Part I Executive Summary

The House Prices: Advanced Regression Techniques project is aim to predict the final price of residential homes in Ames, Iowa. As a supervised learning, this project uses explanatory variables and would put regression techniques in class into real-world practice.

The result is evaluated on RMSE between the logarithm of the predicted value and the logarithm of the observed sales price, which neutralizes the potential difference between errors from cheap houses and their expensive counterparts towards the final result.

The data we use is divided into train set and test set. The 79 different variables of the data comprehensively describe multiple characteristics of house, including size, exterior, facility, age and structure, etc. These characteristics' influences toward housing price needs to be examined.

After initial understanding of the problem and the data set, I need to do features engineering, including imputing missing values, transforming categorical data which has numerical types, logarithm transformation to deal with non-normal distributed variables and getting dummy variables for all categorical features.

Then, I can create naive model, base model and prediction.

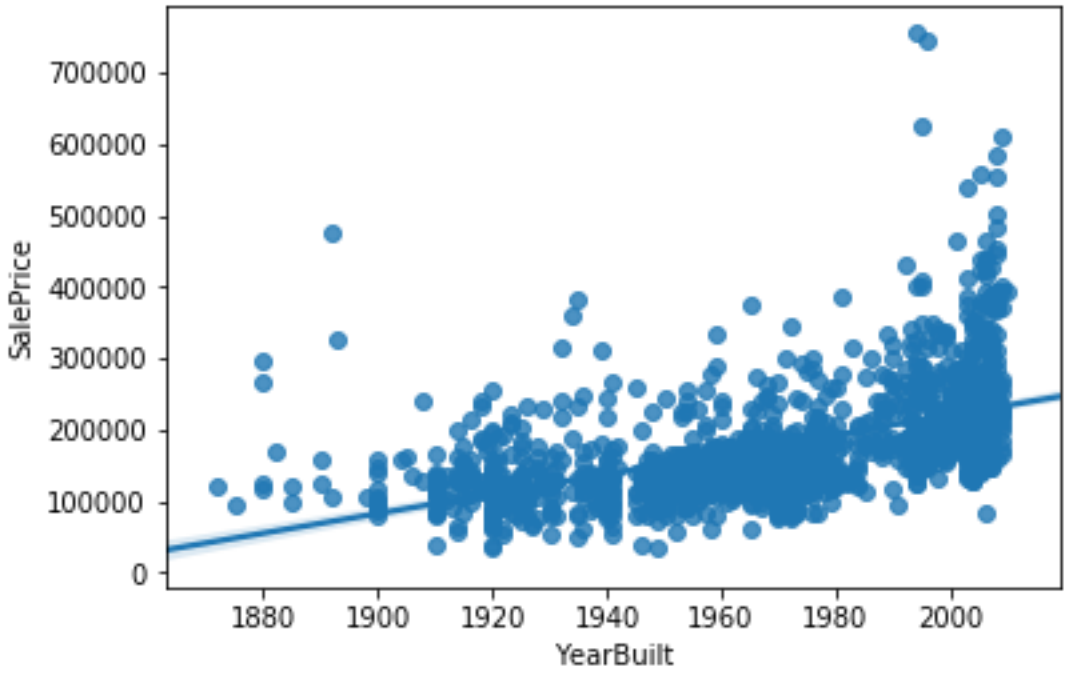
Part II Data description and initial processing

At the beginning of the project, it is time to do exploratory data analysis, including data quality assessment, data explore, relation between variables and variable and target.

After description analysis, I find out that there are 1460 rows in train set and 1459 rows in test set. As for the 81 attributes, 36 of them are numerical and 43 are categorical plus id and saleprice.

