Predicting House Prices with Gradient Boosting

Neil Kutty

May 2, 2017

Abstract

Predicting house prices based on features of the property presents a unique challenge. Some properties sell for an expected price and are predictable with high accuracy given simply aspects of the property itself (size, age, quality). However; some properties sell for dollar amounts that are significantly different from what was expected based solely on the physical measures. We find these outliers present a challenge to having a very high accuracy in prediction due to properties regarding the transaction itself (partial sale, foreclosure) that may not fully capture the difference in expected outcome versus actual.

1 Introduction

The data for this project has 80 native features and an outcome variable for the dollar amount a house sold for. Prior research (see: De Cock, Dean, Journal of Statistics Education Volume 19, Number 3(2011) https://ww2.amstat.org/publications/jse/v19n3/decock.pdf) conducted on this dataset has shown low accuracy 70% for OLS Linear models. We seek to improve on this prior research by using Gradient Boosting, Spectral Clustering for feature derivation, and Principal Component Analysis for feature selection.

The features are a mix of categorical and continuous variables. Around 7000 values are NA out of a total of approximately 111,300 values. We forego utilizing dummy variables for this iteration in order to perfectly match the test set's feature count.

Our best model is through gradient boosting with 250 estimators resulting in a 91.67% accuracy rate when using the top 75 features including derived features obtained through unsupervised learning and then ranked by PCA.

2 Data Exploration

The data for this project contains details related to the sale of a residential property in Ames, Iowa. It contains 2930 observations, and 82 variables (one ID variable that is discarded for modeling, and the outcome variable leaving 80 native predictors).

The following is taken from the data description website (see citation for source). *DESCRIPTIVE ABSTRACT: Data set contains information from the Ames Assessor's Office used in computing assessed values for individual residential properties sold in Ames, IA from 2006 to 2010. SOURCES: Ames, Iowa Assessor's Office. VARIABLE DESCRIPTIONS: Tab characters are used to separate variables in the data file. The data has 82 columns which include 23 nominal, 23 ordinal, 14 discrete, and 20 continuous variables (and 2 additional observation identifiers).*

2.1 Data Cleaning and Processing

The dataset has a mix of continuous/discrete and categorical variables. For expediency we fill NA values with zero as this addresses both the categorical and numerical missing-value identification for modeling. In order to enumerate the entire dataset, we pre-process by separating the numeric and non-numeric columns, retrieve and assign the category codes for the non-numeric columns back to them, and then re-join the full dataset. We derive columns for 'TotalSF', and number of years since the native 'Yr' variables.

