House Sales Price Prediction Project

Part 1 of a 2 Part Series Kaggle Competition Project

Nathalie Pham

11/2/2017

Executive Summary

In the Kaggle competition titled “House Prices: Advanced Regression Techniques”, contestants are tasked with building a model that accurately predicts the sales price of over 1,400 residential real estate listings in the town of Ames, Iowa that were sold between 2006 and 2010. The train and test dataset contain every descriptor variable imaginable used by the Ames City Assessor’s Office in evaluating the price of a house. Of course, the assessed price of a house by the city is never reflected in the final sales price of the property, so this prediction contest challenges contestants to use data engineering to weigh the effects of the house descriptor variables set by the city against what buyers would prioritize when purchasing a house. The final model produced should take the descriptive variables into consideration to see how they influence final sales price.

Most of the house variables reflect the type of information and questions that a typical home buyer considers when looking at a potential property. For example, “When was the house built?”, “What is the overall quality of the house?”, “How much square footage is the general living space?”, “How many full baths does the house have?”. In general, there are a total of 80 possible predictor variables in the datasets. The continuous variables are attributed to measuring various area dimensions of a house such as square footage of the basement, living area, linear feet of street connected to the property, and garage area. There are also discrete variables in the dataset that aim to quantify the number of certain items found in the house such as the number of bedrooms, full baths, kitchens, garage cars able to fit inside of the garage, and also the number of fireplaces. Lastly, there is a sizeable number of categorical variables (both nominal and ordinal) within the datasets. Each is segmented into its own grouping of classes that depict characteristics such as the neighborhood location of the house, the material of the house, the type of central heating, and the amenities within distance of the house.

The process of exploratory data analysis of the train dataset was an effective method to learn in-depth the characteristics, nuances, and limitations of the House Prices dataset. A considerable amount of missing values was discovered for 19 of the descriptor variables. Missing values were addressed through data imputation by replacing the null values with values that were deemed appropriate on a case by case basis. The dependent variable Sale Price was also found to deviate from normality, and displayed both positive skewness and positive kurtosis. In order to address this issue, a log transformation was applied to Sale Price to make it normally distributed for future processing. Correlation coefficients for all predictor variables were found against SalePrice and the top thirteen were analyzed closely for any glaring trends. OverallQual and GrLivArea were found to be the best two predictors of SalePrice; both showed strong, positive correlations with the dependent variable. The top strong predictors of SalePrice were found to be OverallQual, GrLivArea, GarageCars and GarageArea, TotRmsAbvGrd, TotalBsmtSF, 1stFlrSF, and FullBath. Some categorical features were also examined to search for any trends in predictive ability. It was found that Neighborhood, HeatingQC, KitchenQual, Street, PavedDrive, and SaleType had good positive correlations with SalePrice, and should be considered as predictors for future models.

In total, three Kaggle Solutions were evaluated for the House Prices Competition. The most complex and effective model was the result of using a Stacked Model Regression. Performance of this model boosted the contestant to the top 4% of the Leaderboard, with a score of 0.11542. The basic idea behind this model is to split the training data into several stacks and have different models train on iterations of the dataset. From each iteration, the model will create an outfold prediction. At the end, all of the outfold predictions are compiled and used to create a new feature to which the meta-model will train against to produce the final prediction. For the solution, an Average Base Model stacking approach was used with ENet, KRR, and GBoost regression models and the Lasso model was utilized as the final meta-model to produce the prediction results.

Data Description and Initial Processing:

In the Housing Sales Prediction Competition, two datasets are provided for the contestant to create models and predict sales prices from; they include a train and test dataset. The train dataset was utilized for data description and initial processing since this is the dataset that the eventual model will be created from. All figures and tables that have been referenced can be found in the exploratory data analysis notebook submitted along with the report.

General data analysis was first conducted to gain a basic understanding of the data. The train dataset has 1460 rows and 81 columns to work with which indicates that there are 80 variables for prediction of sales price. The test dataset has 1459 rows and 80 columns to work with; there is one less column in the test dataset because it does not include the SalePrice column, which is what needs to be predicted. From a high level view, there is both a combination of qualitative (object) and quantitative (integer and float) variables in the train dataset (Table 1). Out of the 80 columns, 38 columns contain quantitative data while the other 43 contain qualitative data. There are also missing variables present in the train dataset which affects its quality.

