Final Project

House Prices: Advanced Regression Techniques

(Data source: https://www.kaggle.com/c/house-prices-advanced-regression-techniques)

Course: MGMT 6560
Section: 1
To: Jason Kuruzovich
Reporter: Baiting Gai
RIN: 661991250
Report Date: 12/13/2019

Executive Summary

Project Overview

The project aims to predict sale prices of every house according to 79 independent variables representing every aspect of the house and this will help real states better develop house design and clarify the sale price premium by investigating a practical value of one house. The given datasets have been classified to two ones, the train one and the test one which is used to test whether the model we establish applies to other data. According to the evaluation metric-Root Mean Squared Error(RMSE) and the problem attribute of prediction, I can judge this is a regression problem and I tried many advanced regression algorithms..

Learned from other solutions

The dataset shows 79 independent variables including 36 quantitative features and 43 categorical features and one target variable which is sale price. In the three solutions I have learned, I found that they both made massive independent variables processing from dealing with missing values, outliers, standardization and hemoscedasticity to label encoding and box transformation. Additionally, they combine features in train and test dataset to explore more attributes of one feature. In the feature engineering part, they use different methods to create new features like combing similar features into one feature, increasing binary values for numeric features and clustering some features. These solutions all apply more features than original features to their models. It is a hard and time-consuming task but it is worthy because it directly determines the accuracy of models. The models they use are similar, such as Ridge, Lasso, Elastic Net, support vector machine, gradient boosting, LGBM, XGBoost regression, and stacked regression. Interestingly, same model in different solutions gives different results because they use different feature engineering methods but stacking regression model always give the best performance.

Conclusions & Findings

Getting start on the project, I first did some exploration data analysis and data visualization where I selected informative features and test my assumptions with some scientific methods. Using information I found in initial processing, I created new features to do modeling. I chose five base models including Lasso, Ridge, Elastic Net, GBoost and XGBoost regression to predict house price using 254 features. The five models all have their own advantages and I explained why I choose them in the report. Lasso and Elastic Net regression model tends to select more significant features and Ridge puts emphasis on just few features and it is more sensitive to nonsignificant features. GBoost improve its performance by gradient descent algorithm while XGBoost puts gradient descent and regularization together and it considers more features in the model so that the distribution is more even. Although XGBoost gives the least RMSE score, finally I chose Elastic Net to the final model because its RMSE score is just a little more XGBoost and it gives more interpretive features. Ranked by feature importance, the features include overall quality, whether it is commercial house overall condition, total square feet and also whether house is in Somerset or College Creek. Additionally, feature selection from XGboost also shows some interesting features most related with house prices, such as whether one house has a fireplace and its quantity, whether it has a central air, bathroom quality (measured in area) and condition as well as whether it is in medium density region. As a result, I strongly recommend that property developer improve their internal quality (bathroom, garage, fireplace) to raise prices even though the houses are not in an expensive district. Likewise, for people who want to buy houses in commercial or medium density, they can choose the ones which facilities are not perfect to decrease house purchasing expenses.

Benchmarking of Other Solutions

Chart1: Comparison of three solutions

	Solution 1: House Prices: Advanced Regression Techniques	Solution 2: Stacked Regressions to Predict House Prices	Solution 3: House Prices: Lasso, XGBoost, and a Detailed EDA
Features	**Original Features** contain 79 features in total (excluding ID), including 36 quantitative features and 43 categorical features. **Categorical Features Transformation** Since none of categorical features has normal distribution, all 43 of them are transformed, thus, there are 122 (26 + 43 + 43) features in total. **Feature Engineering** Three features have been dropped, five new features have been created by combining already existing correlated features, and another five new features have been updated by adding conditions, thus, there are 129 (122 - 3 + 5 + 5) features in total. **Apply Dummy Features and Final Modification** Expanded to 376 features by applying dummy categorical features. Drop the overfit column and	**Same original features as solution 1.** **Remove Outliers** Based on observation from a GrLiveArea vs SalePrice graph, remove those instances that have GrLivArea values greater than 4000 and SalePrice less than 300000 at the same time. **Data Imputation** Select top 20 features with missing value ratios from high to low. Do data imputation on these 20 features based on data description until all missing values have been predicted and filled in. **Feature Engineering** Update four numerical features by transforming them into categorical features. Encoding some categorical variables and add one new feature that is combined from the previous three other features, result in 79 features. **Apply Dummy Features and Skewed Features** Do box cox transformation on skewed numerical	**Same original features as solution 1.** **Explore Most Important Variables** Use correlation and find the top two numeric predictors that correlated to the output the most: OverallQual, GrLivArea. **Data Cleaning** Impute 34 features with missing values, factorize the remaining character variables, and change two numeric features into factors, result in 56 numeric features and 23 categorical features. **Keep exploring most important variables** Use random forest instead and find some other most important features such as neighbor, MSSubClass, and so on. **Feature Engineering** Adding or combing new features such as total number of bathrooms, neighborhood, total square feet and so on. **Apply Dummy Features, Skewed**

	finally result in 375 features in total.	features, then apply dummy categorical features, finally result into 220 features in total.	**Features, and Final Modification** Drop seven highly correlated variables, remove two outliers, normalization on skewed features, apply dummy categorical features, and remove some features that have few or no observations, finally result into 152 features in total.
Modeling Approach	**First**, separately introduces eight models: ridge regression, lasso regression, elastic net regression, support vector regression, gradient boosting regression, LGBM regression, XGBoost regression, and stacked regression which combines the previous seven methods. **Second**, blend models by assigning different weights on each of the eight predictors and combine them all.	**First**, separately introduces six basic models: lasso regression, elastic net regression, kernel ridge regression, gradient boosting regression, XGBoost regression, LGBM regression. **Second**, stack model by averaging four basic models, lasso regression, elastic net regression, kernel ridge regression, and gradient boosting regression. **Third**, stack models by averaging three models: elastic net regression, kernel ridge regression, and gradient boosting, and add lasso regression as meta-model instead. **Fourth**, decide to use the second stacked model since it gives a better score. Blend this stacked model together with XGBoost regression and LGBM regression, assign	**First**, try out different parameter values on two basic regression models: lasso regression and XGBoost regression. **Second**, stack model by averaging these two models and weighting the lasso model double.

		weights on each model and do ensemble.	
Train and test split	Put train and test dataset together to do modeling and split them with 10-fold cross validation method.	Put train and test dataset together to do modeling and split them with 5-fold cross validation method.	Put train and test dataset together to do modeling and split them with 5-fold cross validation method.
Performance	This solution gives a smallest RMSE score of 0.10649.	This solution gives a middle RMSE score of 0.0752. (the best one in three methods)	Although the author does not explicitly indicate the final RMSE score, we can infer it should fluctuate between 0.1121 (score from lasso) and 0.1162 (score from XGBoost) since the final score averages from these two scores.

