Ames Housing Dataset Analysis

Slade
CS 5850
Utah State University
Logan, United States
jaaronslade@gmail.com

***Abstract*—This paper covers an analysis of the Ames Housing dataset. Important features of the dataset were discovered, engineered, and analyzed to create a predictive regression model for the dataset. The results of the analysis give insight into the most useful features of the dataset for housing price prediction.**

I. Introduction

This report will detail an analysis of the Ames housing dataset. Many revisions of the analysis were conducted which led to the engineering of more features and further data preprocessing. In a previous course I had attempted to perform a regression of the data using a Ridge classifier, but due to limited experience, little-to-no data preprocessing, feature selection, and feature engineering, my team and I were relatively unsuccessful. This new analysis yielded a coefficient of determination over 0.9178 and and provided valuable insight into the dataset. A major challenge experienced was the time needed to preprocess the data due to the large number of features offered by the dataset.

II. Method

A. Data Source

I sourced the data from Kaggle for the competition: House Prices - Advanced Regression Techniques.

B. Data Preprocessing - Overview

In the course of my analysis, I underwent several iterations of preprocessing and training. I began with data visualization and plotted histograms of each feature. In fig 1 are some histograms of features highly correlated with SalePrice. Figure 2 shows scatterplots of some of the highly correlated features.

C. Data Preprocessing - Outlier Detection and Removal

After examining the histograms for the numerical features I saw that none of the features had data spanning many orders of magnitude. This led me to choose standard scaler to scale all numerical features. In fig 2 it can be seen in the plot of GrLivArea vs SalePrice that there are two datapoints that lie far outside the trend.

If we take a deeper look into the data, we see that there are 4 records total for houses with GrLivArea $\geq$ 4000. Two of the houses sold for over \$700,000. The two apparent outliers sold for under \$200,000 even though they both had overall quality ratings of 10 and more than 11 rooms above ground. Looking at the plot of TotBsmtSF vs SalePrice, there appears to be one data point that is outside the trend. When we inspect