2.2 Correlating Predictors to Outcome

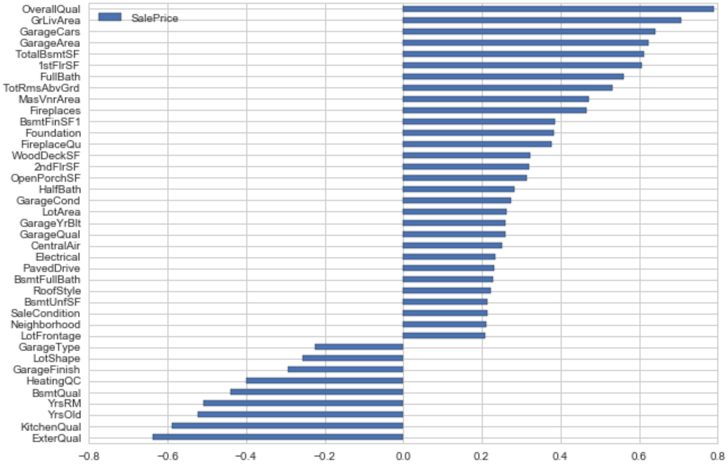
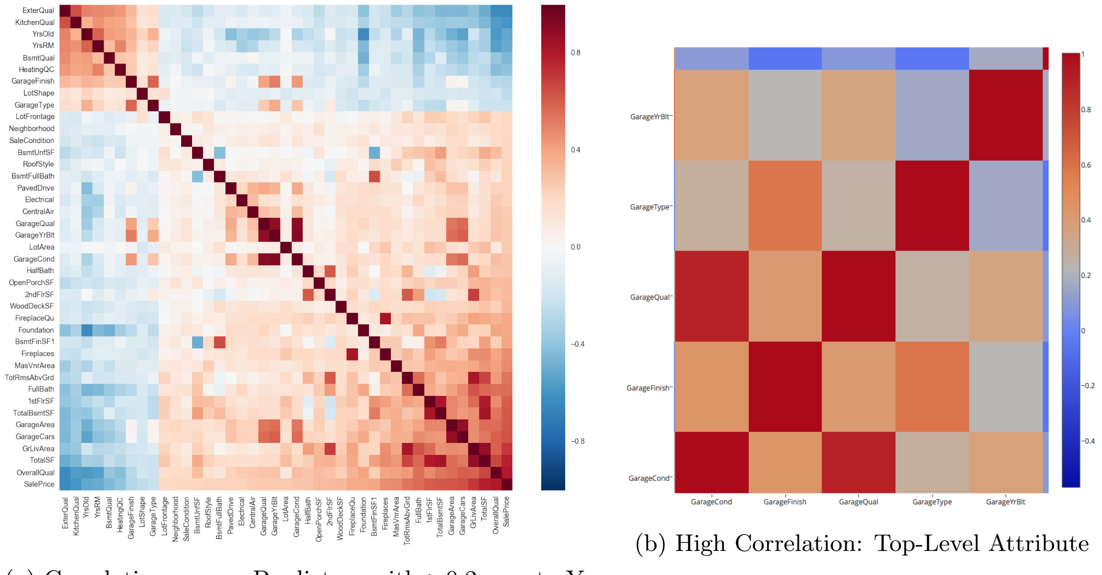


Figure 1: Correlation of Predictors to Outcome where coef > 0.2.

As expected, the size of the house is the top most highly correlated predictor to it's sale price. We also observe that the 'OverallQual' predictor, an ordinal ranking of the overall quality of the property, ranks highly. The high negative correlation of 'KitchenQual' and 'ExterQual' is due to their inverted scales where a low value 1 indicates high quality and a higher value like 5 indicates lower quality.

2.3 Correlations Among Predictors

Looking at correlations among the different predictor variables...



(a) Correlation among Predictors with >0.2 corr to Y

(b) High Correlation: Top-Level Attribute

Figure 2: Visualize Multicollinearity Among Predictors

The presence of near zero correlation across the dataset indicates that we have a a fairly healthy dataset in terms of generally low multicollinearity. Observed pockets of highly correlated predictor variables generally are associated with individual predictors that all describe a common top-level attribute of the property (i.e. Garage vars, Basement vars) (see Figure 2 (b)). For future iterations of this project, it would be worthwhile to explore engineering core derived features for these top-level attributes that combine these native features.

2.4 Outliers

Some outliers, like the one below, have a low price for "higher-price" oriented features seemingly captured exclusively in the SaleCondition predictor variable. However, the value for SaleCondition ('Partial') in this case may be associated with higher variability in associated outcome value, thus simply relying on this variable for an accurate quantitative outcome may not be fruitful.