Comparing all three solutions, during the feature processing steps, I would say all three models make a lot of efforts on manipulating these features. They all finish data cleaning, feature engineering, applying dummy features, and some other modifications. However, the third solution has focused too much on dealing with these features, and finally result in 152 features being used for modeling, where the first solution has 375 features and the second provides 220. Definitely, more features would provide a better performance on accuracy while less features resulted from feature selection enables the algorithms to train faster. In addition to features, modeling is another important factor that affects the performance. It seems like each of the basic regression models gives a similar RMSLE score of around 0.1. However, compared to the first two solutions, the third one's modeling approach is quite simple: it only takes two basic models and average on them. It results in the final score not changing much, still around 0.1, which gives the worst performance. The second approach is a bit too complicated: after introducing six basic regression models, it keeps stacking models and averaging them several times, which makes the structure ambiguous and also complicated to understand. As a result, the first solution focuses more on training eight basic models, and blends models by assigning weights carefully. With the addition of detailed feature processing work, it provides the best performance among three solutions.

Data description and Initial Processing

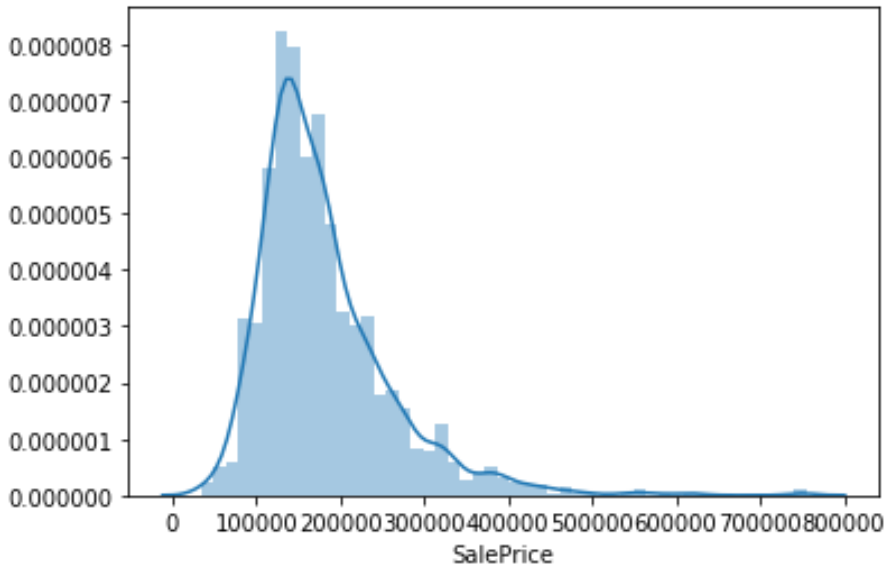
Taking a first look at the dataset, I found there are 79 independent variables and one target variable, which is the house price. Since there are a lot of independent variable, I classify the independent variables into 36 quantitative features and 43 categorical features and plan to analyze different kinds of features with different visualization and make feature selection and

creation and engineering. In this part, I will first explore and know a little bit of the target variable-house price. Then I will make some assumptions about the independent variable selection and test them with the correlation analysis. After confirming the independent features used to make models, I will do data cleaning including dealing with the missing values and outliers to better make models. The followings are some details.

Dependent Variable Description and analysis

By first examining the description of the target variable, it is a good thing that the minmum of the it is larger than zero and places in a reasonable value, which means that there is no missing value in the target value and no personal traits that would destroy the model we will establish. To further know how the value distribute, I use histogram to do data visualization and calculate kurtosis(1.88) and skewness(6.54) to show the distribution is more shape and positively skew thean the normal distribution. So we need to adjust the distribution before we making models.

Plot 1: Distribution Of House price



Independent Variable Description and analysis

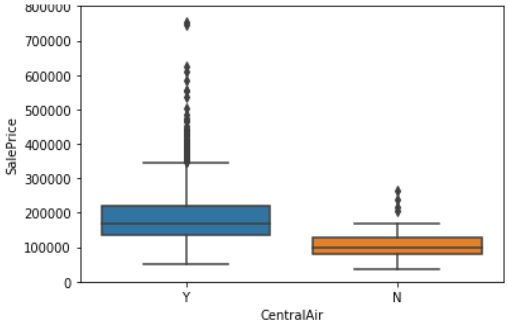
Based on my assumptions, I selected several possible variables which I list below.

- Utilities: Firstly, I guess there is a positive relationship between utilities and house price, but I find that all values of the variable is ‘AllPub’. So I will exclude this.
- LotArea: More LotArea will lead to higher house price.
- Neighborhood: House price is affected mainly by the location.
- OverQuall: Good quality and decorations lead to high house price.
- YearBuilt: Normally, new houses are more expensive.
- TotalBsmtSF&GrLivArea: More space means more expensive.
- CentralAir: Some houses have, some not.
- MiscVal: Some valuable items, like sheds which prices are in the range of $400-$1500.

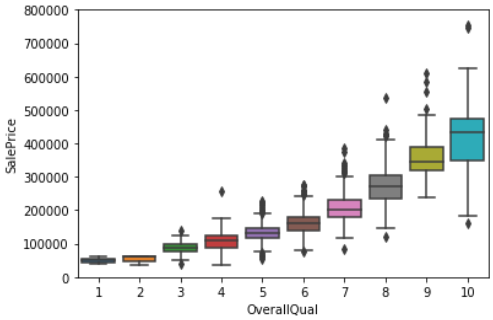
- GarageCars&GarageArea: Whether there is a garage determines whether will choose the house to a certain extent.

