functional, EnclosedPorch, X3SsnPorch, ScreenPorch, PoolArea, PoolQC, MiscFeature, MiscVal.

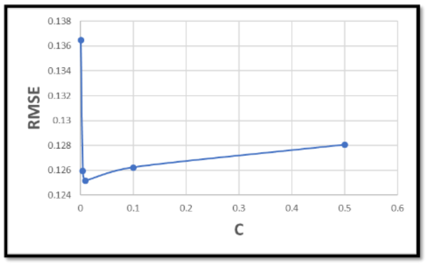


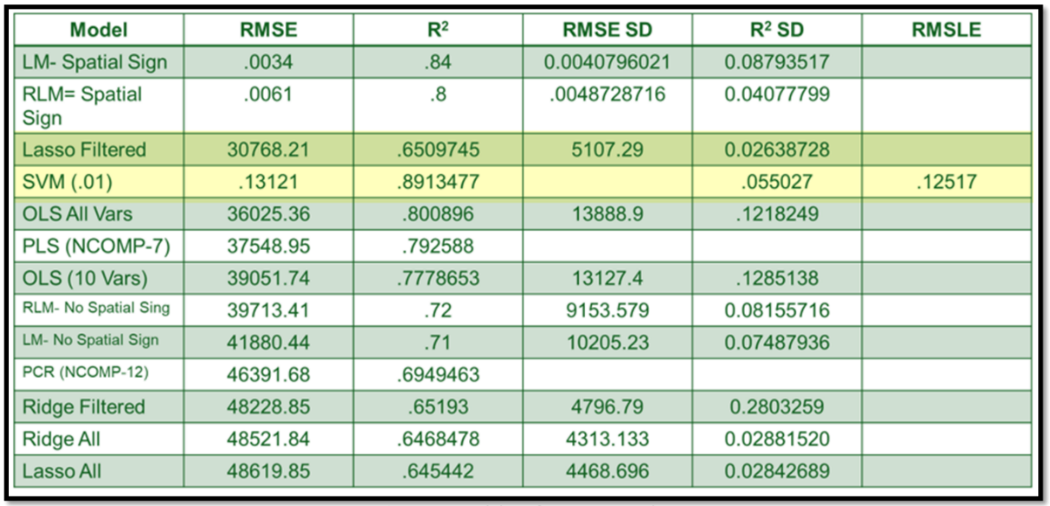
Figure 13 SVM Tuning Results

A support vector machine (SVM) model was trained for the preprocessed data set. A linear kernel was selected after various kernel were tried. In figure 11 are results for multiple values of the tuning parameter C. results were cross-validated using ten-fold cross-validation. A tuning parameter of C=.01 was selected, resulting in an RSME of 0.13121, R^2 of .8913477, an RMSE SD of .055027, and a R^2 SD of .12517. The sales price was predicted for the test data set, and the results were uploaded to Kaggle, the output scored in the top 29 percentile of results. Removing near zero variance predictors resulted in a higher R squared value. When checked against the test set on Kaggle effect of near variance predictors was almost negligible as the trained model with near zero variance predictors removed scored 0.12517 versus 0.12590 RSMLE with near zero variance predictors used for training. The model was able to capture a significant amount of the data variance however additional feature engineering may be able to improve the predictive power. While the model performed well it still took a considerable amount of time for the model tune. This will be an important consideration when comparing the SVM model to the other models.

Model Performance Review

Once all models have been fitted the performance, and predictive abilities were placed in the table in Figure 14. This table shows RMSE, R^2, RMSE, and R^2 for all the models. The models were placed in order to RMSE. The two models with the best overall performance were the Lasso Filtered model and the Support Vector Machine model. Due to scaling it is difficult to compare the Spatial Sign LM and the Spatial Sign RLM models. While they did have a lower RMSE then the Lasso Filtered or the SVM they were not chosen as the best models to represent the data due to the time and feature engineering required to fit the data to the model.

Both the Lasso Filtered and the SVM performed well, but it is the assessment the best model to use for the Ames housing dataset is the Lasso Filtered. The SVM is a complicated model and takes a great deal of time to tune to produce results. The results from the SVM are also difficult to interpret. This is in contrast to the time to process the data for the Lasso Filtered and the time to run the Lasso Filtered model. The Lasso Filtered is also a linear model, and it is still reasonably easy to interpret its results and explain those results to decision makers. Based upon these reasons in a business environment the Lasso Filtered model is the best model to use to give a prediction of housing values in an easy to interpret manner in a reasonable amount of time.



Model	RMSE	R^2	RMSE SD	R^2 SD	RMSLE
LM- Spatial Sign	.0034	.84	0.0040796021	0.08793517	
RLM= Spatial Sign	.0061	.8	.0048728716	0.04077799	
Lasso Filtered	30768.21	.6509745	5107.29	0.02638728	
SVM (.01)	.13121	.8913477		.055027	.12517
OLS All Vars	36025.36	.800896	13888.9	.1218249	
PLS (NCOMP-7)	37548.95	.792588			
OLS (10 Vars)	39051.74	.7778653	13127.4	.1285138	
RLM- No Spatial Sing	39713.41	.72	9153.579	0.08155716	
LM- No Spatial Sign	41880.44	.71	10205.23	0.07487936	
PCR (NCOMP-12)	46391.68	.6949463			
Ridge Filtered	48228.85	.65193	4796.79	0.2803259	
Ridge All	48521.84	.6468478	4313.133	0.02881520	
Lasso All	48619.85	.645442	4468.696	0.02842689	

Figure 14 Model Performance Results

Lessons Learned

To collect a dataset, process the data, and prepare the model to be fitted to several different model types a great deal was learned about the process of model creation. The first and possibly most important lesson was to allocate enough time for exploratory data analysis. It takes time to open a dataset and examine the variables. This can take even longer if there is not a clear description of the data resident in the dataset. If a variable is categorical in nature and only contains a grouping of two-letter codes in the observations, it can be difficult to assess the importance of the code. This is even more important if the code is represented numerically. The analyst has to decide if the number should be treated as a categorical variable or if the variable is numerical. A well-defined data key can help the model builder in this aspect.

