House Sales Price Prediction Project

Part 2 of a 2 Part Series Kaggle Competition Project

Nathalie Pham

12/14/2017

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. 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 models produced in this report take the descriptive variables into consideration to see how they influence final sales price.

During the second phase of this House Prices Prediction Project, emphasis was placed on learning how to do feature ranking, feature selection, and modelling. Even though exploratory data analysis was the focal point of Project 1, its importance was appreciated even more so during Project 2 as being a crucial step for understanding the data and helping to frame decision making in the modelling stage. All of the data analysis in Project 1 was run again on the train dataset in addition to feature engineering and other pre-processing steps to better clean the data for modelling.

Feature ranking and feature selection were the first learning curves. Both steps are necessary in modelling since it is not always advised to input all of the available predictor variables into algorithms; this becomes an important consideration especially if the dataset being used is large. Using all of the predictor variables may or may not reduce predictive power of the models, training time, and can create noise in the data. In order to downsize the large train dataset that was created after exploratory data analysis, three different techniques for feature ranking were explored in this project: Pearson’s Correlation, Univariate Analysis, and Random Forest. The first two are examples of statistical tests that rank features using metric scores (Pearson’s correlation coefficient and univariate scores respectively). Random Forest is in of itself a modelling algorithm that naturally feature selects during training and prediction. As a result, it can be trained against a dataset and made to rank the gini importance of independent variables in their predictive capability of the dependent variable. When the ranking results of all three methods were compared, features were chosen for inclusion in the model if they were commonly ranked amongst at least two feature ranking techniques. The final predictor dataset included the features TotalSF, GrLivArea, GarageCars, 1stFlrSF, FullBath, GarageArea, TotalBsmtSF, ExterQual, KitchenQual, TotRmsAbvGrd, Fireplaces, MasVnrArea, OverallQual, Heating, and HeatingQC.

In total, three modelling algorithms were used to predict SalePrice based off of the condensed predictor variable dataset created in the feature selection stage. Since SalePrice consisted of real numbers, regression models were chosen. The models included Random Forest Regressor, Linear Regression, and Ridge Regression. Before any of the selected models were run, a Base Mean Model where all of the SalePrice predictions were the average SalePrice was executed to obtain a baseline. For all models, an 80/20 train test split was done on the curated train dataset of chosen predictor variables and the SalePrice column. The Base Mean Model performed the worst in predicting SalePrice for both the 80% train and 20% test sets. The best performing model in the 80% train set was the Random Forest Regressor model which had a RSME score of 0.06494. The best performing model in the 20% test set was the Ridge Regression model which had a RSME score of 0.15371.

Analysis of Independent Variables:

Before delving into the fun parts of the project (i.e. feature selection and modelling), a revised iteration of exploratory data analysis was re-conducted on the train dataset. All of the

analysis seen in Project 1 was repeated and after further consideration more feature engineering was implemented. As a recap of Project 1, features with missing values were isolated and value imputation was dealt with on a feature by feature basis (Table 1). The dependent variable SalePrice was also found to not follow a normal distribution so the variable was log transformed to normalize the data (Figure 1).

Revised Exploratory Data Analysis:

The new data analysis seen in Project 2 takes a closer look at the predictor variables and were modes of additional "data cleaning" to better prepare the train dataset for feature selection and modelling. First, all of the predictor variables were reviewed with the complimentary Kaggle data descriptions to see if they were categorized correctly. For example, the predictor variables MSSubClass, OverallCond, OverallQual, YrSold, MoSold, YearBuilt, YearRemodAdd, and GarageYrBlt were incorrectly classified as numerical variables; they were all converted to the proper data type (categorical). The categorical variables were reviewed but none were converted to numerical data types. Label encoding was considered for some categorical variables that had an order to their labelling such as OverallQual and OverallCond, but ultimately this was decided against and instead one hot encoding was opted for as the safer option to convert all categorical variables into dummy variables. This step was essential for the modelling portion since Sckikit learn only utilizes real numbers. The numerical variables were also found to have a high degree of positive skewness (for example MiscVal with a skew of 24.45, PoolArea with 14.81, and LotArea with 12.20) (Table 2). Similar to the dependent variable, all numeric variables with skewness greater than 0.0 were log transformed to normalize the data. Finally, a new feature called Total Square Footage (TotalSF) was created to represent the entire square footage of each house. The variable was computed by adding the total basement square footage, the square footage of the first floor, and the square footage of the second floor. Graphing TotalSF against SalePrice on a scatterplot with the accompanying regression line (Figure 2) indicated that the new predictor variable had a very strong positive correlation with the dependent variable.