So the final variable I choose is that ‘CentralAir’, ‘OverallQual’, ‘YearBuilt’, ‘Neighborhood’, ‘LotArea’, ‘GrliArea’, ‘TotalBsmtSF’, ‘MiscVal’ and ‘GarageArea/GarageCars’ and I probe into the relationships between these varibales and the target variable-house price. The followings are some data visualization.

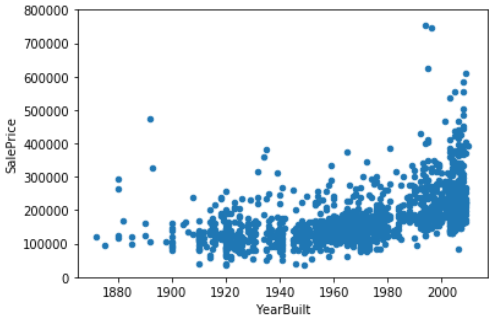
Plot 2: Boxplot between CentralAir and Sale Price



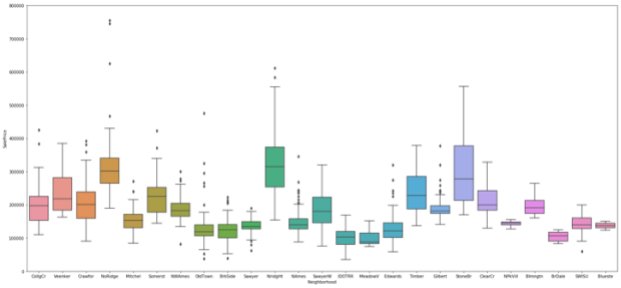
Plot 3: Boxplot between OverallQual and Sale Price



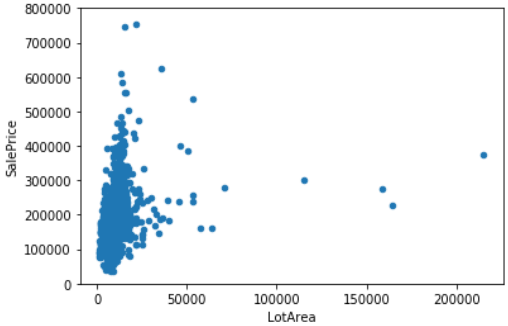
Plot 4: Scatter plot between YearBuilt and Sale Price



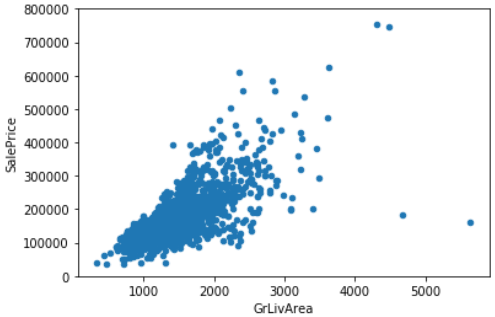
Plot 5: Boxplot between Neighborhood and Sale Price



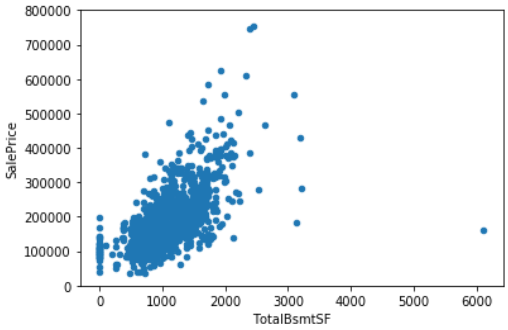
Plot 6: Scatter plot between LotArea and Sale Price



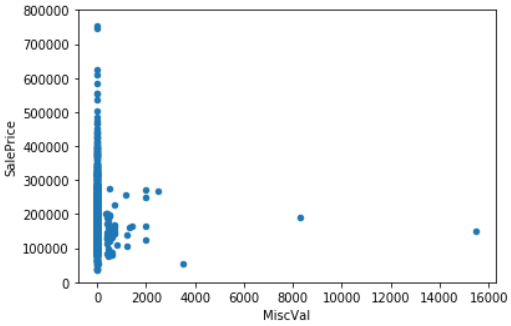
Plot 7: Scatter plot between GrLivArea and Sale Price



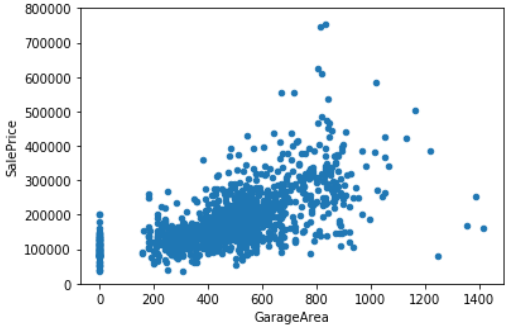
Plot 8: Scatter plot between TotalBsmtSF and Sale Price



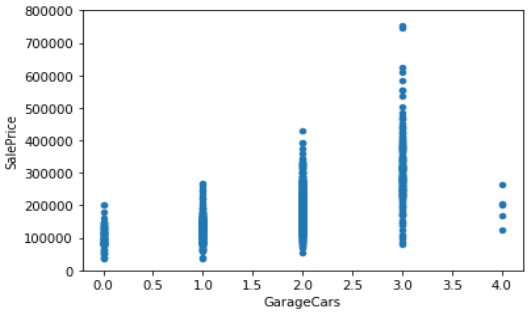
Plot 9: Scatter plot between MiscVal and Sale Price



