Ngoc Nguyen (Nora)
Professor Suning Zhu
BAT-3305
22 March 2023

Kaggle Housing Price Project Report

1) Introduction and description of the competition:

Kaggle Housing Price Project is an online data science project that challenges participants to develop the most accurate prediction models for house price. In particular, the competition provides a train and a test data set. Participants can submit their predictions for the test dataset online to see the accuracy of their models.

2) Data description:

Kaggle provides participants with both train and test datasets. Both have 79 explanatory variables describing different aspects of residential homes in Ames, Iowa. The train dataset has an extra column of house sale price for the model training process. All data in both datasets were collected in 2012.

3) Initial data cleaning:

For initial data cleaning, I inspected potential problems and solved them one by one for every observation. I believe this method is faster and more holistic than considering each variable at a time. The potential problems of the provided data are assessed as follows:

- Checking if there are any duplicate observations: Duplications should be removed for better prediction. From the result of this step, there is no duplicated row in this data set.
- Checking if any observation has NAs on more than 50% of the columns: Columns with overwhelmingly missing values do not provide useful insights for the models (except when the NA values implying something else other than insufficiency in data collection). From this step, we know there is no column with more than 50% of the values being missing.
- Identifying missing values and replacing them with:
  - most common value in the same category for categorical variables
    - For example, missing MSZoning values for houses in the "Mitchel" Neighborhood are replaced with the most common MSZoning for other houses in Mitchel Neighborhood – “RL”.
  - mean of the observations in the same category
    - For example, missing LotFrontage for observations with FV MSZoning are replaced with the mean for other houses with MSZoning being FV – which is 59.5.
  - a different level if the NA is representing some other meanings
- Applying changes to some numeric variables: I converted some numeric into categorical to get better details from each level (e.g., OverallQual, MoSold, etc.). Additionally, YearBuilt was used to calculate HouseAge, which represents the newness of houses. Finally, with many variables representing area of different types of porches, I combined them into one variable to represent the overall porch area.

- Identifying typos for categorical variables: For this step, I wrote a function to print out different levels of each categorical var to see if it is consistent with the dataset description. If there is some inconsistency, I will change the level to what is listed in the data description.
- Checking variable distribution: I wrote a function to filter out columns with imbalance distribution (those having 95% or more observations belonging to just one level). I then excluded those variables from model building because they do not give much useful information.
- Checking skewness for numeric variables: Variables with skewness larger than 1 will be filtered out to apply suitable transformation (13 numeric variables are highly skewed).
- Converting categorical variables into factor type.
- Handling outliers: I decided to work on outliers after running the plain vanilla model, particularly based on the Cook's Distance plot. Because the Cook's Distance a great way to identify overinfluential points. Another way is plotting the variable and identifying the points falling out of the general trend. However, an extreme value in the plot may not be an outlier with regard to SalePrice and may have important insight. Thus, dealing with overinfluential points after the plain vanilla model is more reasonable.

4) Diagnostic tests and the prescriptions:

The model diagnostic tests were conducted as follows:

- Checking multicollinearity: Before applying the vif() function to check for multicollinearity, I implemented alias() function to identify aliased coefficients in the model and drop the variables that are aliased to the others. Then, from the vif value, I assessed variables with a value larger than 5 and decided which variables to keep and to drop.
- Checking nonlinearity & heteroscedasticity with Residuals vs Fitted plot: In the Residuals vs Fitted plot, since the red curve in the plot is relatively horizontal, the residuals values have a mean of close to 0, implying there is no non-linearity issue. On the other hand, the residuals value are distributed in a wider range below 0 than above and vary more in the middle than the begin and end. Accompanied with the curve pattern in the Scale-Location plot, we can conclude that heteroscedasticity issue exits in the model. The prescription is to apply concave transformation on the response variable – SalePrice.
- Identify influential points with Cook's Distance plot: Two points with Cook's Distance way exceeding 1 were removed from the model.

