Data-Driven Real Estate: Multiple Linear Regression of Home Prices

What determines the final sales price of a home? A knee-jerk answer of “location” may come to mind. However, we propose a quantitative solution. We have analyzed a comprehensive real estate data set to develop a model to predict the final sales price of homes sold in Ames, Iowa. The model presented herein may be of interest to buyers, sellers, contractors, or simply those wishing to understand the critical drivers in the often-overwhelming process of home buying. The results are only relevant to Ames, IA. We encourage those interested in building models for other regions to understand our analysis workflow: while we will end up with completely different models, the variable selection methods demonstrated can be applied to other studies.

Problem Statement

We aim to develop a multiple linear regression model based on a data set of explanatory variables collected in Ames, IA to accurately predict home final sales price.

Limitations

This study is observational: one cannot make causal inferences between the explanatory variables and predicted sale price. Also, the data is only representative of the Ames, Iowa market. Without any randomness in the sample selection, conclusions derived are downsized in scope to this dataset only. Thus, any generalization would be highly skeptical and should be avoided.

An additional limitation is there may be variables influencing the final sale price that are not captured in the master data set. If the model was deployed to, for example, a real estate agency, model evolutions would likely occur with the addition of new explanatory variables. Also, as with any model, our final model cannot predict housing market declines, sluggish housing starts, or other force majeure events. Another out of scope issue is that there may be evidence of significant nonlinear combinations of variables. However, this analysis seeks to find a simple, linear solution.

Data Set Description

There are 2 different data sets provided, both consist of housing sales data from 2006-2010 in the Ames, Iowa area:

- Train file with 1460 records with a record identifier, 75 explanatory variables and 1 response variable (SalePrice). This file is used to develop the linear model to explain the variation in SalePrice based on selected explanatory variables.
- Test file with 1459 records with a record identifier and 79 explanatory variables. This file contains 4 additional variables as compared to the Train file namely Alley, PoolQC, Fence and MiscFeature which won’t be used in the prediction.

A linear model using the train data will be used to predict the response (Sale Price of the house) in the test file. The explanatory variables include: lot size, bedrooms, bathrooms, square footage, and many other features that can be important for those buying a house. The predicted values will then be uploaded to Kaggle and the Kaggle Score will be used to determine how well the model predicts sales price. The full description of the 79 pre-analysis variables can be found here: Data dictionary in Kaggle.

Exploratory Data Analysis

The starting point to build the prediction model was to consider the kind of data that is present in the two data sets. Upon examining the data, we noticed many values as -1 in the train data set. However, there is no such instance in test data. On further examination, we found that in the train data, it appeared that instead of NA, -1 was populated in the data source. The column headings were also impacted due to this issue. For example, the column heading KitchenAbvGr appeared as Kitche-1bvGr in the train file. Keeping this in mind, our data cleanup step involved similar treatment to 'NA' or '.'or '-1' values.

One key point in this analysis is there are many variables, likely some not captured in the master data set, that could influence the final sales price of a home in Ames, IA. A qualitative analysis of variable names indicates some redundancy. For example, there are four variables quantifying the presence of various bathroom types. We sought to reduce this mentioned complexity by simply combining similar terms. For a detailed list of created variables, please refer to the variable description section. These combinations were made to achieve a concise model with robust predictors.

An area of focus for us initially was on the square footage variables; there appeared to be an opportunity to combine some of those variables.

TotalFinishedSqFt: Total finished square feet of a house, obtained by sum of GrLivArea, GarageArea (if finished) and result of substraction of BsmtUnfSF from TotalBsmtSF
TotalConSqFt: Total constructed area of the house, obtained by sum of GrLivArea, TotalBsmtSF and GarageArea
TotalOutSF: Total porch and deck area of the house, obtained by sum of WoodDeckSF, OpenPorchSF, EnclosedPorch, SsnPorch and ScreenPorch

After combining terms, we plotted all the square foot variables against the SalePrice using Matrix Scatterplots looking for evidence of a linear correlation (Figure 1). Reviewing these results, we felt it was necessary to calculate a log transformation of the SalePrice and square footage variables. We verified these results and can be seen in Figure 2. From the graphs, we concluded that there is strong correlation between TotalFinishedSF and TotalConSqFt. Thus, we decided to include only TotalConSqFt in our model. Also, a plot of TotalOutSF and MasVnrArea suggested that it can be divided into two sets – one with zero values and another having non-zero values. So, we created categorical variables named OutSFFlag and VnrAreaFlag with values 0 and 1.

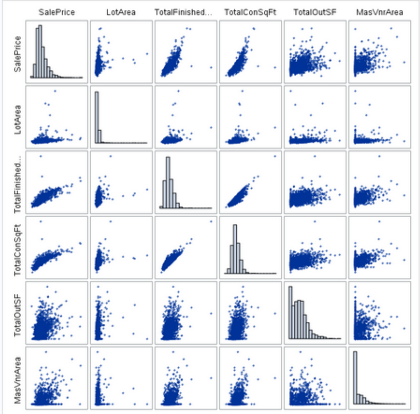


Figure 1: SalePrice vs Square Footage Variables

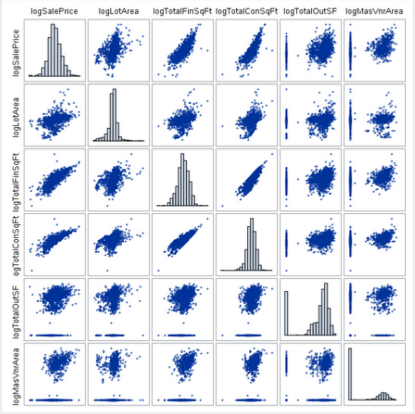


Figure 2: logSalePrice vs log of Square Footage Variables

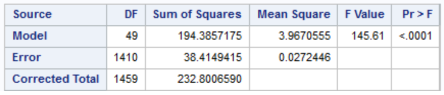
Our intuitive exploratory analysis also led us to graphically explore our potential categorical and continuous predictors. While we did perform an exhaustive analysis on all variables, it is encouraged that the following observations are validated from a real estate subject matter expert in Ames:

- ID's 1299 and 524 due to potentially incorrectly recorded square footage
- ID 633 due to very low final sales price of the home compared to other homes in the data set

With many more variables to consider as predictors, we applied both visual analysis and descriptive statistics to explore all other possible variables. We quickly saw that there were variables that should be excluded for several reasons. A few examples are MSZoning, MSSubClass, and LandContour. For MsZoning, 79% of the data had a value of RL. MSSubClass indicated that more than 1/3 of the homes are 1 story, 1946 and Newer, All Styles. LandContour had 90% of homes listed as LVL. There were several categorical variables that could be removed using this logic. We felt these types of variables would not help in ultimately predicting the sales price of a home.

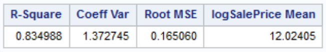
Variable Screening

One tactic we employed to reduce complexity in the model is to aggregate certain variables. The need for this arose when variables, although measuring different portions of a house, proved to be intuitively aggregated (total bathrooms for example). After visual inspection of predictors, we started with a preliminary model with just two variables - Neighborhood and logTotalConSqFt (Total constructed area – Sum of GrLivArea + TotalBsmtSF + GarageArea) with interaction term. We found this model significant and explained 83.5% of variation in SalePrice.



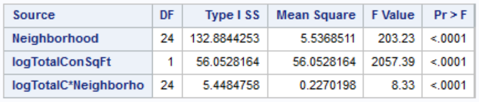
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	49	194.3857175	3.9670555	145.61	<.0001
Error	1410	38.4149415	0.0272446		
Corrected Total	1459	232.8006590			

Table 1: Overall ANOVA - Baseline Model



R-Square	Coeff Var	Root MSE	logSalePrice Mean
0.834988	1.372745	0.165060	12.02405

Table 2: Baseline Model Summary

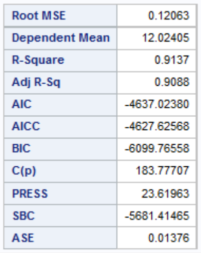
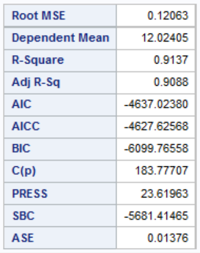
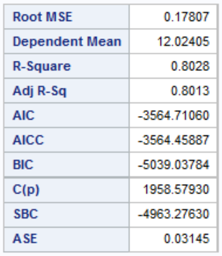
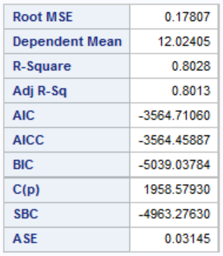


Source	DF	Type I SS	Mean Square	F Value	Pr > F
Neighborhood	24	132.8844253	5.5368511	203.23	<.0001
logTotalConSqFt	1	56.0528164	56.0528164	2057.39	<.0001
logTotalC*Neighborho	24	5.4484758	0.2270198	8.33	<.0001

Table 3: Baseline Significance

Realizing we could improve upon this baseline model, we utilized the following variable selection techniques: LASSO, LAR, backward, and stepwise to pick other variables that would ultimately help us make better predictions. We do recognize the inherent weaknesses associated with stepwise and backward variable selection, but also understand that such techniques may provide insight into variables previously considered weak. Through the iterative modeling process, the above selection methods certainly produced some variable candidates that were common in several runs. As such, these variables deserve a thorough screening, both quantitatively and qualitatively. Out of the 75 explanatory variables, we picked a subset of what appeared to be linear predictors and ran the variable selection algorithms and then based on the results, we built our custom model. See Table 4 below.

Table 4: Master Summary of Model Selection Techniques

Model Name	**Backward**	**Stepwise**	**LASSO**	**LAR**
Variables in the selected model	Fireplaces, logTotalOutSF, GarageCars, logLotArea, Neighborhood, SaleCondition, OverallQual, OverallCond, PropAge, TotalBath, OutSFFlag, logTotalConSqFt*Neighborhood	Fireplaces, logTotalOutSF, GarageCars, logLotArea, Neighborhood, SaleCondition, OverallQual, OverallCond, PropAge, TotalBath, OutSFFlag, logTotalConSqFt*Neighborhood	Fireplaces, GarageCars, logLotArea, OverallQual_8, OverallQual_9, HeatingQC_Ex, KitchenQual_Ex, KitchenQual_TA, PropAge, TotalBath, logTotalConSqFt	Fireplaces, GarageCars, logLotArea, OverallQual_8, OverallQual_9, HeatingQC_Ex, KitchenQual_Ex, KitchenQual_TA, PropAge, TotalBath, logTotalConSqFt
Parameter table of the model	 Root MSE 0.12063 Dependent Mean 12.02405 R-Square 0.9137 Adj R-Sq 0.9088 AIC -4637.02380 AICC -4627.62568 BIC -6099.76558 C(p) 183.77707 PRESS 23.61963 SBC -5681.41465 ASE 0.01376	 Root MSE 0.12063 Dependent Mean 12.02405 R-Square 0.9137 Adj R-Sq 0.9088 AIC -4637.02380 AICC -4627.62568 BIC -6099.76558 C(p) 183.77707 PRESS 23.61963 SBC -5681.41465 ASE 0.01376	 Root MSE 0.17807 Dependent Mean 12.02405 R-Square 0.8028 Adj R-Sq 0.8013 AIC -3564.71060 AICC -3564.45887 BIC -5039.03784 C(p) 1958.57930 SBC -4963.27630 ASE 0.03145	 Root MSE 0.17807 Dependent Mean 12.02405 R-Square 0.8028 Adj R-Sq 0.8013 AIC -3564.71060 AICC -3564.45887 BIC -5039.03784 C(p) 1958.57930 SBC -4963.27630 ASE 0.03145