Plot 10: Scatter plot between GarageArea and Sale Price



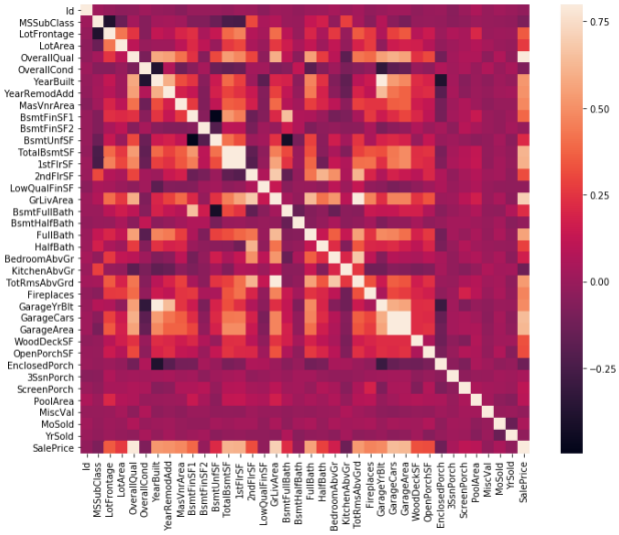
Plot 11: Scatter plot between GarageCars and Sale Price



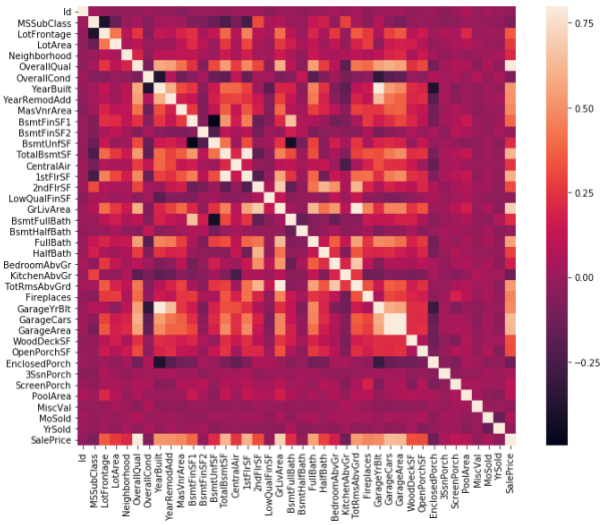
According to the above charts, I conclude a lot. First, 'CentralAir', 'OverallQual', 'YearBuilt' are positively realte to high house price. Second, 'GrliArea' and 'TotalBsmtSF' are more tenable variable than 'LotArea'. Third, 'GarageArea/GarageCars' show a approximately linear relationship with house price. Forth, 'MiscVal' is irrelevant with house price and 'Neighborhood' should be considered more in the following analysis.

To test my assumption and give a more scientific analysis, I do correlation analysis between every variables. The chart are below.

Plot 12: the correlation matrix between variables



Plot 13: the second correlation matrix between variables

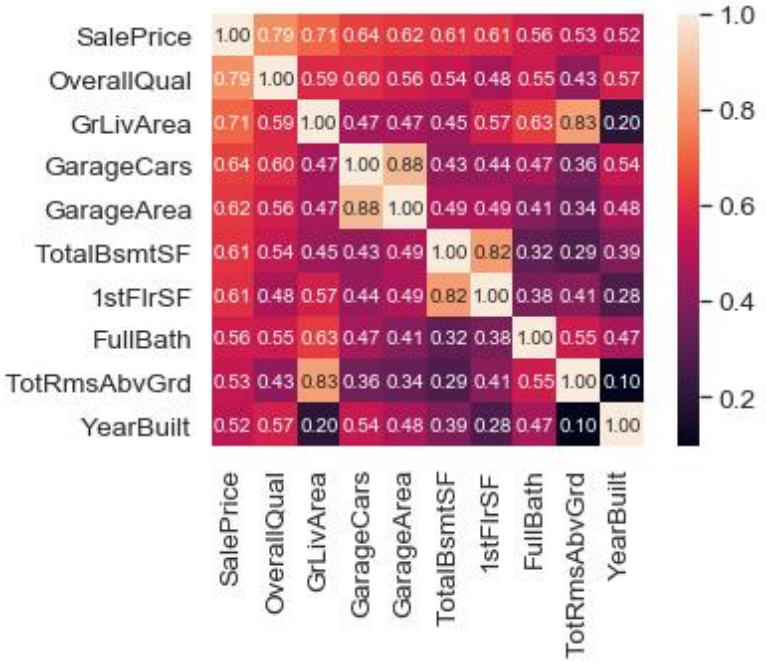


From the correlation matrix, we know that light colors mean a strong relationship between variables and dark colors mean a weak relationship between variables. So I conclude that ‘OverallQual’,‘YearBuilt’,‘ToatlBsmtSF’,‘1stFlrSF’,‘GrLiveArea’,‘FullBath’,‘TotRmsAbvGrd’, ‘GarageCars’ and ‘GarageArea’ are top variables that have a strong relationship eith houe price. These are mostly same with our assumptions, but there is also some interesting variables I did not take into consider, like the ‘Fullbath’. Additionally, I did not choose the strongest area-related areas, which can not be only selected out by institute observation and historical experience. In the following analysis, a thing that is worthy to mention is that how to choose the closer feature like ToatlBsmtSF’ and ‘1stFlrSF’, ‘GarageCars’ and ‘GarageAera’, ‘TotalBsmtSF’ and ‘1stFlrSF’ and how to do feature creation.

