

Kaggle Name: Andrew Cowan-Nagora

Competition: House Prices: Advanced Regression Techniques

<https://www.kaggle.com/c/house-prices-advanced-regression-techniques>

Teams: 4370

Rank: 1348 (Top 31%)

1-- House Prices: Advanced Regression Techniques

- Predict sales prices and practice feature engineering, RFs, and gradient boosting

- 4356 Teams

2 -- Data Exploration and Cleaning

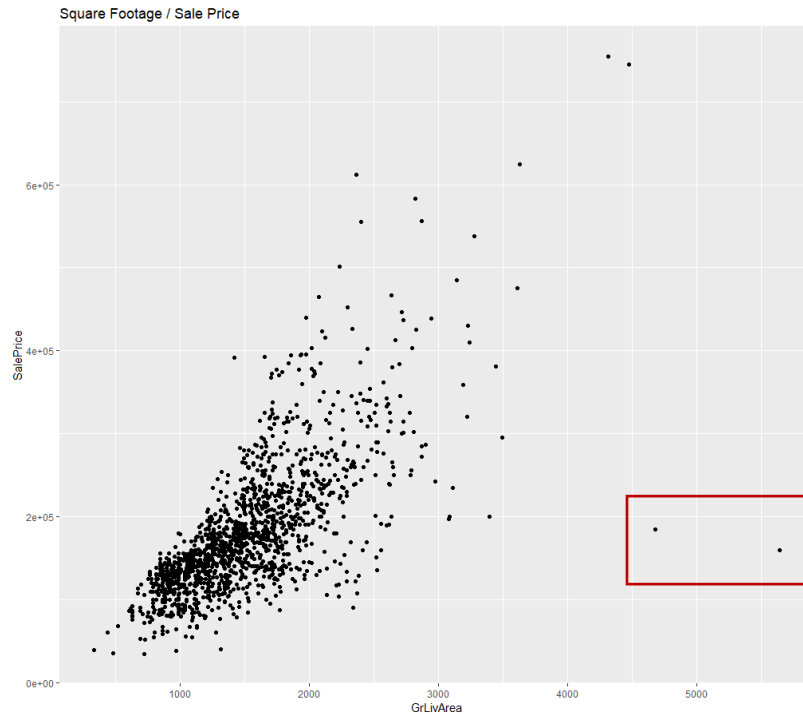
Initial exploratory analysis of the housing prices dataset revealed that a series of cleaning and feature engineering steps would be required in order run optimal regression models.

The three primary challenges that need to be addressed are **outliers/skewed values**, **missing data** and **modifying multiple categorical class variables**.

Outliers – Square Footage

Reviewing the data description and getting an understanding of what each column represents leads to some initial hypotheses about which features are likely to be strong predictors of home values.

Square footage would seem to be important in that larger houses would be more expensive so a scatter plot was built to see if this was the case:

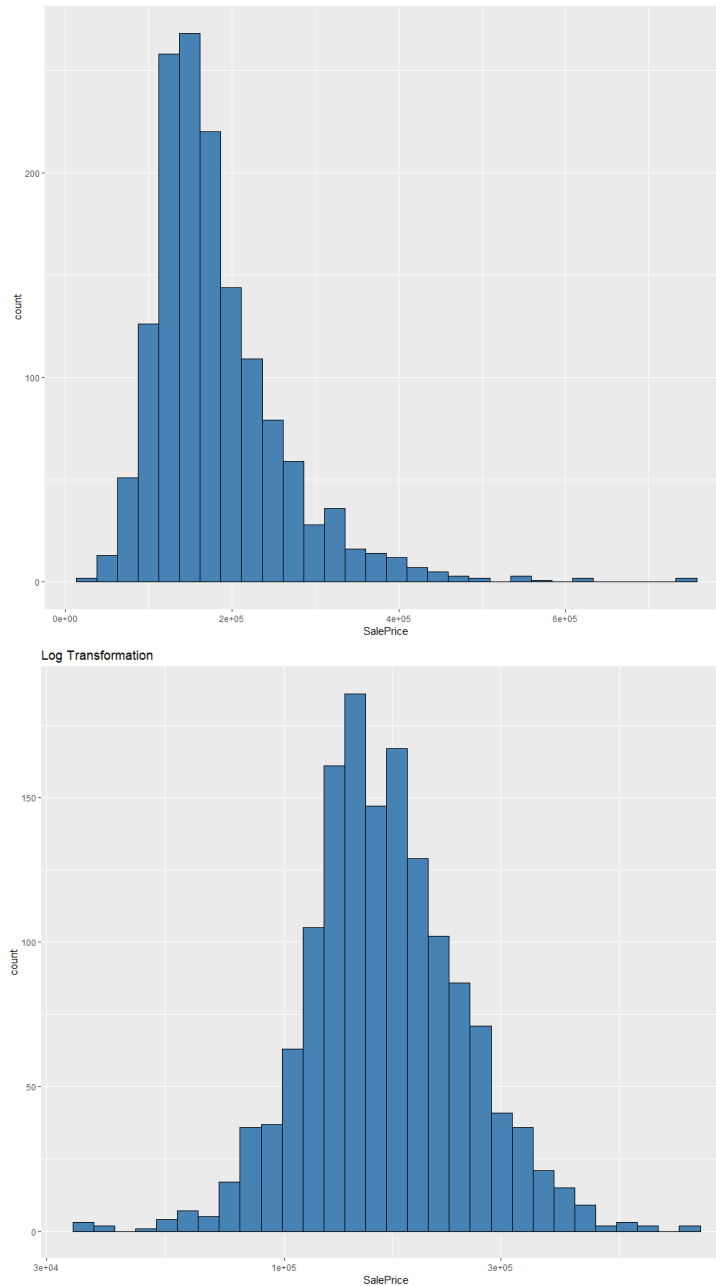


There is clearly a positive relationship but the plot reveals two of the largest homes as having a relatively low sale price. These cases probably had extenuating circumstances or are simply data errors but they will need to be dealt with in order to optimize modelling. The first approach involved a log transformation while the second substituted the values with mean.

Distribution – Sale Price

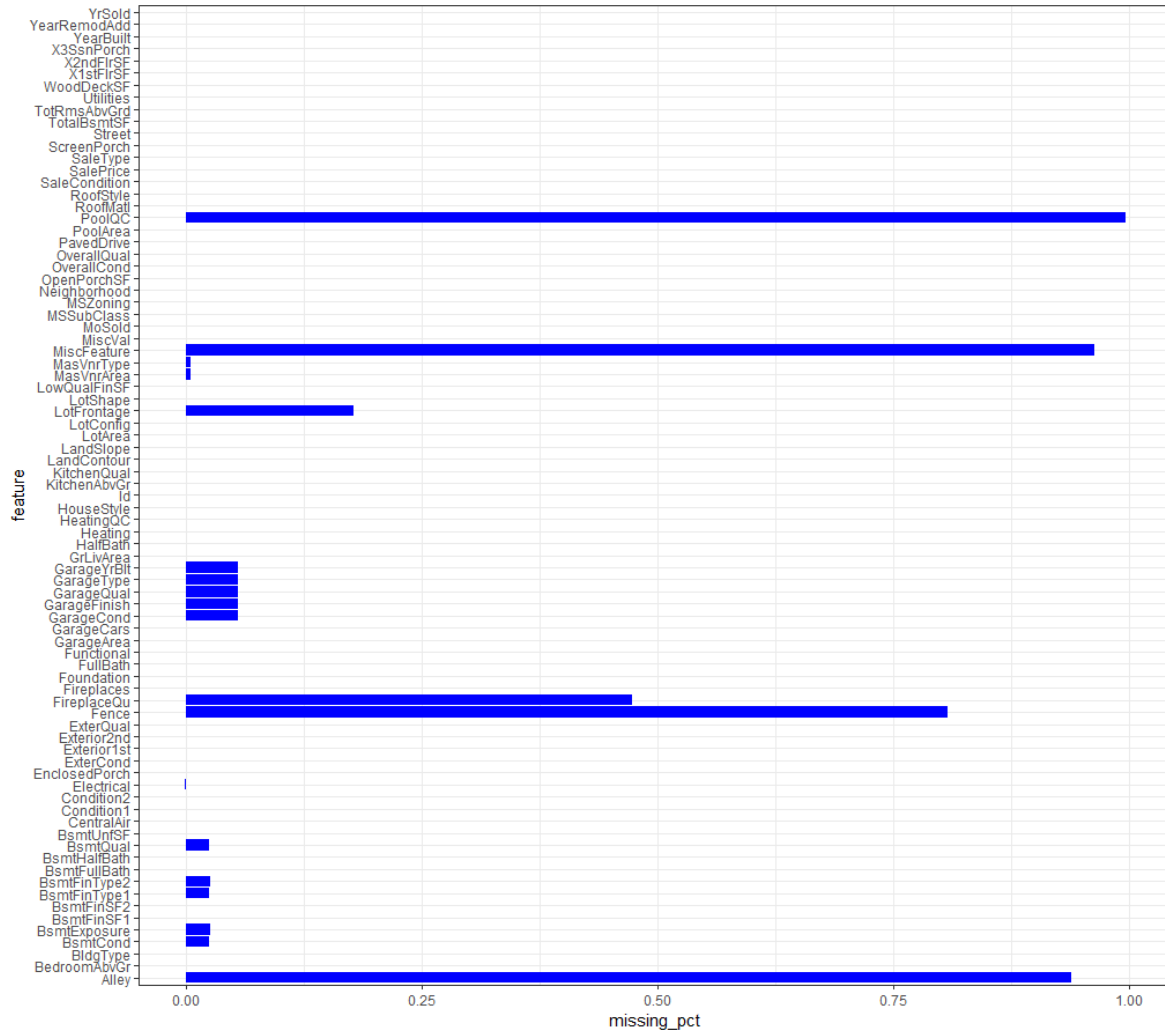
As sale price is the metric being predicted, examining the distribution in the training set is of critical importance in assessing its suitability for regression.