Addressing Missing Variables:

Missing variables were first addressed in the dataset before any data processing was conducted. There are a total of 19 predictor variables that have varying amounts of missing values (Table 3). The top six variables that have the most missing values include PoolQC (pool quality), MiscFeature (miscellaneous features not covered in other categories such as elevator, 2nd garage, tennis court, etc.), Alley (type of alley access to the property), Fence (fence quality), FireplaceQu (fireplace quality), and LotFrontage (linear feet of street connected to the property). Amongst the 19 variables with missing data, all contain qualitative data except for LotFrontage, GarageYrBlt (garage year built), and MasVnrArea (masonry veneer area in square feet) which are quantitative features.

Each feature with missing values was analyzed in order to determine how best to deal with the null fields. Data imputation was carried out where necessary. As seen in Figure 1, Pool Quality and Pool Area were plotted on a barplot. Only seven houses in the dataset have pools and the 1453 houses that do not have pools have a pool area of 0 square feet. For those 1453 houses, a value of “None” was imputed for the PoolQC to reflect the fact that no pools were present. About 1400 houses do not have miscellaneous features included in their design (Figure 3), so those houses were given a MiscFeature value of “None”. Whether a house contains a miscellaneous feature or not does not seem to affect the SalePrice too much since the mean SalePrice for all categories is more or less the same (Figure 4). Houses with the Alley feature have either gravel or paved alley access (Figure 5), the latter being a feature of houses sold at higher prices than those of the former. It can be assumed that missing values correspond to houses that have no alley access, therefore their missing data can be imputed as “None” (Figure 6).

Fireplace quality (specifically in excellent condition) has an overall greater effect on SalePrice than fireplace quantity (Figure 7). In particular, houses with 2 fireplaces in excellent condition sell for very high prices. If there are zero fireplaces in the house, it is assumed that the fireplace quality is “None” so this is what is imputed for the missing values (Figure 8). All of the features describing Garages had the same number of missing values (81), indicating that they most likely referenced the same set of houses. If the feature was qualitative (such as GarageCond, GarageType, GarageFinish, and GarageQual) a value of “None” was inserted while a value of 0.0 replaced null values for the GarageYrBlt feature. All of the null values pertaining to Basement features were given a value of “None” because they were qualitative. For both Garages and Basements, null values indicated that no garage or basement was present in a particular house. Not all houses have fences either, so the missing fields were given a missing value of “None” as well.

For the Electrical feature (Figure 10), and it was observed that most of its values fall under the Standard Circuit Board & Romex category (SBrkr). Due to the majority, the one missing value in the Electrical column was determined to be “SBrkr”. For the two features MasVnrArea (masonry veneer area) and MasVnrType (masonry veneer type), it was assumed that if the indication was 0 square feet or “None”, a particular house did not have any masonry characteristics. Null values for MasVnrArea and MasVnrType were replaced with 0.0 and “None” respectively. Lastly, the LotFrontage feature was assessed. Figure 13 indicates that the majority of houses have less than 150 feet of street connected to the property. The cluster on the scatterplot is dense with not that much distribution along the Y axis; this shows that LotFrontage might not be a good predictor variable for consideration. There are some extreme outlier values for LotFrontage though, so missing values were replaced with the median (69 feet) rather than the mean (70.05 feet). After this last data imputation, there were no longer any missing variables in the train dataset (Table 4).

Descriptive Statistics:

Basic statistics and visualizations were conducted for SalePrice to gain a better understanding of the dependent variable. From a first glance at the descriptive statistics (Table 5), the SalePrice variable is positively skewed. The minimum value ($34,900), 25% quartile ($129,975), median ($163,000), and 75% quartile ($214,000) values are all relatively close to one another while the maximum value ($755,000) is considered an outlier. In order to verify that SalePrice does not follow a normal distribution, a histogram plot, probability plot, and boxplot were created. As seen in Figure 14, SalePrice does not have a symmetric distribution but is rather positively skewed because the majority of values clump around the lower end of the distribution spectrum. The distribution of SalePrice also has positive kurtosis (leptokurtic), also known as very high peakedness. Actual values of skewness and kurtosis for SalePrice are 1.8829 and 6.5363 respectively. The fact that the plotted values do not lie on the best-fit line in the probability plot (Figure 15) also indicates a departure from normality and also that there are some obvious outliers on the upper end of the value spectrum (select houses were sold for very high prices). The boxplot distribution of SalePrice (Figure 16) shows clearly the amount of outliers within the variable. To account for a non-normal distribution, a log transformation was applied to SalePrice. The resulting histogram after log transformation shows a symmetric bell-shaped curve (Figure 17) and the value distribution follows the best-fit line closely on the probability plot (Figure 18).