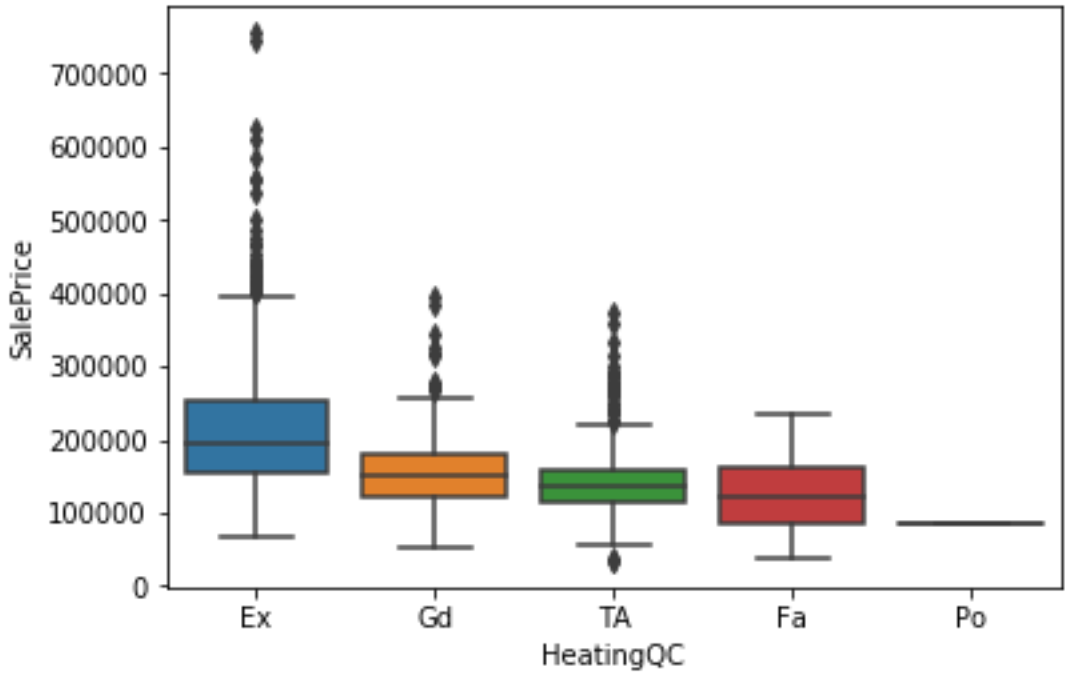
However, some of the numerical variables are de facto categorical because their value actually represents rank (e.g. MSSubClass identifies the type of dwelling involved in the sale by arrays.). This problem needs to be dealt with in the following process.

For basic statictics, I will take YearBuilt and saleprice and their relation as example:



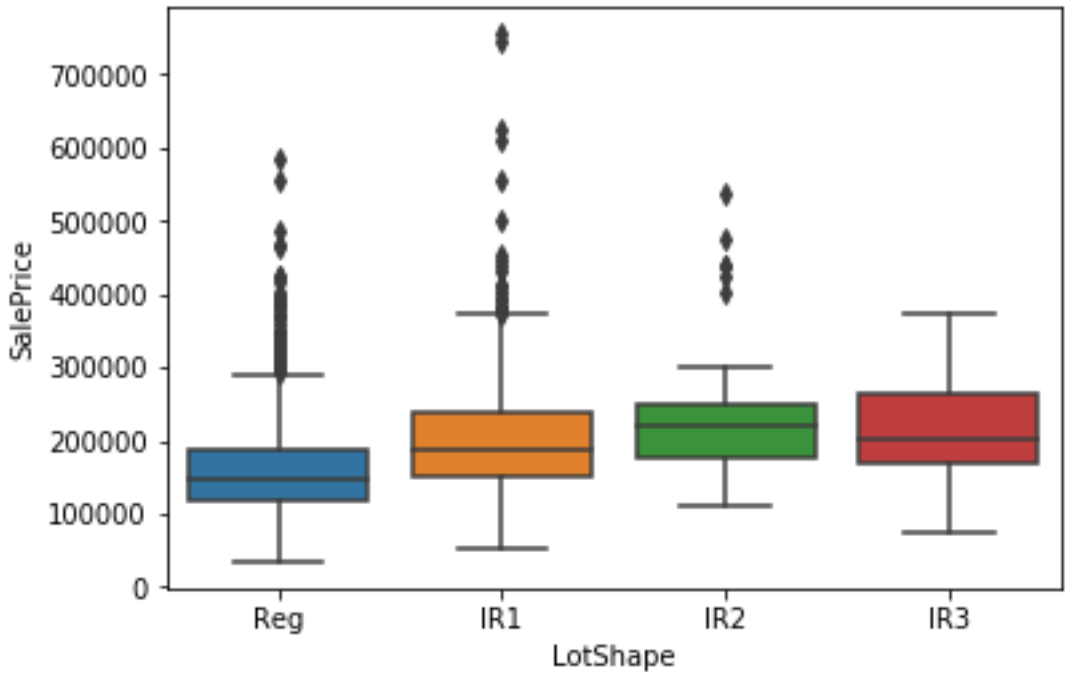
As can be seen from the regression plot, a clear positive correlation between these two variables exist, and this result fits common sense.