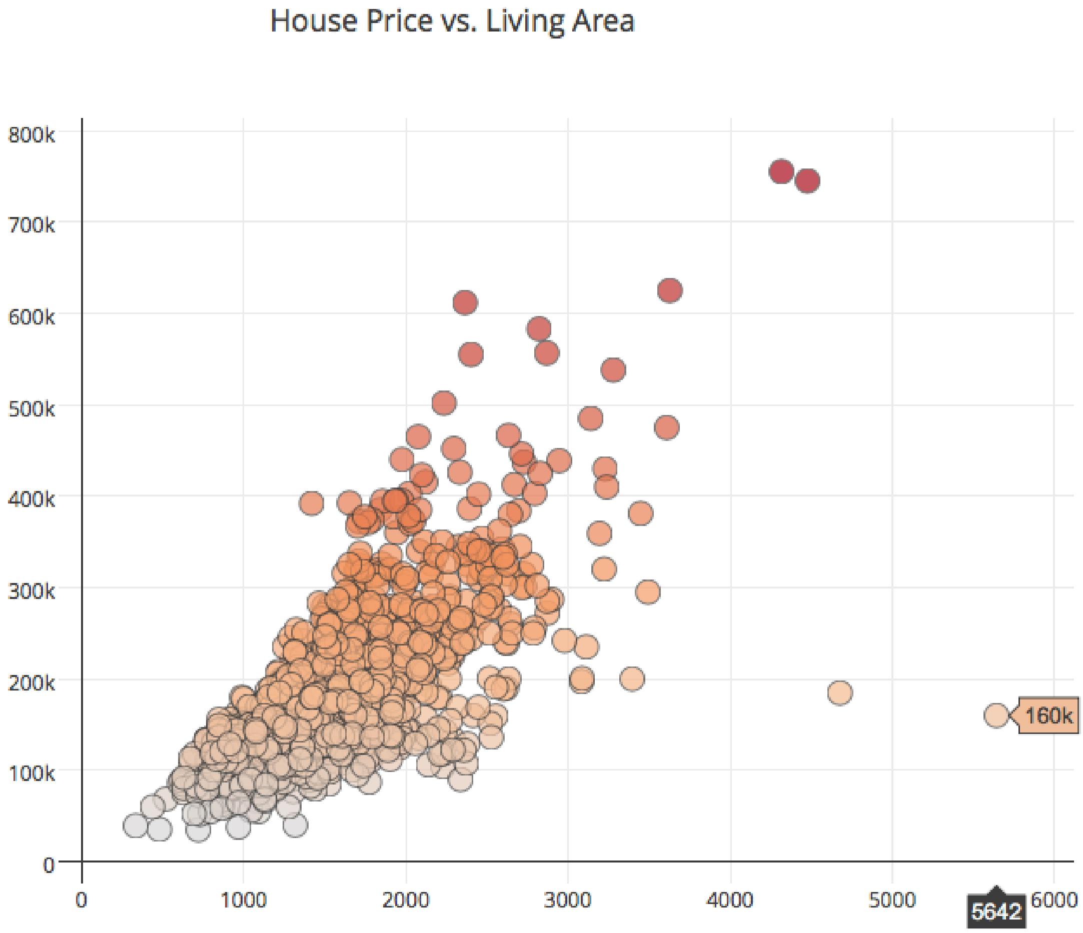


Figure 3: Large House, Low Price

GrLivArea (SqFt above ground)	Sale Price ($)	Sale Condition
5,642	$160,000	Partial

Table 1: Outlier: large house, low price

We actually try utilizing a dummy variable for 'SaleCondition' however the results are a negligible flat increase in accuracy. This is likely due to the fact, as posited above, that this value does not organically capture the actual reason for the house selling at the lower price it sold at. A 'foreclosure' or 'partial' value for this variable does not necessarily indicate a definite distance between expected sale (based on property features) and actual dollar sale (i.g. a partial sale for an expected $500k house does not necessarily mean it will actually sell at $160k. It may sell for $80k, $20k, $300k, or another different value).

This outlier dynamic in the dataset, we observe, is likely responsible for our inability to break above an approximately 90% accuracy rate in our model training below.

3 Unsupervised Learning

We utilize Spectral Clustering in order to identify possibly robust features in an attempt to improve our model accuracy.