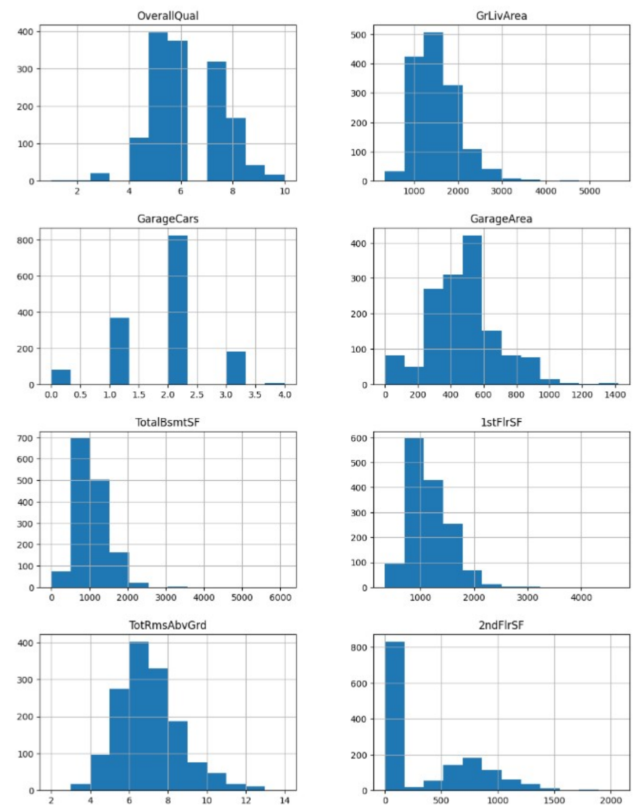

Fig. 1. Histograms of highly correlated features

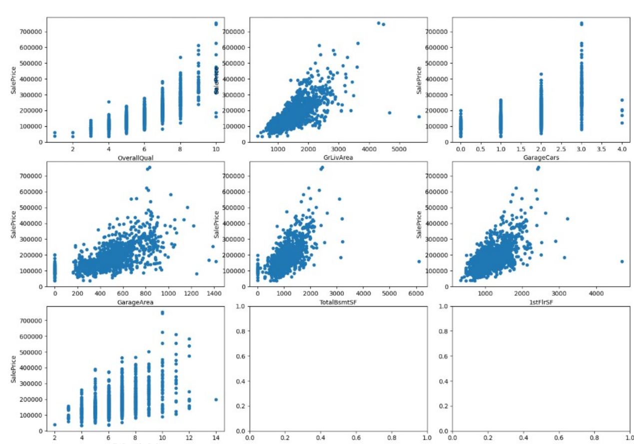

Fig. 2. Scatterplots of highly correlated features

it, it turns out that it is one of the same records that didn't fit the trend for GrLivArea and SalePrice. I chose to remove the two records that did not fit the trend for GrLivArea and SalePrice (records 1299 and 524).

D. Data Preprocessing - Data Summarization

There are 1460 records in the training dataset. Measures of central tendency (mean, median, and mode), and measures of dispersion (variance, standard deviation, range, and interquartile range) were calculated and examined for the features. As stated earlier, histograms were used to analyze the distributions of the features. Pearson correlation coefficients and covariance were calculated between all numerical variables. Mutual information was calculated between all variables and the SalePrice. Entropy was calculated for all categorical features.

Figure 3 shows the features with the highest correlation with SalePrice and fig 4 shows the highest correlated feature pairs. Throughout the various iterations of preprocessing and analysis, correlation and mutual information was calculated several times to compare newly engineered features to the original features.

		Coefficient
SalePrice	SalePrice	1.000000
OverallQual	SalePrice	0.790982
GrLivArea	SalePrice	0.708624
GarageCars	SalePrice	0.640409
GarageArea	SalePrice	0.623431
TotalBsmtSF	SalePrice	0.613581
1stFlrSF	SalePrice	0.605852
FullBath	SalePrice	0.560664
TotRmsAbvGrd	SalePrice	0.533723
YearBuilt	SalePrice	0.522897

Fig. 3. Features highly correlated with SalePrice

E. Data Preprocessing - Data Cleaning

I inspected the features to see which ones had a lot of missing data. The results of that inspection can be seen in fig 5. I used a few different strategies to handle missing data. For the LotFrontage feature I used a KNNImputer(n_neighbors=5).

		Coefficient
GarageCars	GarageArea	0.882475
GarageArea	GarageCars	0.882475
YearBuilt	GarageYrBlt	0.825667
GarageYrBlt	YearBuilt	0.825667
TotRmsAbvGrd	GrLivArea	0.825489
GrLivArea	TotRmsAbvGrd	0.825489
TotalBsmtSF	1stFlrSF	0.819530
1stFlrSF	TotalBsmtSF	0.819530

Fig. 4. Highly Correlated Feature Pairs

For some features, where a missing value implied that feature was not present, I filled in missing values with a constant. I did this with a SimpleImputer for GarageYrBlt, MasVnrArea, FireplaceQu, GarageQual, GarageCond, BsmtCond, BsmtQual, and MasVnrType. The other numerical features I imputed with the median value and the other categorical features I imputed with the mode. I initially attempted to reduce any noise using PCA, but discovered that it was not necessary in this analysis. After a few iterations of preprocessing and training, PCA no longer yielded an improvement in the model as the preprocessing improved.

F. Data Preprocessing - Data Transformation

Ordinal encoding was used for these features: "KitchenQual", "Functional", "FireplaceQu", "GarageQual", "GarageCond", "PoolQC", "HeatingQC", "BsmtCond", "BsmtQual", "ExterCond", "ExterQual". After the ordinal encoding was completed, a StandardScaler was applied to the features. A StandardScaler was also applied to the numerical features. The remaining categorical variables were encoded using one hot encoding.

G. Data Preprocessing - Feature Selection

There were several features that had limited examples on which to train a model so I decided to remove them ('Alley', 'PoolQC', 'Fence', 'MiscFeature'). In fig 5 it is shown that most of these features were missing in over 90% of the records.