Considering the results, we created some new categorical variables. For Example, for variable HeatingQC, only 'TA' and 'Ex' values were shown significant and picked by the LASSO and LAR models so we created HeatingQCClass variable to have a default value of 0, value 1 if HeatingQC is 'TA' and value 2 if HeatingQC is 'Ex'. Below were the conclusions made from the results of Automatic Model Selection Algorithms:

a. Important Categorical Variables to be included in the model:
SaleCondition, OverallQual, OverallCond, Neighborhood,

b. Important Continuous Variables to be included in the model:
logTotalOutSF, TotalBath, PropAge, LogTotalConSqFt, LogLotArea, FirePlaces, GarageCars

c. Interaction term:
LogTotalConSqFt*Neighborhood

d. New Categorical Variables Created:
kitchenQualClass: Default value=0, 1 if kitchenQual='TA', 2 if kitchenQual='Ex'
HeatingQCClass: Default value=0, 1 if HeatingQC='TA', 2 if HeatingQC='Ex'

NOTE: - Inclusion of variables Condition1, Condition2 and bldgtype into the model improved the adjusted R-square and a better Kaggle score but since the additional explained variance was just 0.0036, we decided to drop them from the model to keep it concise.

Model Selection and Analysis

Considering the variables selected above, before proceeding with our final model, we created figure 3 as shown below to get a visual feel of the linear relationship, as well as validate our findings thus far that our explanatory variables have a strong linear relationship with the dependent variable. The chosen distinct count variables also help explain variation of home sale prices in Ames, IA.

After our Exploratory Data Analysis and Variable Screening, the below variables were selected to remain as we proceed with selection methods in SAS to choose our final linear model.

Categorical

- Neighborhood: the actual neighborhood of the home sold
- OverallQual: short for 'overall quality,' this variable indicates the overall material and finish quality
- OverallCond: short for 'overall condition,' this variable indicates the overall condition rating
- SaleCondition: Condition of Sale
- KitchenQualCLass: this variable indicates the quality class of kitchens (1 if Kitchen Quality='TA', 2 if 'Ex', 0 otherwise).
- HeatingQCClass: heating quality class (1 if Heating Quality='TA', 2 if 'Ex', 0 otherwise)
- OutSFFlag: variable showing whether porch/deck exists or not.

Numerical

It is recommended one references the EDA section of this analysis to gain clarification on the need for the created variables

- GarageCars: the size of garage in car capacity
- logTotalOutSF: the logarithmic transformation of TotalOutSF, a calculated variable. TotalOutSF is a summation of WoodDeckSF (*Wood deck in square feet*), OpenPorchSF (*Open porch area in square feet*), EnclosedPorch (*Enclosed porch area in square feet*), SSnPorch (*Three season porch area in square feet*), ScreenPorch (*screen porch in square feet*)
- PropAge: a calculated variable. Property age is equal to YrSold (*Year sold*) – YearBuilt(*original construction date*)
- TotalBath: a calculated variable. TotalBath is a summation of BsmtFullBath(*basement full bathrooms*), FullBath(*full bathrooms above ground*), BsmtHalfBath/2(*basement half baths*), HalfBath/2(*half baths above ground*). The need for this variable comes down to a more concise model. Please see the EDA section for additional information.
- logTotalConSqFt: the logarithmic transform of TotalConSqFt, a calculated variable. TotalConSqFt is a summation of GrLivArea(*Above grade/ground living area in square feet*), TotalBsmtSF(*total square feet of basement area*), GarageArea(*size of garage in square feet*)
- logLotArea: the logarithmic transform of lotArea, short for lot size in square feet
- Fireplaces: number of fireplaces

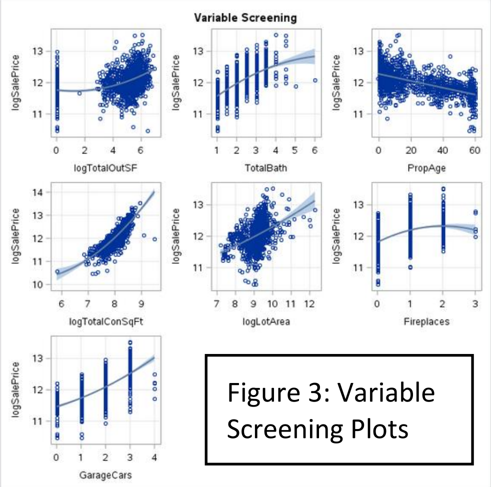
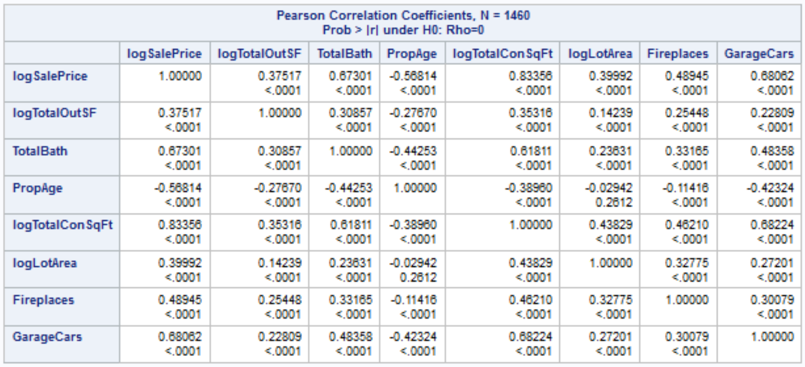