3.1 Spectral Clustering

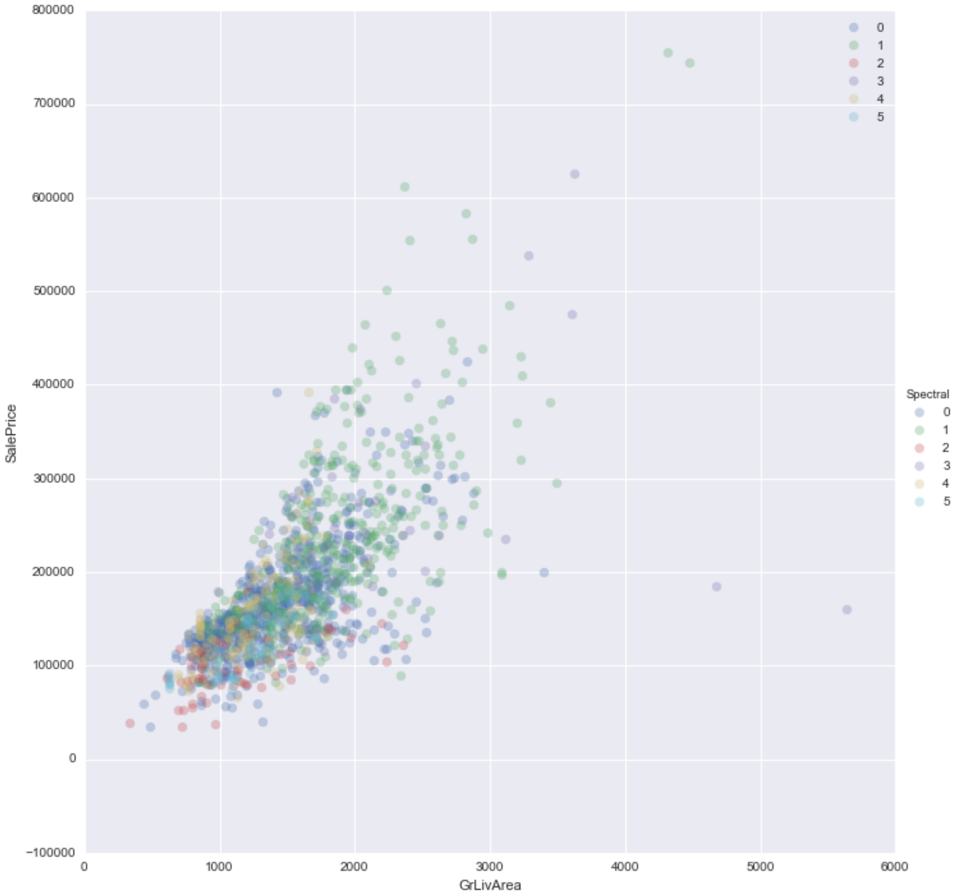


Figure 4: Spectral Clustering Results for 6 Clusters

We define a function below to fit a Spectral Clustering model on the predictors for a given number of n_clusters and return the resulting cluster labels. The determined cluster assignment of each record, for the given clustering fit, is then added to the dataset as a new feature. We perform this feature derivation for six different iterations of cluster numbers (6, 10, 12, 14, 18, & 24).

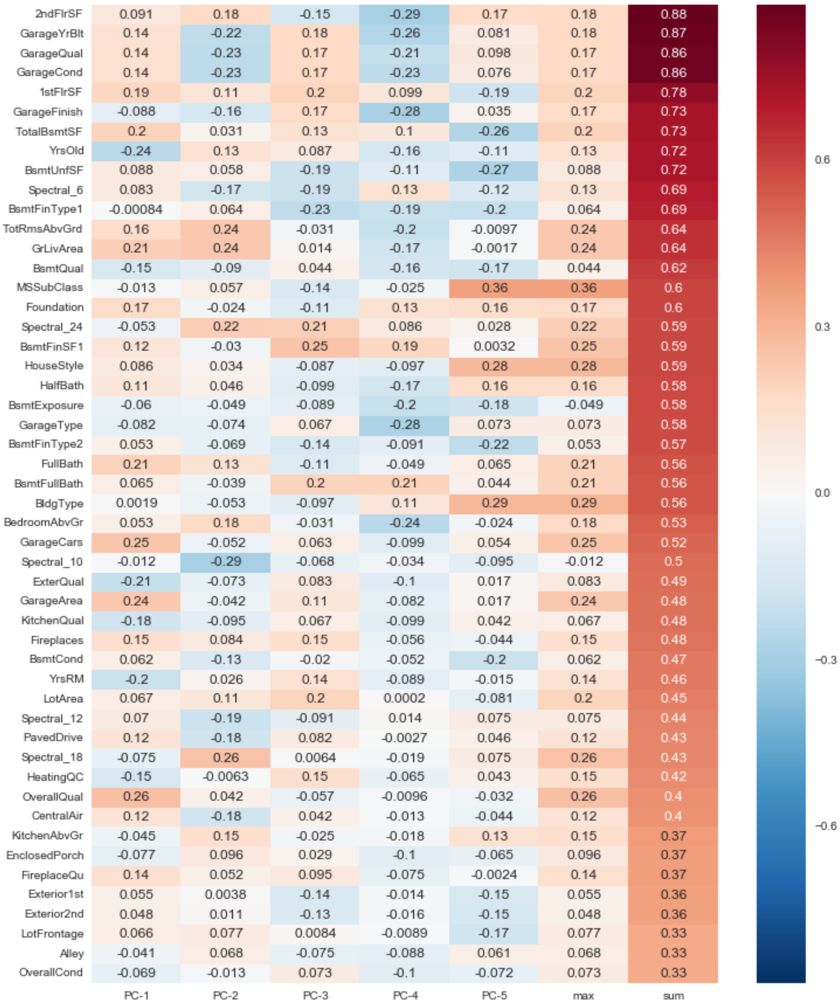
```
from sklearn.cluster import SpectralClustering

def spectralFeatureAppend(clusters, data):
    sc = SpectralClustering(n_clusters=clusters,
                            eigen_solver='arpack',
                            affinity="rbf")
    sc.fit(np.array(data.drop('SalePrice',axis=1)))
    labels=sc.labels_.astype(np.int)
    return labels
```