Exploration of Feature Ranking Methods for Feature Selection:

A large learning curve in this project involved understanding the criticality of feature ranking and feature selection as a pre-step to modelling. Even with the best data analysis and feature engineering, if irrelevant variables are inputted into an algorithm this can lead to noise in the data and a decrease in accuracy for many models. Feature selection becomes even more important when the number of predictor variables is large (as seen with this Ames Iowa dataset after the addition of dummy variables and feature engineering). Feature selection therefore helps to reduce algorithm training time, reduce overfit, and also helps to improve the accuracy of the models. Three methodologies were utilized for feature ranking to justify feature selection: Pearson's Correlation, Univariate Analysis, and Random Forest Regressor.

Pearson's Correlation

Pearson's Correlation is a type of statistical filter method that can be used to show which numerical features are highly correlated with a numerical dependent variable. The metric value is between -1 and +1 where scores closer to the extreme values indicate a negative linear dependence or a positive linear dependence, respectively, between two continuous variables. After conducting the test and mapping the results on a heatmap (Figure 3), it was found that SalePrice was highly correlated with TotalSF (0.81), followed by GrLivArea (0.73), GarageCars (0.68), 1stFlrSF (0.61), FullBath (0.58), TotRmsAbvGrd (0.54), and Fireplaces (0.51). The heatmap also highlighted predictor variables that were correlated with each other (incidence of multicollinearity) such as TotalSF with GrLivArea (0.86), TotalSF with 1stFlr (0.75), TotRmsAbvGrd with GrLivArea (0.83), and GarageArea with GarageCars (0.73). Instances of

multicollinearity make sense since the area available in the garage relates to the number of cars one can fit in it, and variables that depict square footage in the house (such as TotalSF) naturally are correlated with other similar variables of square footage and space (such as GrLiveArea and TotalBsmtSF).

Univariate Analysis:

Univariate analysis was the second feature ranking method used. Since Pearson's Correlation only took into consideration continuous variables that correlated with SalePrice, Univariate Analysis was conducted to also determine the importance of categorical variables. Univariate feature selection is another mode of statistical analysis that can select the best features from a dataset by returning a univariate score; high scores indicate the importance of that predictor variable in determining the dependent variable. In the project, an f regression was conducted since SalePrice is a continuous variable and not a binary variable. The resulting univariate scores (Figure 4) were compared with the feature names (Figure 5) and it was found that the top predictors were TotalSF, GrLivArea, GarageCars, GarageArea, and TotalBsmtSF.

Random Forest:

The final method of feature ranking involved using a Random Forest algorithm to rank the predictors based on how well each improved the purity of the nodes within the algorithm. A Random Forest Regressor model was trained with the entire independent variable dataset and the resulting metric was a gini importance value assigned to each independent variable (Figure 6). The sum of all of the gini values of the predictor variables equals one, so features with a gini value close to one are considered important. SelectFromModel was used to distinguish independent variables that had a gini importance over a threshold of 0.004. Table 4 ranks these top 21 features and it can be noted that TotalSF was the top predictor and had a gini importance of 0.64952.

Feature Selection:

The accompanying Excel spreadsheet titled Feature Selection attached to this report highlights the thought process behind feature selection based off of feature rankings. Sheet 3 (Univariate Results) shows how the top 30 features were pulled from the Univariate Analysis based on their univariate scores. Conditional formatting was conducted on the Univariate Scores in order to return the top 30 features (green cells). These were then matched with the Univariate Score Order (red cells) and the complimentary Feature Names (yellow cells). Sheet 2 (Univariate Orders) is a listing of the top 30 features from Sheet 3; a sorting function was applied to the Score column in order to get the ranking in descending order.

The most important sheet is Sheet 1 (Feature Selection). The top features ranked amongst each method were listed in each column. The top 10 features across all three methods (highlighted within the dark black box) were considered for inclusion. Features highlighted in yellow were commonly ranked across all three methods and were definitively chosen as input model variables. Rationale behind this was if all three methods deemed the same variables as important, then the variables must have some degree of importance in predicting SalePrice. Features highlighted with any other color (orange, purple, and green) that fell within the black box were also included for modelling. Even though these colors represented features found commonly between only two selection methods, this was considered enough support to include in the model. Features within the black box that were not highlighted were not included (as this indicated no commonality amongst the feature ranking methods).

Overall, based off of the results of Pearson's Correlation, Univariate Analysis, and Random Forest Regressor, TotalSF, GrLivArea, GarageCars, 1stFlrSF, FullBath, GarageArea, TotalBsmtSF, ExterQual, KitchenQual, TotRmsAbvGrd, Fireplaces, MasVnrArea, OverallQual, Heating, and HeatingQC were considered relevant for modelling (Table 10).