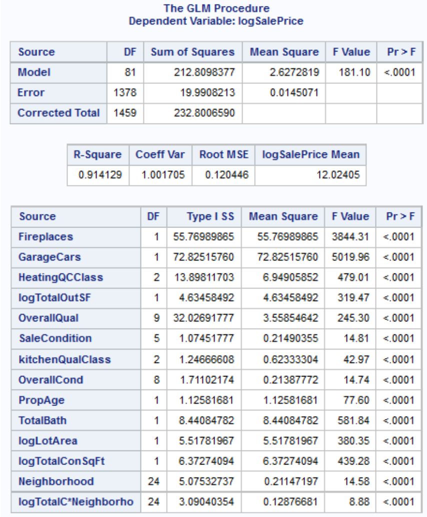
Figure 3: Variable Screening Plots



Pearson Correlation Coefficients, N = 1460
Prob > |r| under H0: Rho=0

	logSalePrice	logTotalOutSF	TotalBath	PropAge	logTotalConSqFt	logLotArea	Fireplaces	GarageCars
logSalePrice	1.00000	0.37517 <.0001	0.67301 <.0001	-0.56814 <.0001	0.83356 <.0001	0.39992 <.0001	0.48945 <.0001	0.68062 <.0001
logTotalOutSF	0.37517 <.0001	1.00000	0.30857 <.0001	-0.27670 <.0001	0.35316 <.0001	0.14239 <.0001	0.25448 <.0001	0.22809 <.0001
TotalBath	0.67301 <.0001	0.30857 <.0001	1.00000	-0.44253 <.0001	0.61811 <.0001	0.23631 <.0001	0.33165 <.0001	0.48358 <.0001
PropAge	-0.56814 <.0001	-0.27670 <.0001	-0.44253 <.0001	1.00000	-0.38960 <.0001	-0.02942 0.2612	-0.11416 <.0001	-0.42324 <.0001
logTotalConSqFt	0.83356 <.0001	0.35316 <.0001	0.61811 <.0001	-0.38960 <.0001	1.00000	0.43829 <.0001	0.46210 <.0001	0.68224 <.0001
logLotArea	0.39992 <.0001	0.14239 <.0001	0.23631 <.0001	-0.02942 0.2612	0.43829 <.0001	1.00000	0.32775 <.0001	0.27201 <.0001
Fireplaces	0.48945 <.0001	0.25448 <.0001	0.33165 <.0001	-0.11416 <.0001	0.46210 <.0001	0.32775 <.0001	1.00000	0.30079 <.0001
GarageCars	0.68062 <.0001	0.22809 <.0001	0.48358 <.0001	-0.42324 <.0001	0.68224 <.0001	0.27201 <.0001	0.30079 <.0001	1.00000

Figure 4: Pearson Correlation Coefficients



The GLM Procedure
Dependent Variable: logSalePrice

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	81	212.8098377	2.6272819	181.10	<.0001
Error	1378	19.9908213	0.0145071		
Corrected Total	1459	232.8006590			

R-Square	Coeff Var	Root MSE	logSalePrice Mean
0.914129	1.001705	0.120446	12.02405

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Fireplaces	1	55.76989865	55.76989865	3844.31	<.0001
GarageCars	1	72.82515760	72.82515760	5019.96	<.0001
HeatingQCClass	2	13.89811703	6.94905852	479.01	<.0001
logTotalOutSF	1	4.63458492	4.63458492	319.47	<.0001
OverallQual	9	32.02691777	3.55854642	245.30	<.0001
SaleCondition	5	1.07451777	0.21490355	14.81	<.0001
kitchenQualClass	2	1.24666608	0.62333304	42.97	<.0001
OverallCond	8	1.71102174	0.21387772	14.74	<.0001
PropAge	1	1.12581681	1.12581681	77.60	<.0001
TotalBath	1	8.44084782	8.44084782	581.84	<.0001
logLotArea	1	5.51781967	5.51781967	380.35	<.0001
logTotalConSqFt	1	6.37274094	6.37274094	439.28	<.0001
Neighborhood	24	5.07532737	0.21147197	14.58	<.0001
logTotalC*Neighborho	24	3.09040354	0.12876681	8.88	<.0001

Figure 5: Final model parameters

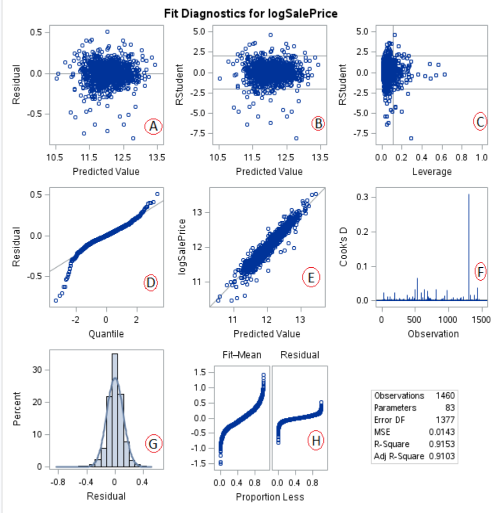
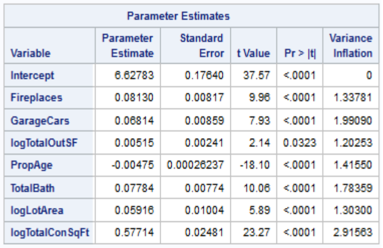