The previous scatter plot shows that a smaller number of houses have much higher values and a histogram confirms a right skewed distribution. This can be managed by performing a log transformation which will reduce the negative impact on model performance. The same method is used on other variables with high skewness.



Missing Values

Missing values in both numeric and categorical variables is also a significant issue that needs to be addressed before modelling can take place. The below plot shows the percentage of missing values for each feature.



For the numerical variables there are two options. The first is based on the assumption that they are missing at random so median or mean substitution can be used to generate estimated values.

The second option involves looking closely at each column and deciding that the values are not missing but instead actually zero. For example, missing values in lot frontage can be assumed to indicate that there is no street space on the property. In this case a zero should be substituted in rather than the median or mean.

This approach is also applied to the missing categorical values. Fireplace quality for instance has different class levels so it is likely that missing values indicate that no fireplace is present in the home. In these cases, the NA levels are replaced with a new class called none.

Feature Engineering

Additional techniques were applied to the categorial features in later modeling stages in an attempt to improve performance.

In many cases the class variables indicate a quality order. These were recoded into factors as pseudo continuous numerical sequences which for example converted basement condition from none, poor, fair, normal, good and excellent to values of 0,1,2,3,4,5.

Categorical features that do not imply an order such as zoning classification (A, C, FC, etc.) were changed into dummy variables of 1 and 0 which expanded the number of columns in the dataset to 242.

3 -- Regression Models

With the primary challenges and approaches to handle them now addressed, the specific process and methodology behind each model is outlined below.

Model 1: Multiple Linear Regression (Limited Prep)

- Step 1:** Replace missing numerical values with median
- Step 2:** Replace missing categorical values with new class 'not available'
- Step 3:** Perform log transformation on sale price and square footage
- Step 4:** Convert quality character levels into numerical sequences
- Step 5:** Run standard linear regression model on selected features (Appendix: Model 1)
- Step 6:** Predict against test data, assess accuracy, submit results

RMSE: 0.15662

Kaggle Rank: 2984 (Top 70%)

Model 2: Multiple Linear Regression (Additional Prep)

- Step 1:** Replace square footage outliers with mean
- Step 2:** Perform log transformation on sale price
- Step 3:** Substitute numerical missing values with zeros
- Step 4:** Recode quality character levels into numerical sequences
- Step 5:** Convert non-ordered character values into dummy variables
- Step 6:** Perform log transformation on other heavily skewed variables
- Step 7:** Run standard linear regression model on all features (Appendix: Model 2)
- Step 6:** Transform price back to exponent, predict against test data, assess accuracy, submit results

RMSE: 0.13401

Kaggle Rank: 1960 (Top 46%)

Model 3: Lasso Regularized Regression

Same steps as above but with a lasso model.