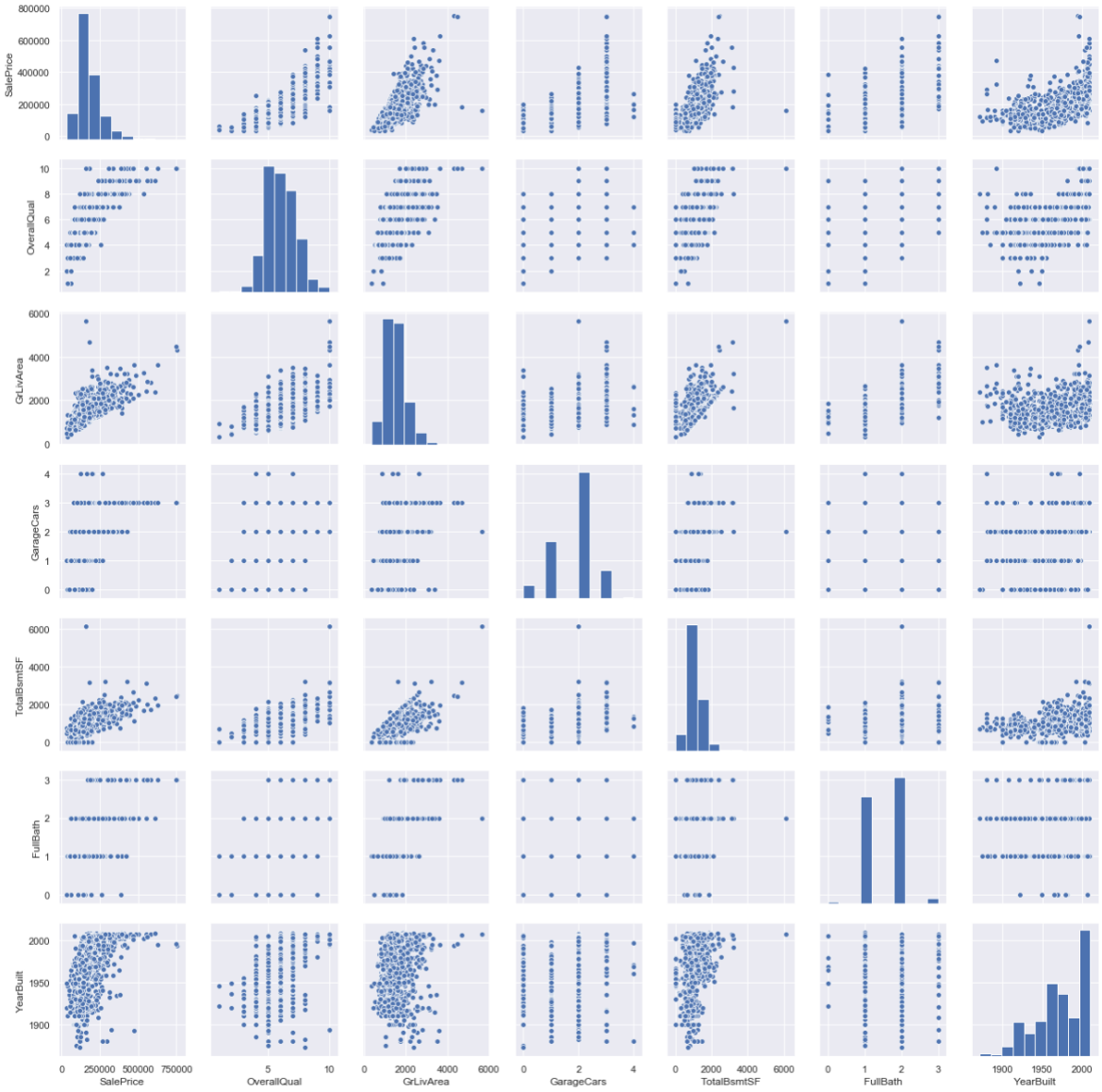
Another question is that the correlation matrix only shows that the numeric variables but we also have two categorical variables we assumed before and we should put them into the correlation matrix. There is the correction matrix after I added the ‘CenteralAir’ and ‘Neighborhood’. From this, we can see that the relationship between ‘CenteralAir’ and sale price and the relationship between ‘Neighborhood’ and sale price are weak, so I will exclude the two variables form my list.

To better know top variables, I selected the top 10 related independent variables and give specific plots between them.

Plot 14: the third correlation matrix between top 10 variables



Plot 15: Scatter plots between 'SalePrice' and correlated variables



From above plot, we can clearly know better how one independent variable relates to another one and how one independent variable relates to sales price. For those variables who have a linear relationship with another one, I will mark them and pay attention when I do feature engineering in the modelling part.

Initial Processing

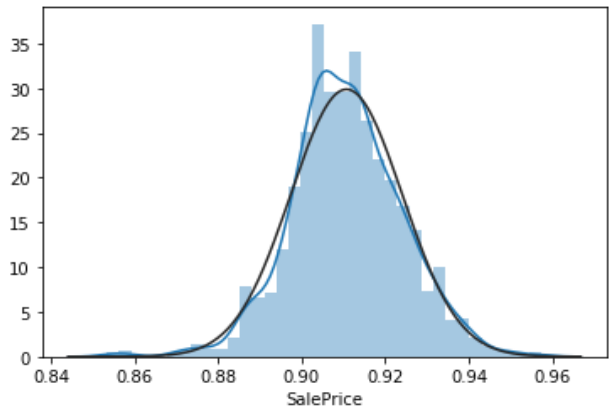
After doing exploratory data analysis, I start to deal with missing values. I selected the variables containing missing values and calculate its proportion in these variables. The details are below. From the chart, I will exclude the 'PoolQC', 'MiscFeature, 'Alley', 'Fence' and 'FireplaceQu' variables because there are exceeding 40% of the variable values are missing values. For 'LotFrontage', I will average to fill out the missing value due to its approximate normal distribution. For those who have a around 5% percent missing values, I will also fill them using average and for those who have a below 3% percent missing values, I will drop them.

Chart 2: Missing values description

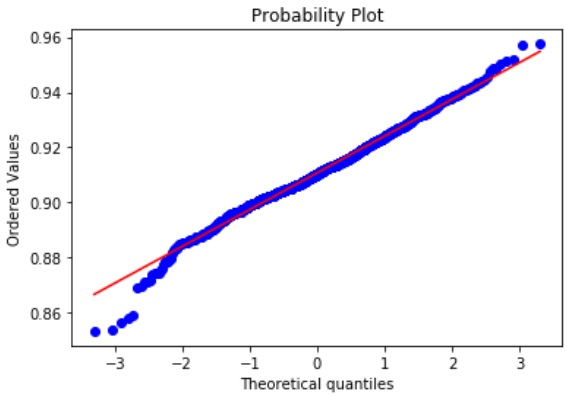
Variables	Total	Percent
PoolQC	1453	0.995205
MiscFeature	1406	0.963014
Alley	1369	0.937671
Fence	1179	0.807534
FireplaceQu	690	0.472603
LotFrontage	259	0.177397
GarageCond	81	0.055479
GarageType	81	0.055479
GarageYrBlt	81	0.055479
GarageFinish	81	0.055479
GarageQual	81	0.055479
BsmtExposure	38	0.026027
BsmtFinType2	38	0.026027
BsmtFinType1	37	0.025342
BsmtCond	37	0.025342
BsmtQual	37	0.025342
MasVnrArea	8	0.005479
MasVnrType	8	0.005479
Electrical	1	0.000685

After dealing with missing values, I will deal with outliars in ‘sales price’ which will affect model accuracy. But how to define outliars? To define it, I will first standardize the data. In this context, data standardization means converting data values to have mean of 0 and a standard deviation of 1. Fortunately, there is almost no ouliars in ‘sales price’ but there are two higher outer range of the distribution need to be careful. Like we can see in the distribution of ‘sale price’ before, we need to use log processing to normalize it. From the chart below, we can see the probability plot looks better because the points are nearly in a line. Finally, I deal with dummy variables to transform the 79 variables into 122 variables. As for the hemoscedasticity between independent variables, I will consider it into feature engineering in the modelling part.