The newly derived Spectral Clustering features are then appended to the dataset increasing the feature count by six.

4 Principal Components Analysis

Principal Components Analysis gives us vectors of coefficients (eigenvectors) which convey significance of those variables for a particular component. We perform PCA for five components on the dataset resulting in a set of eigenvectors of which the absolute value sum is obtained. The feature ranking based on the sum of this series becomes a sorted list of features with which we use to eliminate features in the model fitting performed below.



	PC-1	PC-2	PC-3	PC-4	PC-5	max	sum
2ndFlrSF	0.091	0.18	-0.15	-0.29	0.17	0.18	0.88
GarageYrBlt	0.14	-0.22	0.18	-0.26	0.081	0.18	0.87
GarageQual	0.14	-0.23	0.17	-0.21	0.098	0.17	0.86
GarageCond	0.14	-0.23	0.17	-0.23	0.076	0.17	0.86
1stFlrSF	0.19	0.11	0.2	0.099	-0.19	0.2	0.78
GarageFinish	-0.088	-0.16	0.17	-0.28	0.035	0.17	0.73
TotalBsmtSF	0.2	0.031	0.13	0.1	-0.26	0.2	0.73
YrsOld	-0.24	0.13	0.087	-0.16	-0.11	0.13	0.72
BsmtUnfSF	0.088	0.058	-0.19	-0.11	-0.27	0.088	0.72
Spectral_6	0.083	-0.17	-0.19	0.13	-0.12	0.13	0.69
BsmtFinType1	-0.00084	0.064	-0.23	-0.19	-0.2	0.064	0.69
TotRmsAbvGrd	0.16	0.24	-0.031	-0.2	-0.0097	0.24	0.64
GrLivArea	0.21	0.24	0.014	-0.17	-0.0017	0.24	0.64
BsmtQual	-0.15	-0.09	0.044	-0.16	-0.17	0.044	0.62
MSSubClass	-0.013	0.057	-0.14	-0.025	0.36	0.36	0.6
Foundation	0.17	-0.024	-0.11	0.13	0.16	0.17	0.6
Spectral_24	-0.053	0.22	0.21	0.086	0.028	0.22	0.59
BsmtFinSF1	0.12	-0.03	0.25	0.19	0.0032	0.25	0.59
HouseStyle	0.086	0.034	-0.087	-0.097	0.28	0.28	0.59
HalfBath	0.11	0.046	-0.099	-0.17	0.16	0.16	0.58
BsmtExposure	-0.06	-0.049	-0.089	-0.2	-0.18	-0.049	0.58
GarageType	-0.082	-0.074	0.067	-0.28	0.073	0.073	0.58
BsmtFinType2	0.053	-0.069	-0.14	-0.091	-0.22	0.053	0.57
FullBath	0.21	0.13	-0.11	-0.049	0.065	0.21	0.56
BsmtFullBath	0.065	-0.039	0.2	0.21	0.044	0.21	0.56
BldgType	0.0019	-0.053	-0.097	0.11	0.29	0.29	0.56
BedroomAbvGr	0.053	0.18	-0.031	-0.24	-0.024	0.18	0.53
GarageCars	0.25	-0.052	0.063	-0.099	0.054	0.25	0.52
Spectral_10	-0.012	-0.29	-0.068	-0.034	-0.095	-0.012	0.5
ExterQual	-0.21	-0.073	0.083	-0.1	0.017	0.083	0.49
GarageArea	0.24	-0.042	0.11	-0.082	0.017	0.24	0.48
KitchenQual	-0.18	-0.095	0.067	-0.099	0.042	0.067	0.48
Fireplaces	0.15	0.084	0.15	-0.056	-0.044	0.15	0.48
BsmtCond	0.062	-0.13	-0.02	-0.052	-0.2	0.062	0.47
YrsRM	-0.2	0.026	0.14	-0.089	-0.015	0.14	0.46
LotArea	0.067	0.11	0.2	0.0002	-0.081	0.2	0.45
Spectral_12	0.07	-0.19	-0.091	0.014	0.075	0.075	0.44
PavedDrive	0.12	-0.18	0.082	-0.0027	0.046	0.12	0.43