- Step 1:** Replace square footage outliers with mean
- Step 2:** Perform log transformation on sale price
- Step 3:** Substitute numerical missing values with zeros
- Step 4:** Recode quality character levels into numerical sequences
- Step 5:** Convert non-ordered character values into dummy variables
- Step 6:** Perform log transformation on other heavily skewed variables
- Step 7:** Run lasso model (Appendix: Model 3)
- Step 8:** Transform price back to exponent, predict against test data, assess accuracy, submit results

RMSE: 0.12600

Kaggle Rank: Top 35%

Model 4: Lasso Regularized Regression (Alternate Prep)

- Step 1:** Perform log transformation on sale price
- Step 2:** Perform log transformation on other heavily skewed variables
- Step 3:** Convert character values into dummy variables
- Step 4:** Set NA categorical levels to zero
- Step 5:** Replace missing numeric values with the mean of that feature
- Step 6:** Run lasso model (Appendix: Model 4)
- Step 7:** Transform price back to exponent, predict against test data, assess accuracy, submit results

RMSE: 0.12391

Kaggle Rank: Rank 1348 (Top 31%)

Conclusion

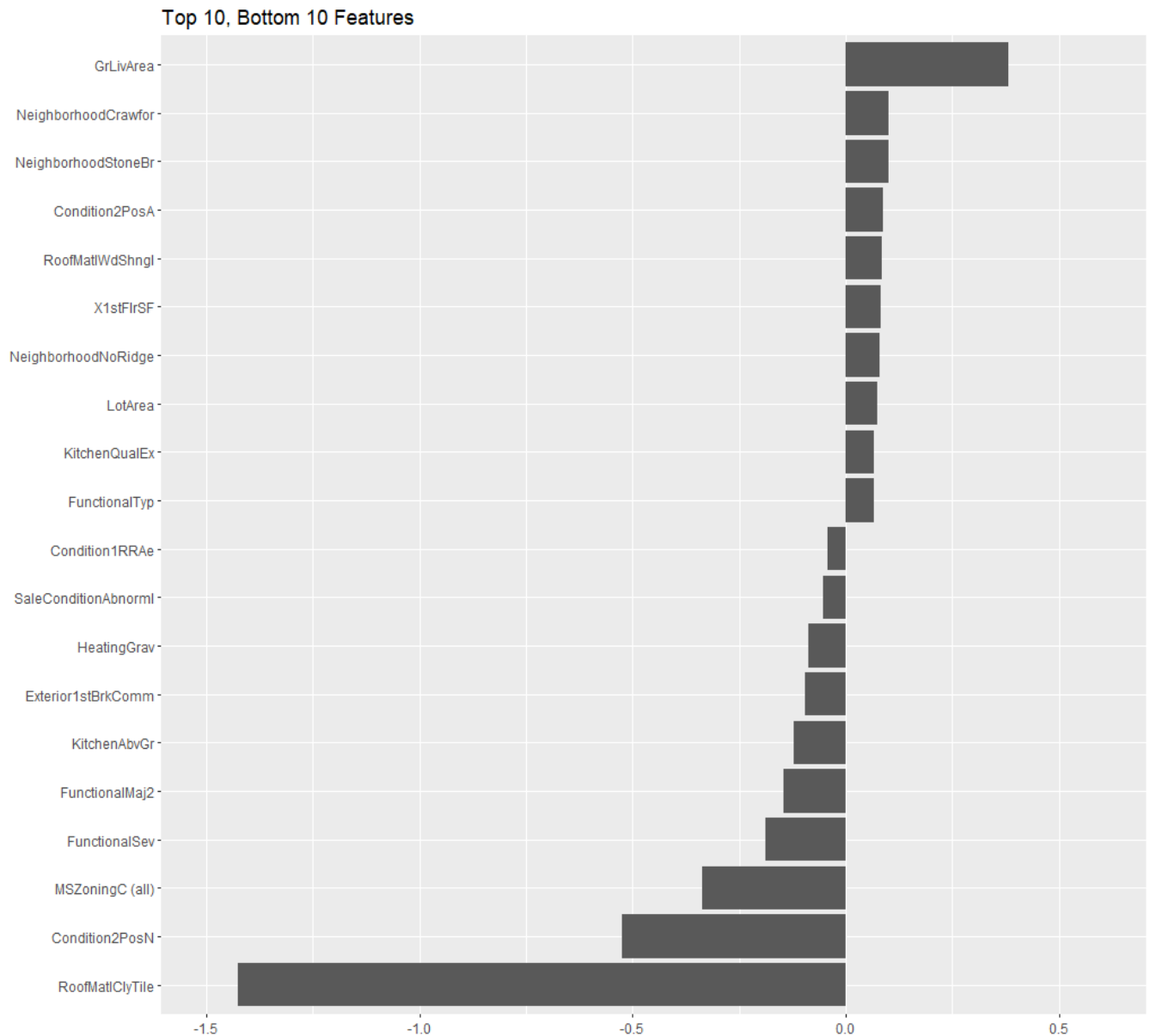
The lasso models performed better than standard multiple regression.