Plot 16: Standardized Distribution Of Sales Price



Plot 17: Probablity Plot Of Standardized Distribution Of Sales Price



Modeling

In the modeling part, I will report in three parts: further data processing, advanced analysis of relevance of independent variables and comparisons of different models in terms of their performance and feature selection.

Further data processing

From the exploratory data analysis and data visulization I have done before, we know train data in a basic level. To filter more effective data and generate more information, we need further data processing including more detailed missing value imputation, creating some new features which is very beneficial for the project and also some normal steps like Label Encoding, transformation and deal with dummy variables.

Firstly, I increased data input by integrating train and test dataset to include feature characteristics as more as possible. It is useful for checking missing values and outliers in test data and consider them into establishing models. I dropped data which GrLivArea is more than 4000 or SalePrice is smaller than 300000 based on the scatter plot between GrLivArea and SalePrice.

Secondly, I conducted more detailedly data cleaning. For features which have a high ratio of missing values, I checked the reason that original data use Na instead of Na in Python. For the "lotFrontage" variable, I find there is a relationship between "Neighborhood" and "LotFrontage", so I imputed missing values with the LotFrontage median value of those house that has Neighborhood. Alternatively, I deal with missing values with different method including imputing quantitative and qualitative missing values with different null values-0 and None and choosing different imputing method for different features. In addition to dealing with missing value, I also did Label Encoding for categorical variables in numeric types.

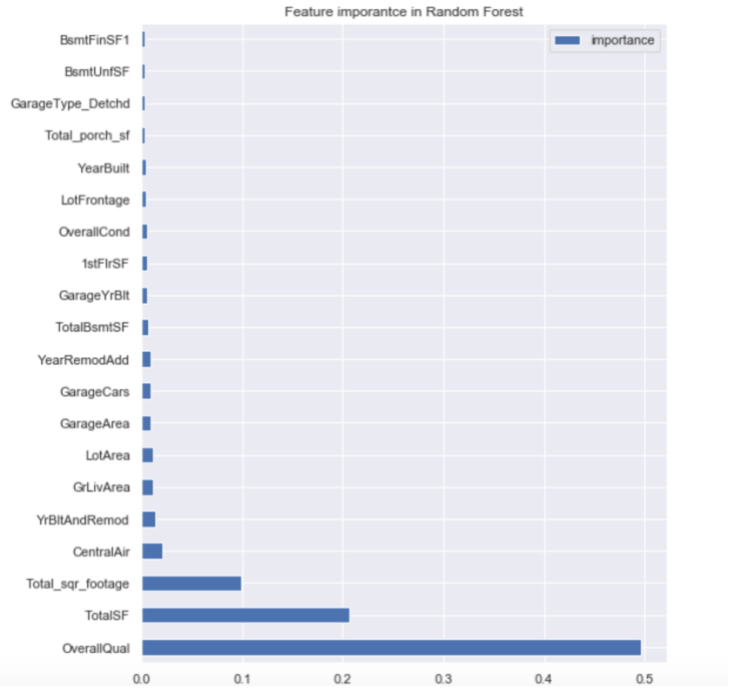
The most important thing I did in this part is to create new features based on the interpretations and actual meaning of every feature. It is essential because some features will be useful just only when combing with other features, like "YearBuilt" and "YearRemodAdd". I generated five new features-" YrBltAndRemod", "TotalSF", "Total_sqr_footage", "Total_Bathrooms" and "Total_porch_sf" based on mixing some original features. I also created some simplified features like "haspool", "has2ndfloor", "hasgarage", "hasbsmt" and "hasfireplace" which are some binary values. I created them because I want to test what drives more on one house price prediction between whether own one facility and how much or how good the facility is. One more point is that when I create new features, I did not delete the original ones because at that time, I can't tell whether the generations are correct, or whether the new features are more meaningful more the original ones.

In addition to the three things, transforming for those which skewness larger than 0.5 and dealing with dummy variables can not be forgot, which are very important processes for regression modeling. I didn't do any feature selection and any dimensionality reduction to reduce the features in this part because the observations are more than the features. I believe that more features will benefit modelling.

More Analysis of relevance of independent variable

In addition to the correlation matrix and the feature creation I have done before, I use random forest to check out the top 20 feature importance which is used to compare the selected features from the models I will establish after. From the graph below, we can see that the "OverallQual" is the most important feature(contribute 50% to the house price prediction model), followed by "TotalSF"(contribute 21% to the model) and "Total_sqr_footage"(contribute 10% to the model). The results shows 70% of a house price can be explained by overall quality and total square feet.

Plot 18: Feature importance calculated by Random Forest



Fitting Models and Feature selection

Before establishing models, we need to confirm our aim. The loss function in this project we choose is mining the errors between the root mean of errors between predict values and actual values. Another thing before real modeling is to choose a validation method to prevent overfitting. In this case, I chose kfolds method and set n=10. I defined the two process in one function which called rmsle_cv() which is used to evaluate the performance of each model.