5) Comparing models by various analytical:

Besides the plain vanilla model, I applied 9 methods in total: Forward AIC, Backward AIC, Hybrid AIC, Lasso, Ridge, Simple Regression Tree, Bagging, Radom Forests, and Boosting.

Plain Vanilla	Forward AIC	Backward AIC	Hybrid AIC	Lasso	Ridge	Simple Regression Tree	Bagging	Radom Forests	Boosting
0.13528	0.28571	0.13418	0.28571	**0.12553**	0.13138	0.24225	0.14825	0.14877	**0.12511**

Table 1: Public scores of different models generated from different machine learning methods

Note: Tree Pruning was applied. It is indicated that the generated tree does not need pruning because the best tree is similar to the tree from the Simple Decision Tree method.

Based on the public score table, the Boosting method generated the most accurate model, followed by Lasso. However, since the Boosting model has low interpretability, the Lasso model is superior due to its interpretability and relatively great accuracy. Therefore, in my opinion, the Lasso model would provide better insights on the predictors regarding the response variable.

6) Insights from the best model:

With a public score of 0.126, the Lasso model is relatively accurate. From this model, the most important coefficients include: GrLivArea, OverallQual9, FunctionalSev, OverallQual10, OverallQual2, LotArea, OverallCond3, FunctionalMaj2. These are the key aspects to determine the house price. The coefficients of these variables can be interpreted as follows:

- GrLivArea: For every 1% increase in above-ground living area in square feet, the median house price will, on average, increase by around 0.414%, holding everything else constant.
- OverallQual most important levels:
    - OverallQual9: Holding everything else constant, the median price of a house with overall quality of 9 is on average 29.66% higher than that of houses with overall quality of 1.
    - OverallQual10: Holding everything else constant, the median price of a house with overall quality of 10 is on average 23.62% higher than that of houses with overall quality of 1.
    - OverallQual2: Holding everything else constant, the median price of a house with overall quality of 2 is on average 18.62% lower than that of houses with overall quality of 1.
- Functional most important levels:
    - FunctionalSev: Holding everything else constant, the median price of a house with severely damaged home functionality is on average 22.43% lower than that of houses with home functionality of major deduction 1.
    - FunctionalMaj2: Holding everything else constant, the median price of a house with home functionality of major deduction 2 is on average 14.05% lower than that of houses with home functionality of major deduction 1.
- LotArea: For every 1% increase in lot size in square feet, the median house price will on average increase by around 0.068% holding everything else constant.
- OverallCond3: Holding everything else constant, the median price of a house with overall condition of 3 is on average 14.35% lower than that of houses with overall condition of 1.

On the other hand, the unimportant numeric variables are LotFrontage, LowQualFinSF, BsmtHalfBath, BedroomAbvGr, and TotRmsAbvGrd. In additional, some categorical variables have majority of their levels being unimportant, including MSSubClass, Exterior1st, ExterQual, and GarageCond. Initially, I thought some of these predictors, number of bedrooms, number of total rooms above ground, and dwelling type, would be more important to a house's sale price. Surprisingly, based on Lasso, they are not that important to determine the house sale price.

7) Final conclusion and personal reflection on the project:

Kaggle House Price project is a great project to practice one's data science process, from data cleaning, exploration to applying advanced machine learning algorithms to improve model performance. Throughout this project, I had a chance to experiment with different ways to optimize the data cleaning

process, which is different from my previous projects. I started to write functions that reduce much typing effort. After applying all the machine learning methods, I proceeded to try adding interaction to the models. In particular, I identified pairs of categorical and numeric variables that are highly correlated to the other, which then would be the interaction term in my model. In the end, I could improve my chosen model – Lasso – slightly by adding the interaction between OverallQual - GrLivArea and OverallQual – LotArea. By trying different trials, I learned that interaction could improve some model while decreasing the accuracy of some others. I also noted that Simple Decision Tree method cannot handle interaction terms. In general, the project helps me reinforce my data cleaning skills and better understand the machine learning techniques. However, I believe it is not entirely representable for real-world data, where there are a lot of noises and less available information.