H. Data Preprocessing - Feature Engineering

I spent a large portion of my time attempting to engineer new features to improve my model performance. Many engineered features were useful, but others were not. I initially engineered four new features: total_sf, finished_sf, quality_sf, and total_baths. total_sf was the

	count	percentage
PoolQC	1453	99.520548
MiscFeature	1406	96.301370
Alley	1369	93.767123
Fence	1179	80.753425
FireplaceQu	690	47.260274
LotFrontage	259	17.739726
GarageYrBlt	81	5.547945
GarageCond	81	5.547945
GarageType	81	5.547945
GarageFinish	81	5.547945
GarageQual	81	5.547945
BsmtExposure	38	2.602740
BsmtFinType2	38	2.602740
BsmtCond	37	2.534247
BsmtQual	37	2.534247
BsmtFinType1	37	2.534247

Fig. 5. Features missing data

summation of 1stFlrSF, 2ndFlrSF, and TotalBsmtSF. finished_sf was total_sf minus BsmtUnfSF. quality_sf was finished_sf minus LowQualFinSF. total_baths was $FullBath + BsmtFullBath + 0.5 * (BsmtHalfBath + HalfBath)$. I then dropped the features: 'FullBath', 'BsmtFullBath', 'BsmtHalfBath', 'HalfBath', '1stFlrSF', '2ndFlrSF', 'TotalBsmtSF', 'BsmtUnfSF', 'LowQualFinSF' from the data. As seen in fig 6, all of these new features were now in the top ten highest correlated with SalePrice. total_sf even had a higher correlation than GrLivArea and almost as high as OverallQual.

		Coefficient
SalePrice	SalePrice	1.000000
OverallQual	SalePrice	0.790982
total_sf	SalePrice	0.782260
GrLivArea	SalePrice	0.708624
finished_sf	SalePrice	0.708047
quality_sf	SalePrice	0.707980
ExterQual	SalePrice	0.682639
KitchenQual	SalePrice	0.659600
BsmtQual	SalePrice	0.650138
GarageCars	SalePrice	0.640409
total_baths	SalePrice	0.631731

Fig. 6. Engineered Feature Correlation with SalePrice 1

As I continued on with my analysis I sought to further improve the model through feature engineering. I looked at the features 1stFlrSF, 2ndFlrSF, LowQualFinSF, and GrLivArea. Previously, I was under the impression that 1stFlrSF + 2ndFlrSF was all the square footage above ground. However, I discovered that GrLivArea (all the SF above ground) is actually the sum of 1stFlrSF, 2ndFlrSF, and LowQualFinSF. This would imply that some of my engineered features were not actually what I wanted them to be. I inspected the below grade square footage features as well. I found that for all records, BsmtFinSF1, BsmtFinSF2, and BsmtUnfSF sum to TotalBsmtSF. I also verified that BsmtFinSF2 is always 0 if BsmtFinType2 is Unf. I re-engineered three out of my four original engineered features. total_sf became the sum of GrLivArea and TotBsmtSF, finished_sf became $GrLivArea + TotalBsmtSF - BsmtUnfSF$, and quality_sf was calculated as follows:

```
def calculate_bsmt_quality_sf(row):
   low_quality_bsmt_grades = ["LwQ", "Unf",
       "NA"]
   quality_sf = 0
   if row['BsmtFinType1'] not in
       low_quality_bsmt_grades:
      quality_sf += row['BsmtFinSF1']
   if row['BsmtFinType2'] not in
       low_quality_bsmt_grades:
      quality_sf += row['BsmtFinSF2']
   return quality_sf

def add_quality_sf(df):
   quality_sf = df['GrLivArea'] -
       df['LowQualFinSF'] +
       df.apply(calculate_bsmt_quality_sf,
       axis=1)
   df['quality_sf'] = quality_sf
```

Now a few more features were dropped as they made up these newly engineered features ('FullBath', 'BsmtFullBath', 'BsmtHalfBath', 'HalfBath', '1stFlrSF', '2ndFlrSF', 'TotalBsmtSF', 'BsmtUnfSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFinSF2', 'BsmtFinSF1').