Lasso and Ridge Model

I first chose Lasso and Ridge Regression. I choose Lasso Regression and Ridge Regression because they can prevent overfitting and select important features automatically using L2 and L1 regularization. I set alpha=0.0005 and random_state=1 for both models. That are some normal parameters and the best parameters I can tune. The rmse score mean (standard deviation) for Lasso and Ridge are 0.1111 (0.0141) and 0.1204 (0.0166). The results seem good and the Lasso performed better than Ridge. I think the reason is due to different algorithm in two models. L2 regularization is prone to not control the number of feature but work well in preventing a model put most weighs in one feature while L1 regularization is prone to minimize the number of features as more as possible and put most weighs in few features. In this case, it seems that the

features are more or less relevant to house prices. Therefore, when we set coefficients for excessive features which probably should not be 0 with 0, it seems unreasonable. the result also be explained by the fact that L2 regularization perform better in high-dimensional sparse data and L1 preforms better in low-dimensional sparse data.

Elastic Net Model

To find a balance for L2 and L1 regularization, I try Elastic Net model. How balance the L2 and L1 regularization parameters are important in the model. I choose six alphas and eight L1 ratio to see what combination gives the best performance. As a result, I choose the combination of alpha=0.0005 and l1_ratio=0.9 and it gives an awesome result 0.1110 (0.0142), which is smaller than both Lasso and Ridge.

GBoost Model

To further improve model performance, I tried Gradient Boosting(GBoost) model. Two methods which to prevent overfitting and minimizing the RMSE are regularization which I have tried befo re and ensemble models. Boosting is a learning algorithm that USES more simple models to fit d ata, and these simpler models are also known as base learner or weak learner. The method Boost use to prevent overfitting is to increase the weight of the mis-divided instances using the residual calculation in the last steps. It set the weight of correct instances with a value tends to 0. This all ows the model to more focus on the instances that were previously misallocated. Another advant age of the model is that it has more nonlinear transformation and it do not need complex characte ristic engineering and the characteristic transformation. Also, it has Strong interpretability with it s function that can automatically do feature importance order. The disadvantage of the model tha t not good for high-dimensional sparse data is not important in the project so I believe it will be a good model. But the score is not very well, which is 0.1120 (0.0176).

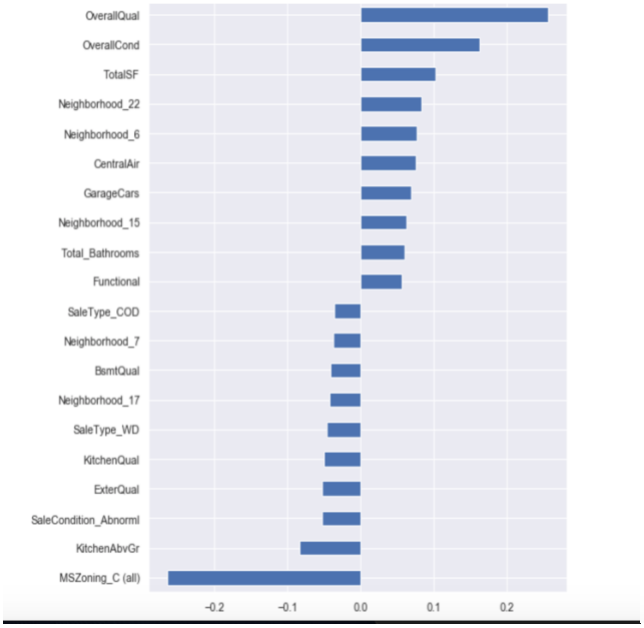
XGBoost Model

Having tried two improvement that regularization and gradient boosting, I tried a model which c an combine two of them. It is XGBoost. Traditional GBoost uses CART as the base classifier, sp ecifically referring to the gradient promotion decision tree algorithm while XGBoost also support s linear classifier (gblinear), where XGBoost is equivalent to logistic regression (classification pr oblem) or linear regression (regression problem) with regularized terms of L1 and L2 regularizati on. Xgboost adds regularization to the loss function to control the complexity of the model. The r egularization contains the sum of the number of leaf nodes in the tree and the sum of the squares of the L2 module of score output on each leaf node. From the perspective of bias-variance trade off, the regular term reduces variance of the model, making the learned model simpler and preve nting overfitting, which is also a feature of XGBoost superior to traditional GBoost. Due to the a dvantages, XGBoost gives the best performance, the RMSE score is 0.1108 (0.0161)

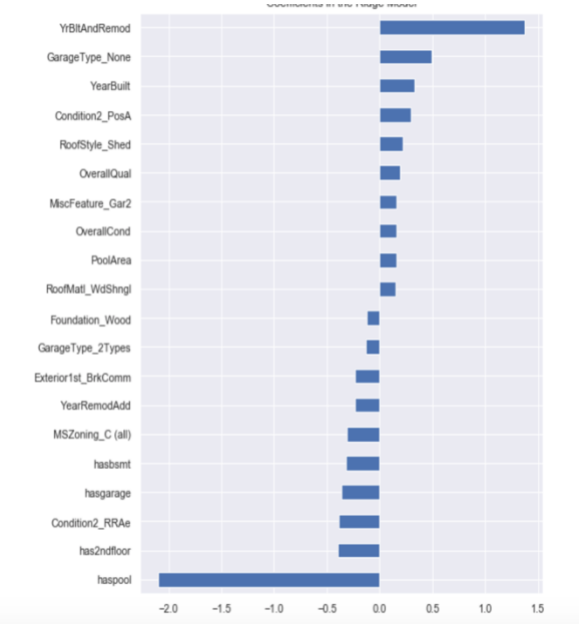
Feature selection comparisons

The following graphs provide the most important features in each models. In the lasso, ridge and elastic net models, I use the absolute value of coefficients of each feature to evaluate the feature importance. In the XBoost and XGBoost models, I use the feature importance which are one of the attributes of the models.