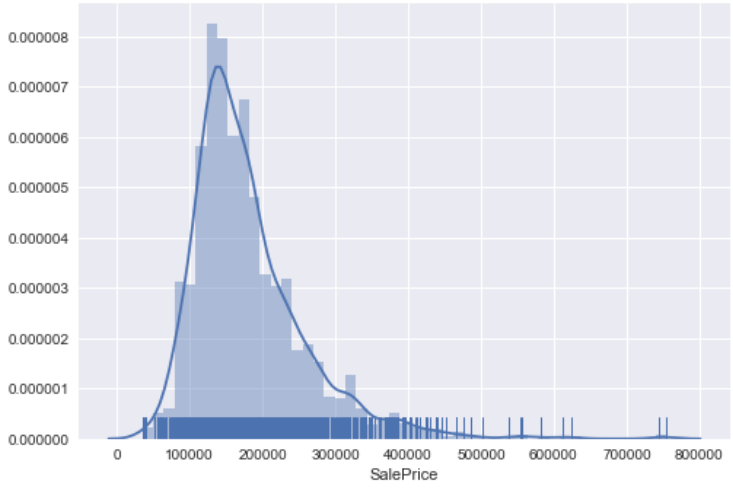

Figure 14: Histogram plot of SalePrice.

Assessing Predictor Variables:

To assess the viability of some quantitative features being predictors of SalePrice, a correlation coefficient chart and heat map were created. From the correlation chart (Table 6), SalePrice has a very strong positive correlation with OverallQual (0.817) and GrLivArea (0.701). Both detail the general perception of the quality and size of the house and are typical considerations of buyers. Overall, 10 features have correlation coefficients with SalePrice above 0.5, indicating that they are potential strong predictor candidates of sales pricing. In continuing with analysis, Fireplaces and MasVnrArea were also included even though they had less than a 0.5 correlation coefficient with SalePrice (0.489 and 0.427 respectively); this was done to broaden the scope of analysis. From the heat map created (Figure 19), it can be seen that along with high correlations with SalePrice, some of the other features correlate strongly with each other. This indicates multicollinearity, which in the future can present obstacles when trying to determine which specific variables affect SalePrice. Some obvious pairs of highly correlated predictors include TotRmsAbvGrd and GrLivArea, GarageArea and GarageCars, and TotalBsmtSf and 1stFlrSF. Some pairs of features can be considered redundant such as GarageArea and GarageCars; the number of cars that can fit in a garage depends on its area, so both features are different ways of representing the same parameter. One consideration then would be to remove one feature from the pair to reduce multicollinearity.

Figures 20-30 indicate visualization analysis of the top 12 features that have the highest correlation coefficients with SalePrice. In general, increasing the value, quantity, or metric of any of the 12 features typically correlates to higher SalePrice. For example, an OverallQual score of 10 is a strong predictor of high SalePrice (Figure 20). More space that constitutes living area (seen with the predictor variables TotalBsmtSF, 1stFlrSF, GrLivArea, and TotRmsAbvGrd) are also indicative of high sales prices since their figures show positive, increasing trends (Figure 21, 22, 25).

Some interesting things to note include that four car garages do not seem to affect SalePrice that strongly; houses with three car garages sell for higher. Buyers might consider three car garages to be a good median, with four car garage houses being too much and 2 being too average (Figure 24). The year the house was built also does not affect SalePrice strongly. More recently built houses generally sell for higher prices, but prices of recently built houses are comparable with prices of those built even 50 years ago (Figure 26). A similar trend is seen in the scatterplot for YearRemodAdd (Figure 30) in which recently remodeled houses generally sell for higher SalePrice, but the effect is not as strong. It seems that the buyers of today do not generally weigh remodel year or the year the house was built that heavily into their decision. They may buy the house to remodel themselves anyways so the age of the house would not be a factor they would consider, or maybe older houses are considered vintage with a lot of character and charm so buyers are willing to spend money on them. In all, GrLivArea, GarageCars, TotRmsAbvGrd, TotalBsmtSF, 1stFlrSF, and FullBath are strong potential predictors to be considered for SalePrice.