Spectral_18	-0.075	0.26	0.0064	-0.019	0.075	0.26	0.43
HeatingQC	-0.15	-0.0063	0.15	-0.065	0.043	0.15	0.42
OverallQual	0.26	0.042	-0.057	-0.0096	-0.032	0.26	0.4
CentralAir	0.12	-0.18	0.042	-0.013	-0.044	0.12	0.4
KitchenAbvGr	-0.045	0.15	-0.025	-0.018	0.13	0.15	0.37
EnclosedPorch	-0.077	0.096	0.029	-0.1	-0.065	0.096	0.37
FireplaceQu	0.14	0.052	0.095	-0.075	-0.0024	0.14	0.37
Exterior1st	0.055	0.0038	-0.14	-0.014	-0.15	0.055	0.36
Exterior2nd	0.048	0.011	-0.13	-0.016	-0.15	0.048	0.36
LotFrontage	0.066	0.077	0.0084	-0.0089	-0.17	0.077	0.33
Alley	-0.041	0.068	-0.075	-0.088	0.061	0.068	0.33
OverallCond	-0.069	-0.013	0.073	-0.1	-0.072	0.073	0.33

Figure 5: Principal Components Analysis: Top 50 Features
by Absolute Value Sum of Eigenvalue

According to the PCA results, the top significant feature (across all five components) is ’2ndFlrSF’, the area in square feet of the 2nd floor of a residential property. ’1stFlorSF’ does not rank far behind at Rank #5; however, ’GrLivArea’ (total above ground square feet) Ranks at #13 which is significantly lower than the ranking by Pearson correlation. Furthermore, more of the highest ranking variables by PCA (besides area of the 1st and 2nd floors) concern the top-level Garage attribute compared to correlation, although there is some representation of Garage attribute measures in the Pearson results.

While there is some confirmation of our correlation analysis, the observed divergence conveys that dynamics not fully captured in simple Pearson correlation are being reflected in the Principal Components. The addition of Spectral Clustering derived features may also be contributing to this observed distinction between PCA and simple correlation.

5 Model Training

5.1 Gradient Boosting

```
GradientBoostingRegressor(alpha=0.9, criterion='friedman_mse', init=None,
             learning_rate=0.1, loss='ls', max_depth=3, max_features=None,
             max_leaf_nodes=None, min_impurity_split=1e-07,
             min_samples_leaf=1, min_samples_split=2,
             min_weight_fraction_leaf=0.0, n_estimators=250,
             presort='auto', random_state=1010, subsample=1.0, verbose=0,
             warm_start=False)
```

The first regression we perform is a Gradient Boosting Regression using 250 estimators and including all native predictor variables without including the Spectral Clustering label results. This model fit results in a 90.9% prediction accuracy rate with a high cross-validation score of 92.35% and a low cv of 58.81% for 10 cross-validation iterations.