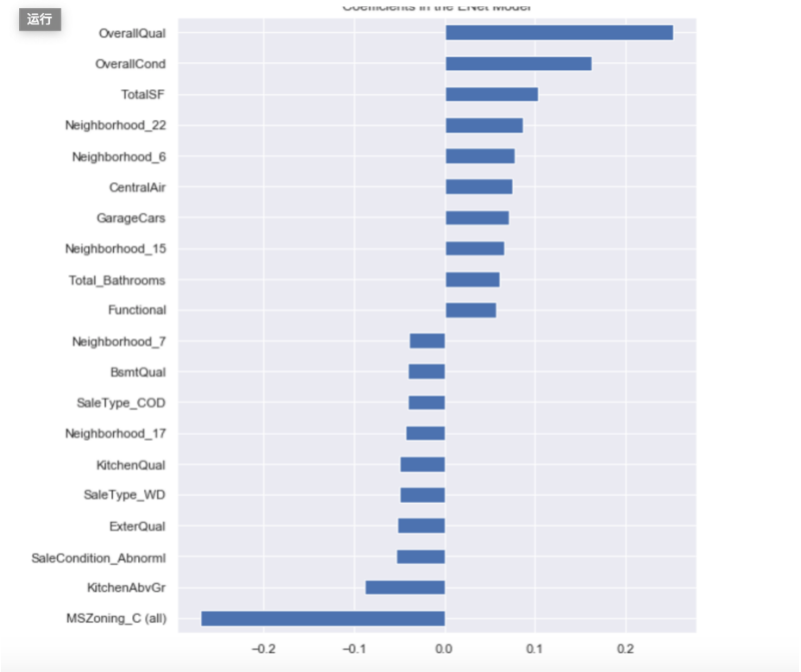
Plot 19: Feature importance calculated by Lasso Regression



Plot 20: Feature importance calculated by Ridge Regression



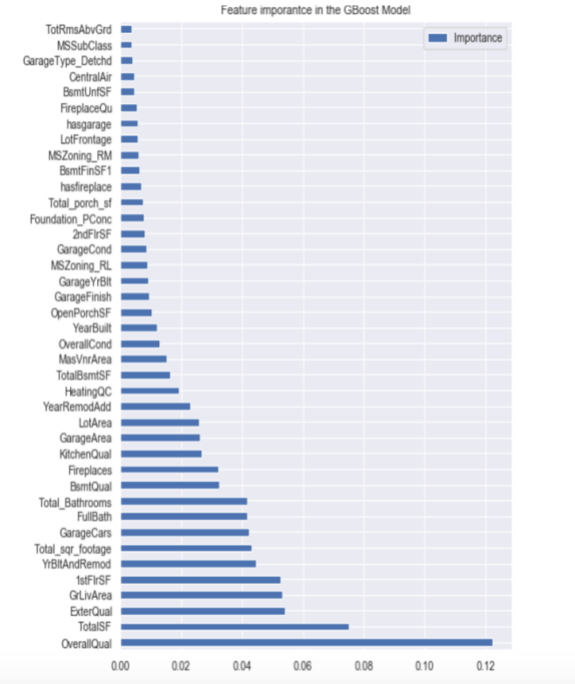
Plot 21: Feature importance calculated by Elastic Net Regression



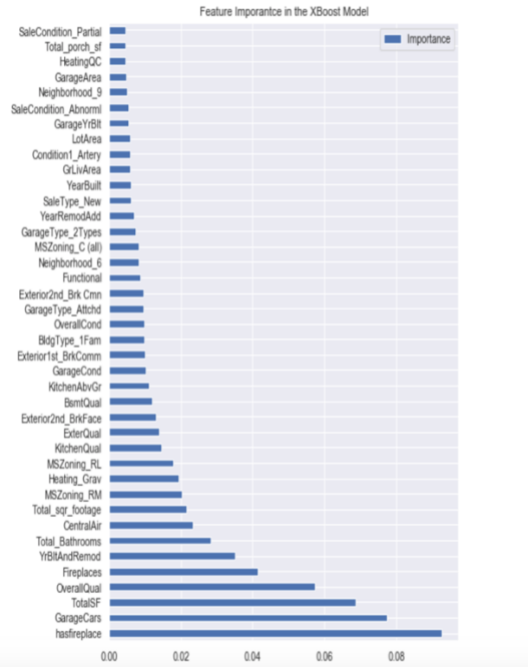
I put the three graphs together because they all use regularization to adjust the models and select the features. We can obviously find that the features of Lasso and Elastic Net model are almost same and Ridge looks a bit difference. From the shape of the bar chart, we can find Lasso disperses importance to several features while Ridge put the emphasis on just one positive variable(“YrBltAndRemod”) and one negative variable(“haspool”) and lead to an inaccurate

result. The feature selections in Ridge are very different from feature importance calculated by random forest while the most few important features in Lasso and Elastic Net are same with the result of random forests.

Plot 22: Feature importance calculated by Gradient Boosting



Plot 23: Feature importance calculated by Extreme Gradient Boosting



The above two graphs give the feature importance in GBoost and XGBoost. Although XGBoost algorithm derives from GBoost algorithms, but they perform differently. The distribution of feature importance in XGBoost looks more even, which means it considers more features and balance the ratio. Checking the most 10 important features in each model, we can find that house quality(internal and external quality) and house square feet(total area, bathroom area, also garage area) contribute more to house prices in GBoost model while XGBoost consider whether has a fireplace is the most important features, followed by garage cars, total square feet and overall quality. XGBoost also considers “YrBltAndRemod” (the combination of built year and rebuilt year), fireplaces, central air, medium density resident zoning which are some features other models did not pay attention.

Chart3: RMSE comparison of different models

Models	RMSE mean(standard deviation)
Lasso	0.1111 (0.0141)
Ridge	0.1204 (0.0166)
Elastic Net	0.1110 (0.0142)
XBoost	0.1120 (0.0176)
XGBoost	0.1108 (0.0161)

To better compare performance of different models more intuitively, I make the results into one table above. Although XGBoost model gives the least square mean error, the result is just a little better than Elastic Net model. Considering the complexity of the model and the computing time,

I will tend to choose the Elastic Net model. However, XGBoost provides us a lot of feature selection ideas and we can do more work on feature engineering to improve our models. Combining features in different models, I selected some features I think the most important. They are overall quality, whether it is a commercial house, overall condition, total square feet and also whether house is in Somerset or College Creek. Also, whether a house has a fireplace and its quantity, whether it has a central air, bathroom quality (measured in area) and condition as well as whether it is in medium density region all important decision variables. According that, I strongly recommend that property developer improve their internal quality (bathroom, garage, fireplace) to raise prices even though the houses are not in an expensive district. Likewise, for people who want to buy houses in commercial or medium density, they can choose the ones which facilities are not perfect to decrease house purchasing expenses.