The second lesson learned indirectly connected to the EDA. Enough time should be allowed to pre-process the data properly. Once the structure of the data is understood it is probably necessary to transform it into usable data type the model will be able to interpret. This pre-processing time is more than converting "NA" to zero values. This also includes checking the predictors for skewness and if required applied the correct transformation. The dataset then needs to be assessed for highly correlated predictors, and then a decision needs to be reached to either remove highly correlated predictors or attempt to use a model that will assist with the removal of high correlated predictors. Then the analyst needs to asses what should happen with missing values, should the values be imputed, if so, what value should be used or should the analyst remove observations with missing values. All of these questions about how to process the data take time and ample time should be allotted for data pre-processing. Compared to building and running the model often data preprocessing was the more time difficult of the processes. Compared to data preprocessing running the model is the easy part of the process.

The last lesson learned was the value of an interpretable model. These models will often be presented to a decision maker who needs to be able to understand the risk of a model. If the decision maker is unable to adequately understand the model parameters and the results from the model they may make a poor risk analysis or potentially lose an opportunity. In this case, if the predictive power for a simple model is similar to a complicated model, it is generally best to use the simple model. In the case of this project the Lasso Filtered and the SVM had similar predictive power, so in this case, it was probably best to use the least complicated model which is the Lasso Filtered model.

Conclusions

This project was able to provide a model that met the research objectives of strong predictive power, quick runtime, and easily interpretability. This was only accomplished after a great deal of time in exploratory data analysis, data pre-processing, building an almost exhaustive list of models, and finally comparing the benefits and detractors of each model. It was only through all of this that a model built that could help a realtor possibly gain a competitive advantage in being able to predict the potential price of a house.

References

Auguie, B. (2017). *gridExtra: Miscellaneous Functions for "Grid" Graphics*. Retrieved from https://CRAN.R-project.org/package=gridExtra

De Cock, D. (2011). Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project. *Journal of Statistics Education*, 1-15.

Kaggle. (2018, October 01). *Kaggle*. Retrieved from House Prices: Advanced Regression Techniques: https://www.kaggle.com/c/house-prices-advanced-regression-techniques#description

Kaplan, J. (2018). *Fast Creation of Dummy (Binary) Columns and Rows fromCategorical Variables*. Retrieved from chrome-extension://oemmndcbldboiebfnladdacbdfmadadm/https://cran.r-project.org/web/packages/fastDummies/fastDummies.pdf

Kuhn, M. (2018). *AppliedPredictiveModeling: Functions and Data Sets for 'Applied Predictive Modeling'*. Retrieved from https://CRAN.R-project.org/package=AppliedPredictiveModeling

Kuhn, M. (2018). *caret: Classification and Regression Training*. Retrieved from https://CRAN.R-project.org/package=caret

Leisch, F., & Dimitriadou, E. (2012). *mlbench: Machine Learning Benchmark Problems*. Retrieved from https://cran.r-project.org/web/packages/mlbench/index.html

Liaw, A. (2002). *Classification and Regression by randomForest*. Retrieved from https://CRAN.R-project.org/doc/Rnews/

Mevik, B.-H. (2018). *pls: Partial Least Squares and Principal Component Regression*. Retrieved from https://CRAN.R-project.org/package=pls

Meyer, D. (2018). *e1071: Misc Functions of the Department of Statistics, Probability, Theory Group*. Retrieved from https://CRAN.R-project.org/package=e1071

Milborrow, S. (2018). *earth: Multivariate Adaptive Regression Splines*. Retrieved from https://CRAN.R-project.org/package=earth

R Core Team. (2018). *R: A Language and Environment for Statistical Computing*. Retrieved from https://www.R-project.org/

RStudio. (2018). *RStudio*. Retrieved from http://www.rstudio.org/

Sarkar, D. (2008). *Lattice: Multivariate Data Visualization with R*. Retrieved from http://lmdvr.r-forge.r-project.org

Schloerke, B. (2018). *GGally: Extension to 'ggplot2'*. Retrieved from https://cran.r-project.org/web/packages/GGally/index.html

Wei, T., & Simko, V. (2017). *R package "corrplot": Visualization of a Correlation Matrix*. Retrieved from https://github.com/taiyun/corrplot

Wickham, H. (2011). *The Split-Apply-Combine Strategy for Data Analysis*. Retrieved from http://www.jstatsoft.org/v40/i01/

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Retrieved from http://ggplot2.org

Wickham, H. (2017). *dplyr: A Grammar of Data Manipulation*. Retrieved from https://CRAN.R-project.org/package=dplyr

Wickham, H. (2017). *tidyverse: Easily Install and Load the 'Tidyverse'*. Retrieved from https://CRAN.R-project.org/package=tidyverse

Zhang, D. (2018). *rsq: R-Squared and Related Measures*. Retrieved from https://CRAN.R-project.org/package=rsq

Zou, H. (2018). *elasticnet: Elastic-Net for Sparse Estimation and Sparse PCA*. Retrieved from https://CRAN.R-project.org/package=elasticnet