Figure 6: Final model graphs



Parameter Estimates					
Variable	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variance Inflation
Intercept	6.62783	0.17640	37.57	<.0001	0
Fireplaces	0.08130	0.00817	9.96	<.0001	1.33781
GarageCars	0.06814	0.00859	7.93	<.0001	1.99090
logTotalOutSF	0.00515	0.00241	2.14	0.0323	1.20253
PropAge	-0.00475	0.00026237	-18.10	<.0001	1.41550
TotalBath	0.07784	0.00774	10.06	<.0001	1.78359
logLotArea	0.05916	0.01004	5.89	<.0001	1.30300
logTotalConSqFt	0.57714	0.02481	23.27	<.0001	2.91563

Figure 7: VIF for numerical variables in final model

Check for Assumptions:

1. **Linearity:** Pearson correlation coefficient (figure 4) for the continuous variables are provided as assurance that we have significant linear predictors. The need for the logarithmic transform originates from right-skewness. Plots of

our chosen predictors, as visual evidence of linearity can be found in the variable screening section (Figure 3). The VIF (Figure 7) calculations also reveal that multicollinearity is not a problem in this model.

2. **Normality of Residuals**: The histogram of residuals (Figure 6 -G) indicates an approximately normal distribution. There is a slight left skewness present. However, neither the data indicates large outliers nor is our working training data sample small. So, we can safely assume that the normality assumption is met.
3. **Constant Variance**: From the residual and QQ plot, normality and constant variance assumption are met. Though there are some outliers and the leverage plot (Figure 6 - C) shows a few observations having leverage, but since Cooks D plot (Figure 6 - F) does not indicate any influential points, the constant variance assumption is met.
4. **Independent Observations**: Observe the predicted versus residual plot: no concerning patterns exist, which leads us to our final assumption: independence. The residual plot does not indicate the presence of serial correlation.

Conclusions

As our analysis suggests, the final sales price of homes in Ames, IA cannot merely be reduced to the real estate adage 'location, location, location." Through a process of visual inspection, variable selection techniques, and descriptive statistics, we present a working predictive model for the Ames market. It is our recommendation to use the final developed model, as we have applied the model to an independent validation/test data set. The low kaggle score of .14 indicates this model will perform well in a production setting. The low Kaggle score also indicates we have overcome the overfitting trap. 91.4% of variation in the Sale Price of the train data is accounted for in the custom model.

If end users are real estate agents, perhaps this analysis could create value in further stochastic modeling for upcoming negotiations. Conversely, buyers could apply the included findings to target desired attributes while maintaining good financial position. In terms of numeric predictors, we assert that "big picture" variables, such as our aggregated space variables, see variable description for details, are ideal predictors for this housing market. While we recognize aggregation is important to this analysis, amenities such as fireplaces proved to be influential. A similar logic applies to our categorical variable selection; attributes such as neighborhood, quality indicators, and building type are all considered "big picture" variables. Our model is a comprehensive approach to predicting the sale price of homes in Ames, IA that holds relevance for both buyers and sellers alike.