In model 3 the missing numerical values were substituted with zeros and ordered quality levels were converted to numeric sequences whereas in model 4 missing numerical values were substituted with the mean and all categorical variables were changed to dummies regardless of ordered sequence.

Both approaches produced a very close RMSE, but model 4 was ultimately the best.

The top 10 and bottom 10 features selected by the lasso model are shown in the graph below.

As expected, square footage was the most important predictor of sale price. Location in specific neighbourhoods was the second most important factor while being adjacent to a positive off site feature and having a wood shingle roof also were significant.




Appendix

Competition: House Prices: Advanced Regression Techniques

Link: <https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview>

Data description included as text file


Kaggle Rank: 1348 (Top 31%)


1348	Andrew Cowan-Nagora		0.12391	4	~10s
------	---------------------	---	---------	---	------

Your Best Entry ↑

You advanced 162 places on the leaderboard!

Your submission scored 0.12391, which is an improvement of your previous score of 0.12600. Great job!

 [Tweet this!](#)




Andrew Cowan-Nagora


Joined 4 days ago · last seen in the past day

[Home](#) [Competitions \(1\)](#) [Kernels](#) [Discussion](#) [Datasets](#) [...](#)


Competitions Novice




Unranked



0




0




0

Kernels Novice

Unranked



0



0

House Prices: Advanced Re...

Ongoing-Top 31%

1,348th
of 4352

Model 1 `> model <- lm(LogPrice ~ OverallQual + Neighborhood +
LogGrLivArea + ExterQual + KitchenQual + Age + GarageCars +
TotalBsmtSF + X1stFlrSF + GarageArea, data = features[1:nrow(train),])`

Model 2

`> mlr <- lm(formula = SalePrice ~., data = train_dummy)`

Model 3

`> lasso <- cv.glmnet(as.matrix(train_dummy[, -241]), train_dummy[,
241])`

Model 4

```
> model_lasso <- train(x=X_train,y=y,  
+                       method="glmnet",  
+                       metric="RMSE",  
+                       maximize=FALSE,  
+                       trControl=CARET.TRAIN.CTRL,  
+                       tuneGrid=expand.grid(alpha=1, # Lasso regression  
+                                           lambda=c(1,0.1,0.05,0.01,seq(0.00  
9,0.001,-0.001),  
+                                           0.00075,0.0005,0.0001)))
```

```
> model_lasso  
glmnet
```

```
1460 samples  
288 predictor
```

No pre-processing

Resampling: Cross-Validated (5 fold, repeated 5 times)

Summary of sample sizes: 1169, 1168, 1167, 1167, 1169, 1168, ...

Resampling results across tuning parameters:

lambda	RMSE	Rsquared	MAE
0.00010	0.1364085	0.8849838	0.08588783
0.00050	0.1315241	0.8921941	0.08329824
0.00075	0.1304120	0.8938013	0.08239250
0.00100	0.1296367	0.8949062	0.08170942
0.00200	0.1274528	0.8980568	0.07990677
0.00300	0.1263227	0.8997518	0.07930003
0.00400	0.1260769	0.9001275	0.07953155
0.00500	0.1262120	0.8999958	0.08014016
0.00600	0.1265722	0.8995779	0.08090338
0.00700	0.1270563	0.8990182	0.08172068
0.00800	0.1277540	0.8981586	0.08262832
0.00900	0.1286558	0.8970003	0.08362045

0.01000	0.1296635	0.8956889	0.08465663
0.05000	0.1729848	0.8397762	0.12102571
0.10000	0.2156593	0.7948456	0.15744685
1.00000	0.3991753	NaN	0.30981341

Tuning parameter 'alpha' was held constant at a value of 1
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were $\alpha = 1$ and $\lambda = 0.004$.