I also spent a lot of time in an attempt to engineer a neighborhood score feature. I plotted histograms of SalePrice by neighborhood and inspected the data. I also examined the central tendency and dispersion statistics of SalePrice by neighborhood (see fig 7). There is a definite difference in prices by neighborhood. I took the ordinal encoded quality features ('OverallQual', 'OverallCond', 'ExterQual','BsmtQual','ExterCond', 'HeatingQC','KitchenQual','FireplaceQu','GarageQual', 'GarageCond', 'Functional', 'BsmtCond'), grouped the data by neighborhood and took the mean. I multiplied each quality feature by its correlation coefficient with SalePrice and took their summation. I then averaged the median and mean house prices of the neighborhoods, transformed them with StandardScaler,and added this to the new qualityscore I had created (see neighborhood scores in fig 8). While the neighborhood score feature had a pearson correlation coefficient of > 0.7 with SalePrice, in practice it actually made my models perform slightly worse.

I. *Training*

I trained two different classifiers on the data: Ridge and GradientBoostingRegressor. I used GridSearches to tune the hyperparameters of the models. The best performing Ridge Classifier used an alpha of 50. The best performing GradientBoostingRegressor used a learning_rate of 0.03, max_depth of 3, min_samples_split of 20, and n_estimators of 550.

III. Experimental Results

The best performing Ridge regressor achieved a root mean squared error of 25241.84, r^2 of 0.8883 and mean absolute error of 17693.2026. The best performing GradientBoostingRegressor achieved a root mean squared error of 21648.35, r^2 of 0.9179, and mean absolute error of 13996.1387. For fun, I submitted model predictions to Kaggle and the best score achieved was 0.13309. I believe that these results reflect a well performing model for this application. I used the GradientBoostingRegressor to analyze the most important features. The ten most important features according to the model are total_sf, OverallQual, quality_sf, YearBuilt, KitchenQual, GarageCars, BsmtQual, finished_sf, LotArea, and total_baths (see fig 9).

Neighborhood	sum	min	max	mean	median	quantile1	quantile2	std
MeadowV	1675800	75000	151400	98576.470588	88000.0	78200.0	131540.0	23491.049610
IDOTRR	3704580	34900	169500	100123.783784	103000.0	55000.0	140040.0	33376.710117
BrDale	1671900	83000	125000	104493.750000	106000.0	86700.0	121000.0	14330.176493
OldTown	14489459	37900	475000	128225.300885	119000.0	87000.0	161000.0	52650.583185
Edwards	12821970	58500	320000	128219.700000	121750.0	82450.0	177200.0	43208.616459
BrkSide	7240375	39300	223500	124834.051724	124300.0	78600.0	181550.0	40348.689270
Sawyer	10122692	62383	190000	136793.135135	135000.0	110530.0	167100.0	22345.129157
Blueste	275000	124000	151000	137500.000000	137500.0	126700.0	148300.0	19091.883092
SWISU	3564784	60000	200000	142591.360000	139500.0	107200.0	185200.0	32622.917679
NAmes	32815593	87500	345000	145847.080000	140000.0	110000.0	180300.0	33075.345450
NPkVill	1284250	127500	155000	142694.444444	146000.0	127900.0	149800.0	9377.314529
Mitchel	7657236	84500	271000	156270.122449	153500.0	118600.0	202060.0	36486.625334
SawyerW	11006792	76000	320000	186555.796610	179900.0	119712.8	264204.0	55651.997820
Gilbert	15235506	141000	377500	192854.506329	181000.0	167520.0	231600.0	35986.779085
NWAmes	13800655	82500	299800	189050.068493	182900.0	152000.0	241200.0	37172.218106
Blmngtn	3312805	159895	264561	194870.882353	191000.0	164424.0	239031.2	30393.229219
CollgCr	29694866	110000	424870	197965.773333	197200.0	132950.0	260150.0	51403.666438
ClearCr	5951832	130000	328000	212565.428571	200250.0	151400.0	277900.0	50231.538993
Crawfor	10741861	90350	392500	210624.725490	200624.0	139000.0	311500.0	68866.395472
Veenker	2626500	162500	385000	238772.727273	218000.0	165000.0	324000.0	72369.317959
Somerst	19382666	144152	423000	225379.837209	225500.0	162250.0	305238.5	56177.555888
Timber	9205403	137500	378500	242247.447368	228475.0	173500.0	321350.0	64845.651549
StoneBr	7762475	170000	556581	310499.000000	278000.0	188100.0	476614.2	112969.676640
NoRidge	13747108	190000	755000	335295.317073	301500.0	250000.0	430000.0	121412.658640
NridgHt	24352838	154000	611657	316270.623377	315000.0	202500.0	438292.4	96392.544954