SAS Appendix: SAS CODE

```
/*Bring in train data to SAS workspace*/
DATA train;
 infile "/home/harisanadhya0/sasuser.v94/MSDS 6372/Project 1/train1.csv" dsd
delimiter="," firstobs=2;
 input Id MSSubClass MSZoning $ LotFrontage LotArea Street $ LotShape $
LandContour $ Utilities $ LotConfig $ LandSlope $ Neighborhood $ Condition1 $
Condition2 $ BldgType $ HouseStyle $ OverallQual OverallCond YearBuilt
YearRemodAdd RoofStyle $ RoofMatl $ Exterior1st $ Exterior2nd $ MasVnrType
$ MasVnrArea ExterQual $ ExterCond $ Foundation $ BsmtQual $ BsmtCond $
BsmtExposure $ BsmtFinType1 $ BsmtFinSF1 BsmtFinType2 $ BsmtFinSF2
BsmtUnfSF TotalBsmtSF Heating $ HeatingQC $ CentralAir $ Electrical $
firstFlrSF secondFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath
FullBath HalfBath BedroomAbvGr KitchenAbvGr KitchenQual $ TotRmsAbvGrd
Functional $ Fireplaces FireplaceQu $ GarageType $ GarageYrBlt GarageFinish $
GarageCars GarageArea GarageQual $ GarageCond $ PavedDrive $
WoodDeckSF OpenPorchSF EnclosedPorch SsnPorch ScreenPorch PoolArea
MiscVal MoSold YrSold SaleType $ SaleCondition $ SalePrice;
/*Create a season indicator for possible use*/
if MoSold = 12 or MoSold = 1 or MoSold = 2 then Season = "Winter";
else if MoSold=3 or MoSold=4 or MoSold=5 then Season = "Spring";
else if MoSold=6 or MoSold=7 or MoSold=8 then Season = "Summer";
else Season = "Fall";
run;

/*Data clean up and log transformations as incidcated*/
Data train1;
set train;
/*  missing or NA or -1 values treated as 0 */
if GrLivArea=. or GrLivArea<0 then GrLivArea=0;
if TotalBsmtSF=. or TotalBsmtSF<0 then TotalBsmtSF=0;
if BsmtUnfSF=. or BsmtUnfSF<0 then BsmtUnfSF=0;
if GarageArea=. or GarageArea<0 then GarageArea=0;
if LotFrontage = . or LotFrontAge<0 then LotFrontage = 0;
if garageCars=. then garageCars=0;
if GarageYrBlt = . OR GarageYrBlt <0 then GarageYrBlt = 0;
if MasVnrArea<1 or MasVnrArea=. then MasVnrArea=0;
if Fireplaces=. or Fireplaces<0 then Fireplaces=0;
if BsmtFullBath=. or BsmtFullBath<0 then BsmtFullBath=0;
if FullBath=. or FullBath<0 then FullBath=0;
if BsmtHalfBath=. or BsmtHalfBath<0 then BsmtHalfBath=0;
if HalfBath=. or HalfBath<0 then HalfBath=0;
/* Cleanup of String Values */
if Neighborhood = "-1mes" then Neighborhood = "NAmes";
if MasVnrType = "NA" then MasVnrType = "None";
/* Create Property Age variable (PropAge) - Continuous variable  */
/* Subtract Year Remodeled or year home built from year sold to provide
relative age of home at time of sale */
if YearRemodAdd>0 then PropAge=yrsold-YearRemodAdd;
else PropAge=yrsold-YearBuilt;
/*Create poolFlag, this indicates if the home has a pool or not*/
poolFlag='N';
if PoolArea>0 then poolFlag='Y';
/*Combine logical square footage variables, perform log transformation on
results, also perform log transformation of SalePrice*/
logSalePrice = log(SalePrice);
logLotArea = log(LotArea);
TotalFinishedSqFt = GrLivArea + TotalBsmtSF - BsmtUnfSF;
if GarageFinish="Fin" or GarageFinish="RFn" then
TotalFinishedSqFt=TotalFinishedSqFt+GarageArea;
```

```
logTotalFinSqFt = log(TotalFinishedSqFt);
TotalConSqFt = GrLivArea + TotalBsmtSF + GarageArea;
logTotalConSqFt = log(TotalConSqFt);
TotalOutSF = WoodDeckSF + OpenPorchSF + EnclosedPorch + SsnPorch +
ScreenPorch;
logTotalOutSF = log(TotalOutSF+1);
logLotFrontage = log(LotFrontage+1);
logMasVnrArea=log(MasVnrArea+1);
/* Combine Bath Data into Total Bath */
TotalBath = BsmtFullBath + FullBath + BsmtHalfBath/2 + HalfBath/2;
run;
proc sgscatter data = train1;
matrix SalePrice LotArea TotalFinishedSqFt TotalConSqFt TotalOutSF
MasVnrArea/ diagonal=(histogram);
run;
proc sgscatter data = train1;
matrix logSalePrice logLotArea logTotalFinSqFt logTotalConSqFt logTotalOutSF
logMasVnrArea/ diagonal=(histogram);
run;
/* Creation of flags TotalOutSF and MasVnrArea */
data train1;
set train1;
OutSFFlag =0;
if TotalOutSF>0 then OutSFFlag=1;
VnrAreaFlag =0;
if MasVnrArea>0 then VnrAreaFlag =1;
run;
proc sgscatter data = train1;
matrix logSalePrice logLotArea logTotalFinSqFt logTotalConSqFt logTotalOutSF
logMasVnrArea logLotFrontage TotalBath BedroomAbvGr KitchenAbvGr
TotRmsAbvGrd Fireplaces GarageYrBlt GarageCars MiscVal MoSold YrSold ;
run;
/* preliminary model */
proc glm data=train1 plots=all ;
class Neighborhood;
model logsaleprice = Neighborhood logTotalConSqFt
Neighborhood*logTotalConSqFt;
run;
/* 2-Way ANOVA - Condition 1 and Condition 2 */
proc glm data=train1 plots=all;
class Condition1 Condition2;
model logsaleprice= Condition1 Condition2 Condition1*Condition2/solution;
lsmeans Condition1 / pdiff tdiff adjust=bon;
run;
/* Interaction term Condition1*Condition2 is not significant but the variables
individually are significant */
/* Model with just Condition 1, Condition 2 and BldgType*/