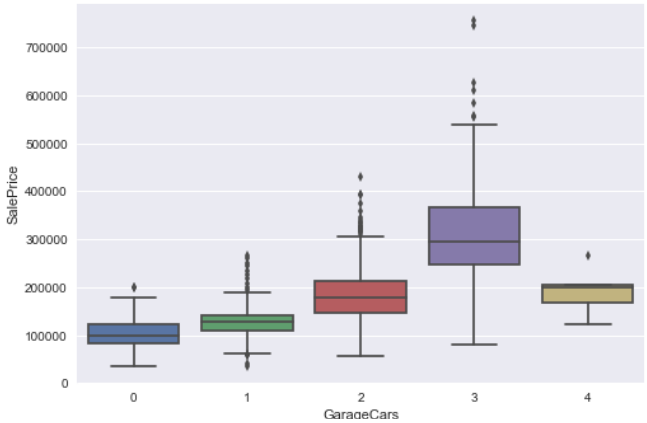

Figure 24: Boxplot of GarageCars vs. SalePrice.

Lastly, Figures 31-41 detail the visualization analysis of some of the categorical features for prediction of SalePrice. Since neighborhood is traditionally the most notable predictor of SalePrice, it is interesting to note that Northridge Heights, Stone Brook, and Northridge are three of the top neighborhoods for high price residential houses (Figure 31). MSZoning also seems to have an effect on SalePrice (Figure 32); houses that are built in residential low density or floating village residential areas typically sell for very high prices in comparison to the other zoning areas. Excellent kitchen quality is also a good predictor of sales price regardless of the quantity in the house (Figure 36). Similar to previous conclusions made about the YearRemodAdd and YearBuilt variable, newer houses generally sell for higher prices (Figure 40).

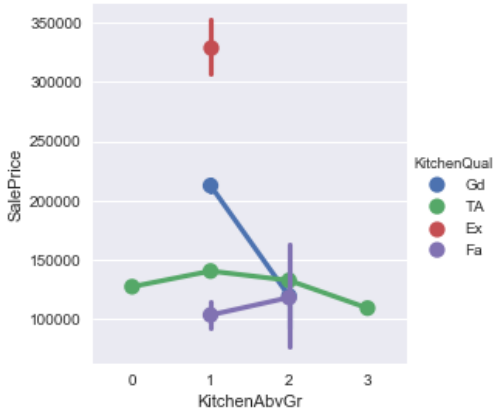

Figure 36: Factorplot of KitchenAbvGrd vs. SalePrice with KitchenQual as a color filter.

Modeling and Evaluation of Kaggle Solutions:

Kaggle Solutions Summary			
Solution	**Features**	**Modeling Approach**	**Performance**
1	- The train and test datasets were first concatenated in order to check for missing data. Data imputation was conducted on all of the variables with missing values; either “None”, a value of 0, or the most common value in the feature was imputed. - New categorical features were created by transforming some numerical variables such as MSSubClass, OverallCond, YrSold, and MoSold. - Label encoding was done for some categorical variables that contained subclasses of information in their ordering set.	- Overall uses a Stacked Model Regression Approach. Utilizes the cross_val_score function from Sklearn to judge effectiveness of each model. - First tested out 6 regression models to be used in the Base model. Tested LASSO, Elastic net, Kernel Ridge, Gradient Boosting, XGBoost, and LightGBM. Based on the cross-validation rmsle error scores, chose ENet, GBoost, KRR, and LASSO to continue with. - Started with the Average Base Models stacking approach. This involves one score computed by	By using the Stacked Model Regression approach the contestant was able to get to the top 4% of the Leader Board. On the Leader Board, the model received a score of 0.11542. This model approach was the most effective and complex out of the three solutions evaluated. It is quite robust due to the