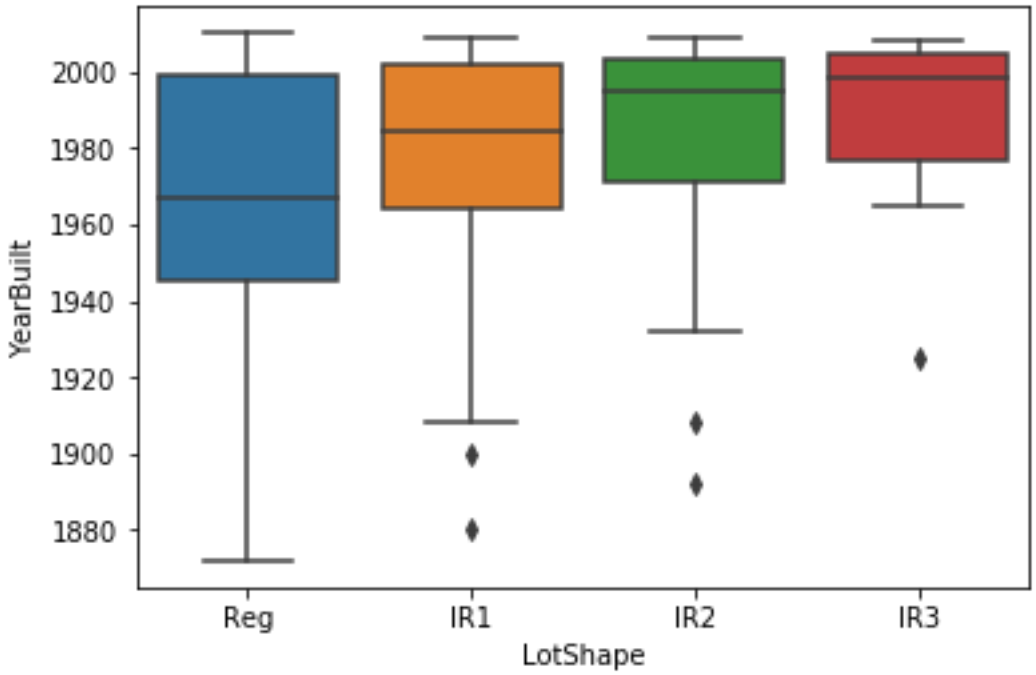
On the other hand, the relation between categorical variables and saleprice is important, too. HeatingQC is one of them.



The graph shows that houses with high condition of heating tend to have higher price.