proc glm data=train1 plots=all;
class Condition1 Condition2 bldgtype Neighborhood;
model logsaleprice= Condition1 Condition2 bldgtype;
lsmeans Condition1 / pdiff tdiff adjust=bon;
run;
/* Model significant with R-square as 0.083 */
ods graphics on;
/* Stepwise */
proc glmselect data=train1
 seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);
class Neighborhood OutSFFlag VnrAreaFlag BldgType Utilities HouseStyle
Condition1 Condition2 OverallQual poolFlag OverallCond RoofStyle HeatingQC
kitchenqual SaleType SaleCondition Exterior1st Exterior2nd ;
model logsaleprice= VnrAreaFlag logMasVnrArea VnrAreaFlag*logMasVnrArea
MoSold Utilities HouseStyle Fireplaces logTotalOutSF logLotFrontage
GarageCars loglotarea Neighborhood BldgType poolFlag RoofStyle SaleType
SaleCondition Exterior1st Exterior2nd OverallQual HeatingQC kitchenqual
OverallCond Condition1 Condition2 PropAge TotalBath OutSFFlag
logTotalConSqFt OutSFFlag*logTotalConSqFt logTotalConSqFt*Neighborhood/
selection = stepwise (stop=SBC) cvmethod=random(5) stats=ALL;
run;
/* Backward */
proc glmselect data=train1
 seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);
class Neighborhood OutSFFlag VnrAreaFlag BldgType Utilities HouseStyle
Condition1 Condition2 OverallQual poolFlag OverallCond RoofStyle HeatingQC
kitchenqual SaleType SaleCondition Exterior1st Exterior2nd ;
model logsaleprice= VnrAreaFlag logMasVnrArea VnrAreaFlag*logMasVnrArea
MoSold Utilities HouseStyle Fireplaces logTotalOutSF logTotalOutSF
logLotFrontage GarageCars loglotarea Neighborhood BldgType poolFlag
RoofStyle SaleType SaleCondition Exterior1st Exterior2nd  OverallQual
HeatingQC kitchenqual OverallCond Condition1 Condition2 PropAge TotalBath
OutSFFlag logTotalConSqFt OutSFFlag*logTotalConSqFt
logTotalConSqFt*Neighborhood/ selection = forward (stop=SBC)
cvmethod=random(5) stats=ALL;
run;
/* Lasso */
proc glmselect data=train1;
class Neighborhood OutSFFlag VnrAreaFlag BldgType Utilities HouseStyle
Condition1 Condition2 OverallQual poolFlag OverallCond RoofStyle HeatingQC
kitchenqual SaleType SaleCondition Exterior1st Exterior2nd ;
model logsaleprice= VnrAreaFlag logMasVnrArea VnrAreaFlag*logMasVnrArea
MoSold Utilities HouseStyle Fireplaces logTotalOutSF logTotalOutSF
logLotFrontage GarageCars loglotarea Neighborhood BldgType poolFlag
RoofStyle SaleType SaleCondition Exterior1st Exterior2nd  OverallQual
HeatingQC kitchenqual OverallCond Condition1 Condition2 PropAge TotalBath
OutSFFlag logTotalConSqFt OutSFFlag*logTotalConSqFt
logTotalConSqFt*Neighborhood/ selection = laSSo (stop=SBC)
cvmethod=random(5) stats=ALL;
run;
/* LAR */
proc glmselect data=train1;
class Neighborhood OutSFFlag VnrAreaFlag BldgType Utilities HouseStyle
Condition1 Condition2 OverallQual poolFlag OverallCond RoofStyle HeatingQC
kitchenqual SaleType SaleCondition Exterior1st Exterior2nd ;
model logsaleprice= VnrAreaFlag logMasVnrArea VnrAreaFlag*logMasVnrArea
MoSold Utilities HouseStyle Fireplaces logTotalOutSF logTotalOutSF
logLotFrontage GarageCars loglotarea Neighborhood BldgType poolFlag
RoofStyle SaleType SaleCondition Exterior1st Exterior2nd OverallQual
HeatingQC kitchenqual OverallCond Condition1 Condition2 PropAge TotalBath
OutSFFlag logTotalConSqFt OutSFFlag*logTotalConSqFt
logTotalConSqFt*Neighborhood/ selection = lar (stop=SBC)
cvmethod=random(5) stats=ALL;
run;
/* New Variables creation */
data train1;
set train1;
kitchenQualClass = 0;
if KitchenQual='TA' then kitchenQualClass=1;
if KitchenQual='Ex' then kitchenQualClass=2;
HeatingQCClass = 0;
if HeatingQC='TA' then HeatingQCClass=1;
if HeatingQC='Ex' then HeatingQCClass=2;
run;
Title 'Correlation Table';
proc corr data=train1;
var logsaleprice logTotalOutSF TotalBath PropAge LogTotalConSqFt LogLotArea
FirePlaces GarageCars;
run;
Title "Variable Screening";
proc sgscatter data=train1;
plot (logsaleprice) * (logTotalOutSF TotalBath PropAge LogTotalConSqFt
LogLotArea FirePlaces GarageCars) / reg=(nogroup clm degree=2) grid;
```

```
run;
title ;
/* Final model chosen */
proc glm data=train1 plots=all;
class Neighborhood OverallQual OverallCond SaleCondition HeatingQCClass
kitchenQualClass ;
model logsaleprice = Fireplaces GarageCars HeatingQCClass  logTotalOutSF
OverallQual SaleCondition kitchenQualClass OverallCond PropAge TotalBath
loglotarea logTotalConSqFt Neighborhood logTotalConSqFt*Neighborhood/
tolerance cli solution;
output out = t student = res h = lev stdr=stdres cookd = cookd;
run;
/* Final model in addition to Condition1, Condition2 and bldgtype */
proc glm data=train1 plots=all;
class Neighborhood OverallQual OverallCond SaleCondition HeatingQCClass
kitchenQualClass Condition1 Condition2 bldgtype;