	- A new feature (TotalSF) was calculated and this reflected the total area of the house. This was computed by adding the area of the basement, first, and second floors. - Box Cox Transformation was done for all of the highly skewed features (such as PoolArea, LotArea, etc.). - Created dummy categorical features on the entire concatenated dataset. - Created a new feature from all of the hold out predictions which was used to train the metal model.	averaging the individual scores of each model run against the dataset. - Added a Meta-model for complexity. In this approach, the total training set is split into several-fold parts. Each of the chosen base models is used to train the data set and then predict on the holdout fold for a particular iteration. Once all of the holdout fold predictions are assembled, these meta-features are then used to create a new feature that trains the meta-model, which conducts the final prediction. - Changed approach to using Stacked Average Model with ENet, KRR, and GBoost and utilized LASSO for the meta-model.	fact that the model allows for the combination of many predictive models to be trained against a dataset to improve its accuracy. The best performance is observed when the base models used are different so that when the stacked model is used to predict it will be able to highlight each individual model's strengths and subdue its weaknesses. The best of all models is incorporated into prediction.
2	- For data imputation, replaced all of the numerical missing variables with medians (less sensitive to extreme values) and all of the categorical missing values with "None". - Changed MSSubClass feature from numeric to categorical. - Changed categorical features into numerical features (ex. BsmtCond). - Dropped entire columns for the Street, Utilities, Condition2, RoofMat1, and Heating variables. - Unskewed all of the data in the train dataset. - Bonferroni test was used to statistically identify outliers and remove them. Also manually detected outliers by visualization. Most obvious outlier removal was within the GrLivArea predictor variable. - Combined both the test and train dataset to get median values in order to fill in the missing values for numerical data. - Removed over-fit columns (MSSubClass and MSZoning).	- Started analysis by building a base model with Random Forest Regressor. Came across issues with using RF though. There was a lot of noise in the cross validation score and the model was very slow. Switched to the XGBoost model for testing train data throughout EDA. - Next, chose to use LASSO model since it was the most performant with the dataset after some feature engineering; it outperformed the XGBoost score. Utilized the LassoCV algorithm instead of the standard LASSO algorithm. Interesting thing about the Lasso model is it will drop features that are not useful due to the use of the L1-norm. It will also indicate features that have high weights in the dataset (ones that are good predictors of SalePrice). - On the final feature dataset, conducted both a Lasso model and XGBoost model and averaged the two scores.	R^2 score for the Lasso model was 0.11912 (top 17%) and for the XGBoost model the R^2 score was 0.11974. The average of both of the scores put the contestant in the top 10% on the Leader Board. The contestant averaged the models to cancel out the errors (under/over estimates) in each model with the predictions of the other model. Both are built in different ways so they are able to account for the other's mistakes. Final score on the Leader Board was 0.11549. Averaging model prediction scores seems to be a common procedure with many Kagglers; this is under the assumption that no one regression

			method is robust enough to produce a high prediction score alone.
3	- Concatenated the train and test datasets for one time manipulation of data. - Removed the ID variable as well as any columns with more than 1000 missing values. - Applied one hot encoding to convert categorical variables into dummy/indicator variables. - Filled in null values of quantitative variables with median. - Performed one hot encoding on categorical variables. - Log transformed variables to correct for skew.	- Started out by using PCA (Principal Component Analysis) to reduce the number of features in the dataset and select the most important variables. This reduced the dataset down to 36 components to predict sales price. Due to this method of feature reduction, contestant was able to limit the amount of feature engineering in the exploratory data analysis. - Next, concatenated dataset was split and multiple models were trained against the train data. First observed the R^2 values using the models with the raw features. Compared those results with the results obtained when the models were run against the PCA selected data. Ran the following models: Lasso, Linear, Bayesian Ridge, SVM Linear, RandomForest, Bagging, Ridge, Huber, SVM RBF, and AdaBoost. - When the models were run against the PCA data, the Huber model was found to be the best performant and had a R^2 value of 0.859653. Huber is a robust model for data containing outliers. Chose to use the Huber Regression Model. - Applied a Simple Neural Network to improve the result.	This Kaggle solution was very simple and provided a different approach than the previous solutions. By using a combination of PCA for feature reduction and the Huber Regression Model in order to predict sales prices, this contestant received a public score of 0.14655 on the Leader Board. I found PCA to be a simplistic yet interesting approach to reducing the number of dimensions in the dataset down to its basic components. The only caveat with this is that while feature reduction is time and effort efficient, reducing predictor variables down to the bare bones strips outside predictive ability of other information from the other variables that were removed, which are now no longer included in the model.

Appendix:

The exploratory data analysis code has been submitted into the Github repository set up for the project. Links to the Kaggle solutions studied can be found below:

1. https://www.kaggle.com/serigne/stacked-regressions-top-4-on-leaderboard/notebook
2. https://www.kaggle.com/fiorenza2/journey-to-the-top-10/notebook
3. https://www.kaggle.com/miguelangelnieto/pca-and-regression/notebook

Outside links that were referenced for further research and understanding:

1. http://blog.kaggle.com/2016/12/27/a-kagglers-guide-to-model-stacking-in-practice/

2. https://www.kaggle.com/jimthompson/ensemble-model-stacked-model-example
3. https://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/