Analysis of Modelling Performance:

A baseline model was first created for comparison of subsequent models. Since SalePrice was previously log transformed to account for skewness, the average of the SalePrice column from the train dataset was computed rather than the median (hence the baseline model was deemed the Mean Model). The average was 12.024 and a new column titled PredMeanSP was imputed into the train dataset with this value (Table 11). An 80/20 train test split was done on the train SalePrice and PredMeanSP columns to make the results comparable to the actual models. The Root Squared Mean Error (RSME) was then calculated for predictions in both the 80% train (0.39531) and 20% test (0.16115) sets. The actual House Prices Kaggle competition evaluates submissions based on the RSME value so it was utilized for this project; the metric is used to measure how close observed values are to the values estimated by a predictive model. It is a good indicator of absolute fit since RSME can also be interpreted as the standard deviation. A low RSME near 0 indicates a better fitting regression model. Thus, as a standalone, the baseline Mean Model had very low predictive power on the SalePrice of both the train and test sets after the train test split.

For the actual models, another 80/20 train test split was conducted on the curated train dataset of selected predictor features and the SalePrice column. This was important for cross validating the chosen models to prevent overfit. Since SalePrice is a variable with real numbers, my model choices strayed towards being regression type models since I considered them simple yet powerful algorithms that would be able to estimate the relationship between my chosen predictors and SalePrice.

Random Forest Regressor:

The first model chosen was Random Forest Regressor since it was known to be a very robust, accurate, and stable algorithm. The premise of regression trees involves the algorithm fitting a regression model to the target variable and splitting the data in a top down fashion from root to node. Features are successively grouped into subsets (nodes) based off of calculations of the Sum of Square Error between actual values and predicted values. A Random Forest Regressor model was first initialized with selected parameters and trained with the train sets. The model was then used to predict SalePrice for both the 80% train and 20% test datasets. The resulting RSME scores were 0.06494 and 0.16115 respectively. It was interesting to note that while Random Forest performed better than the Base model at predicting SalePrice for the 80% train dataset, its performance was the same with the Base model on predicting SalePrice for the 20% test set (both had the same RSME scores).

Linear Regression:

The second model explored was the Linear Regression model. As one of the simplest and most widely known modelling techniques, not a lot of stock was placed into its performance. Linear Regressions suffer from the presence of multicollinearity and outliers and are conducted on the assumption that a linear relationship exists between the independent and dependent variables. As one can recall, the choice was made to keep features that were highly correlated (evidence of multicollinearity) as predictor variables for training, so this caveat was kept in mind. The model had a RSME score of 0.15947 for predicting the 80% train set and a RSME score of 0.15528 for the 20% test set. Linear Regression performed worse than Random Forest and better than the Mean Model on the train set but surprisingly performed better than the previous two models on predicting SalePrice in the 20% test set.

Ride Regression:

Finally, Ridge Regression was chosen as the last model. The algorithm is suited for predicting on datasets where the data suffers from multicollinearity (ideal for the one compiled in this project) and is a classic example of an embedded method with its own built in feature selection. Upon assessing

its performance, the Ridge Regression Model had a RSME score of 0.15991 for the 80% train set and a RSME score of 0.15371 for the 20% test set.

Modelling Results:

The table outlined bellows lists all of the models created in this project with their respective RSME scores for predicting SalePrice on the 80% train and 20% test sets. The models ranked in order from best to worst at predicting the 80% train set were as follows: Random Forest Regressor, Linear Regression, Ridge Regression, and the Base Model. The models ranked in order from best to worst at predicting SalePrice of the 20% test set were as follows: Ridge Regression, Linear Regression, Random Forest Regressor, and the Base Model. In all cases, the Base Model performed the worst (to be expected). More stock is placed in the predictive power of the models for the 20% test set though since the test set constituted unseen data that the model had to predict SalePrice from. All four models were trained using the train datasets and were subsequently used to predict again on the train and then test sets. Within the scope of this project, Ridge Regression and Random Forest Regressor can be considered the best predictive models of SalePrice.

Model	Train 80% RSME	Test 20% RSME
Base Mean Model	0.39531	0.16115
Random Forest Regressor	0.06494	0.16115
Linear Regression	0.15947	0.15528
Ridge Regression	0.15991	0.15371

Appendix:

The exploratory data analysis code has been submitted into the Github repository set up for the project.

Outside links that were referenced for further research and understanding:

1. https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/
2. https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/
3. https://www.analyticsvidhya.com/blog/2015/06/tuning-random-forest-model/
4. http://scikit-learn.org/stable/supervised_learning.html#supervised-learning