model logsaleprice = Fireplaces GarageCars HeatingQCClass Condition1
Condition2 bldgtype logTotalOutSF OverallQual SaleCondition kitchenQualClass
OverallCond PropAge TotalBath loglotarea logTotalConSqFt Neighborhood
logTotalConSqFt*Neighborhood/ tolerance cli solution;
output out = t student = res h = lev stdr=stdres cookd = cookd;
run;
/*Bring in Test data to SAS to use to predict Sale Price for this set of data*/
DATA test;
 infile "/home/harisanadhya0/sasuser.v94/MSDS 6372/Project 1/test.csv" dsd
delimiter="," firstobs=2;
 input Id MSSubClass MSZoning $ LotFrontage LotArea Street $ Alley $ LotShape
$ LandContour $ Utilities $ LotConfig $ LandSlope $ Neighborhood $ Condition1
$ Condition2 $ BldgType $ HouseStyle $ OverallQual OverallCond YearBuilt
YearRemodAdd RoofStyle $ RoofMatl $ Exterior1st $ Exterior2nd $ MasVnrType
$ MasVnrArea ExterQual $ ExterCond $ Foundation $ BsmtQual $ BsmtCond $
BsmtExposure $ BsmtFinType1 $ BsmtFinSF1 BsmtFinType2 $ BsmtFinSF2
BsmtUnfSF TotalBsmtSF Heating $ HeatingQC $ CentralAir $ Electrical $
firstFlrSF secondFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath
FullBath HalfBath BedroomAbvGr KitchenAbvGr KitchenQual $ TotRmsAbvGrd
Functional $ Fireplaces FireplaceQu $ GarageType $ GarageYrBlt GarageFinish $
GarageCars GarageArea GarageQual $ GarageCond $ PavedDrive $
WoodDeckSF OpenPorchSF EnclosedPorch SsnPorch ScreenPorch PoolArea
PoolQC $ Fence $ MiscFeature $ MiscVal MoSold YrSold SaleType $
SaleCondition $;
run;
/*Create Results Data Set for Custom Model for submission to Kaggle*/
data test;
set test;
/*  missing or NA or -1 values by  */
if GrLivArea=. or GrLivArea<0 then GrLivArea=0;
if TotalBsmtSF=. or TotalBsmtSF<0 then TotalBsmtSF=0;
if BsmtUnfSF=. or BsmtUnfSF<0 then BsmtUnfSF=0;
if GarageArea=. or GarageArea<0 then GarageArea=0;
if LotFrontage = . or LotFrontAge<0 then LotFrontage = 0;
if garageCars=. then garageCars=0;
if GarageYrBlt = . OR GarageYrBlt <0 then GarageYrBlt = 0;
if MasVnrArea<1 or MasVnrArea=. then MasVnrArea=0;
if Fireplaces=. or Fireplaces<0 then Fireplaces=0;
if BsmtFullBath=. or BsmtFullBath<0 then BsmtFullBath=0;
if FullBath=. or FullBath<0 then FullBath=0;
if BsmtHalfBath=. or BsmtHalfBath<0 then BsmtHalfBath=0;
if HalfBath=. or HalfBath<0 then HalfBath=0;
/* Cleanup of String Values */
if Neighborhood = "-1mes" then Neighborhood = "NAmes";
if MasVnrType = "NA" then MasVnrType = "None";
/* Create Property Age variable (PropAge) - Continous variable  */
/* Subtract Year Remodelled or year home built from year sold to provide
relative age of home at time of sale */
if YearRemodAdd>0 then PropAge=yrsold-YearRemodAdd;
else PropAge=yrsold-YearBuilt;
/*Create poolFlag, this indicates if the home has a pool or not*/
poolFlag='N';
if PoolArea>0 then poolFlag='Y';
/*Combine logical square footage variables, perform log transformation on
results, also perform log transformation of SalePrice*/
logSalePrice = log(SalePrice);
logLotArea = log(LotArea);
TotalFinishedSqFt = GrLivArea + TotalBsmtSF - BsmtUnfSF;
if GarageFinish="Fin" or GarageFinish="RFn" then
TotalFinishedSqFt=TotalFinishedSqFt+GarageArea;
logTotalFinSqFt = log(TotalFinishedSqFt);
TotalConSqFt = GrLivArea + TotalBsmtSF + GarageArea;
logTotalConSqFt = log(TotalConSqFt);
TotalOutSF = WoodDeckSF + OpenPorchSF + EnclosedPorch + SsnPorch +
ScreenPorch;
logTotalOutSF = log(TotalOutSF+1);
logLotFrontage = log(LotFrontage+1);
logMasVnrArea=log(MasVnrArea+1);
/* Combine Bath Data into Total Bath */
TotalBath = BsmtFullBath + FullBath + BsmtHalfBath/2 + HalfBath/2;
kitchenQualClass = 0;
if KitchenQual='TA' then kitchenQualClass=1;
if KitchenQual='Ex' then kitchenQualClass=2;
HeatingQCClass = 0;
if HeatingQC='TA' then HeatingQCClass=1;
if HeatingQC='Ex' then HeatingQCClass=2;
OutSFFlag =0;
if TotalOutSF>0 then OutSFFlag=1;
VnrAreaFlag =0;
if MasVnrArea>0 then VnrAreaFlag =1;
drop Alley PoolQC Fence MiscFeature ;
run;
/* Combine the test and Train data */
data train3;
set train1 test;
run;
/* Get the model prediction */
proc glm data = train3 plots = all;
class Neighborhood OverallQual OverallCond SaleCondition HeatingQCClass
kitchenQualClass OutSFFlag ;
model logsaleprice = Fireplaces GarageCars HeatingQCClass  logTotalOutSF
OverallQual OutSFFlag SaleCondition kitchenQualClass OverallCond PropAge
TotalBath loglotarea logTotalConSqFt Neighborhood
logTotalConSqFt*Neighborhood/ tolerance CLI solution;
output out = results p = Predict;
run;
/* Create the CSV file to be uploaded to Kaggle */
data resultsStats2;
set results;
SalePrice = exp(Predict);
keep Id SalePrice;
where Id > 1460;
proc export data=resultsStats2
 outfile='/home/harisanadhya0/sasuser.v94/MSDS 6372/Project
1/output_model_test.csv'
 dbms=csv
 replace; run;
```