Fig. 7. Neighborhood Statistics

IV. Conclusion

In conclusion, I found that the GradientBoostingRegressor performed very well on the data. The biggest contributing factor to the success of the analysis was the engineering of the square footage features. Their addition significantly improved both models. I believe the Ridge regressor could also perform very well on this data, as we did come close to a 0.9 r^2, but I think it would require more time spent processing data and engineering features to get a well performing model. I would like to try to continue preprocessing and feature engineering efforts to bring the root mean square error below 19000 and mean absolute error below 12000.

Neighborhood	mean	median	CTScaled	QualityScore	Score
NridgHt	316270.623377	315000.0	2.172720	17.509386	19.682106
StoneBr	310499.000000	278000.0	1.829536	16.887899	18.717435
NoRidge	335295.317073	301500.0	2.217048	16.396214	18.613262
Somerst	225379.837209	225500.0	0.725329	15.916340	16.641669
Timber	242247.447368	228475.0	0.884539	15.246513	16.131052
Blmngtn	194870.882353	191000.0	0.203721	15.808841	16.012562
Veenker	238772.727273	218000.0	0.772612	14.525010	15.297622
CollgCr	197965.773333	197200.0	0.278300	14.849812	15.128111
Gilbert	192854.506329	181000.0	0.107306	14.455389	14.562695
SawyerW	186555.796610	179900.0	0.047941	14.062479	14.110420
Crawfor	210624.725490	200624.0	0.407343	13.124675	13.532019
ClearCr	212565.428571	200250.0	0.419914	13.032443	13.452357
NWAmes	189050.068493	182900.0	0.092025	13.023392	13.115417
Mitchel	156270.122449	153500.0	-0.406884	12.596290	12.189406
NPkVill	142694.444444	146000.0	-0.575988	12.385003	11.809015
Blueste	137500.000000	137500.0	-0.685867	12.478553	11.792686
SWISU	142591.360000	139500.0	-0.628969	11.955709	11.326741
NAmes	145847.080000	140000.0	-0.598834	11.843832	11.244998
Edwards	128219.700000	121750.0	-0.886701	11.846750	10.960049
Sawyer	136793.135135	135000.0	-0.711598	11.537293	10.825695
OldTown	128225.300885	119000.0	-0.908721	11.727102	10.818381
BrDale	104493.750000	106000.0	-1.203442	11.739930	10.536488
BrkSide	124834.051724	124300.0	-0.893406	11.340799	10.447393
MeadowV	98576.470588	88000.0	-1.395346	11.399043	10.003697
IDOTRR	100123.783784	103000.0	-1.262576	10.916531	9.653955

Fig. 8. Neighborhood Scores

	feature_importance
total_sf	0.357856
OverallQual	0.333668
quality_sf	0.112843
YearBuilt	0.028340
KitchenQual	0.017563
GarageCars	0.016284
BsmtQual	0.015550
finished_sf	0.014104
LotArea	0.012121
total_baths	0.010801
FireplaceQu	0.010363
YearRemodAdd	0.008020
GarageArea	0.007872
OverallCond	0.006616
ExterQual	0.003208

Fig. 9. Feature Importances