Table 2: Gradient Boosting: Native Features

Accuracy	Best CV	Worst CV
90.90%	92.35%	58.81%

In our next regression, we include all of the native columns as well as the newly derived Spectral Clustering labels as features. The result is an increase in accuracy to 91.35%, with improvement to 93.66% for the high cross-validation as well as to 59.40% for the low cv. This improvement in accuracy is entirely attributable to the addition of new features based on the spectral clustering labels obtained through unsupervised learning above.

Table 3: Gradient Boosting: Native + Spectral Features

Accuracy	Best CV	Worst CV
91.35%	93.66%	59.40%

Now we eliminate the lower ranking features as determined by Principal Components Analysis. We take the top 75 predictors thereby eliminating the bottom 10 PCA scored features. This results in an even greater improvement in accuracy to 91.67%, with a flat change in high cv to 93.59% and an increase in low cross-validation accuracy to 60.08%. This is our best performing model thus far; furthermore, we validate our use of unsupervised learning for feature derivation as well as the ranking of our Principal Components Analysis scores.

Table 4: Gradient Boosting: Top 75 PCA Features

Accuracy	Best CV	Worst CV
91.67%	93.59%	60.08%

5.2 Random Forest Regressor

Fitting a Random Forest Regressor to the dataset (including Spectral Clustering derived features) gives similar albeit slightly lower accuracy; however, this model fit gives a better cross-validation score for the lower cv accuracy measure versus Gradient Boosting.

Table 5: Random Forest Regressor: Native + Spectral Features

Accuracy	Best CV	Worst CV
89.80%	92.62%	64.81%

5.3 Lasso Regression

We implement a Lasso Regression to test a more simplistic model as a benchmark. The results are lower accuracy with similar to slightly lower cross-validation scores.

Table 6: Lasso Regression: Native + Spectral Features

Accuracy	Best CV	Worst CV
85.34%	90.86%	55.65%

5.4 MLP Regressor

For posterity, we also fit a Multilayer Perceptron Regressor to the data but get wildly varying results. Although the model accuracy is competitive with Gradient Boosting, we are getting extreme values for cross-validation and a noncompetitive high cv score value.

Table 7: MLP Regression: Native + Spectral Features

Accuracy	Best CV	Worst CV
91.04%	80.94%	-4.37%

5.5 Random Forest Classifier

Finally, we perform binning using pandas native cut() function, and then run our data through a Random Forest classifier. The results are actually a healthier expected out-of-sample error through cross-validation albeit with a lower initial accuracy. However, as we are attempting to predict a continuous outcome, the pursuit of classification in this case may not be a fruitful endeavor.

Table 8: Random Forest Classifier: Native + Spectral Features

Accuracy	Best CV	Worst CV
85.21%	90.91%	85.45%

6 Conclusions and Remarks for Future Research

Utilizing Spectral Clustering for feature derivation organically improves our model accuracy when Gradient Boosting with 250 estimators. Dimensionality reduction with PCA does not yield us significantly greater accuracy although there is marginal improvement, we should turn to advanced feature engineering. It may prove worthwhile to try modeling with dummy variables while being aware of overfitting and test set distinction. Given the high multicollinearity across features concerned with a top-level property attribute, it would be of interest to derive features to represent core property attributes (Bsmnt, Garage) rather than relying on multiple measures for each one. Addressing outliers may prove the most beneficial for improvement in accuracy and breaking through the 90% resistance level. The most robust potential may be through deriving or otherwise acquiring more dimensionality on features regarding the transaction itself.

It would also be of interest to explore a combination of predictor models (GBR, Lasso, Ridge, SGDR, ElasticNet) for greater accuracy rather than relying on one model alone.

7 Citations

https://ww2.amstat.org/publications/jse/v19n3/decock/DataDocumentation.txt