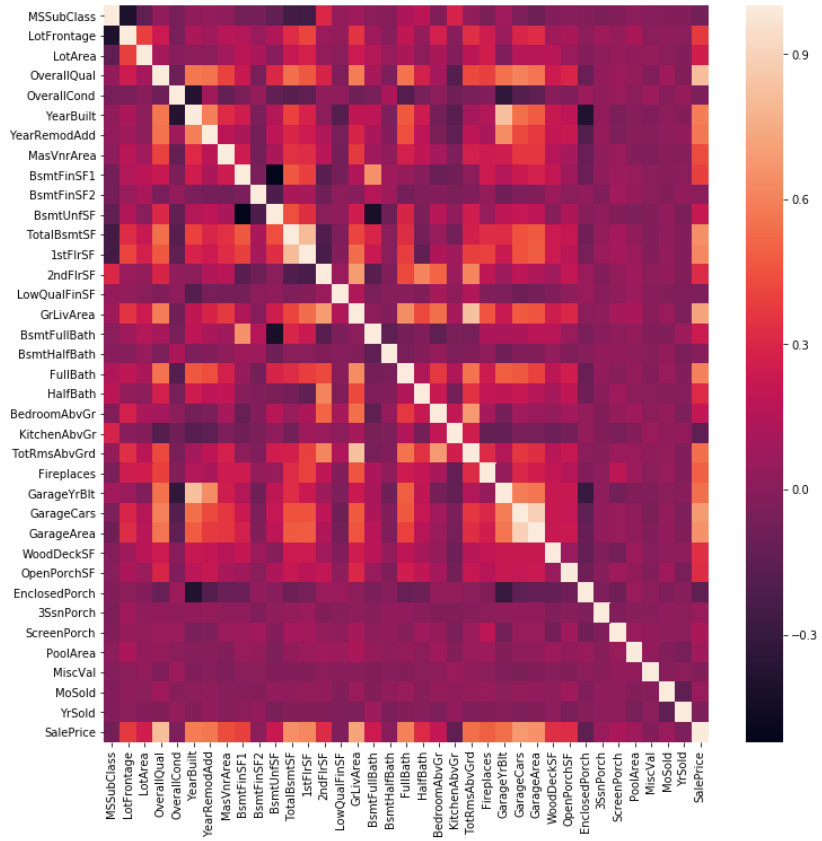
Although most simple correlations make sense, sometimes strange thing happens. For example, the relation between house shape and price is interesting, as can be seen below.





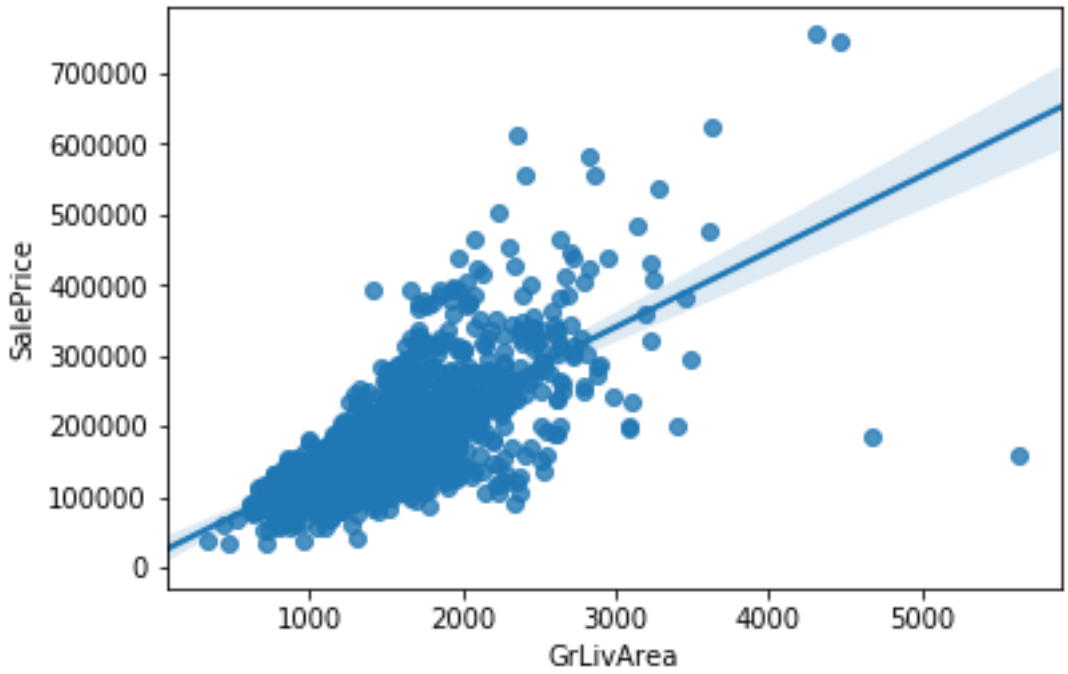
Then I realize that irregular houses are relatively new, which explain why they are more expensive.

Of course, the correlation between variables are important. Therefore, I create a correlation matrix.



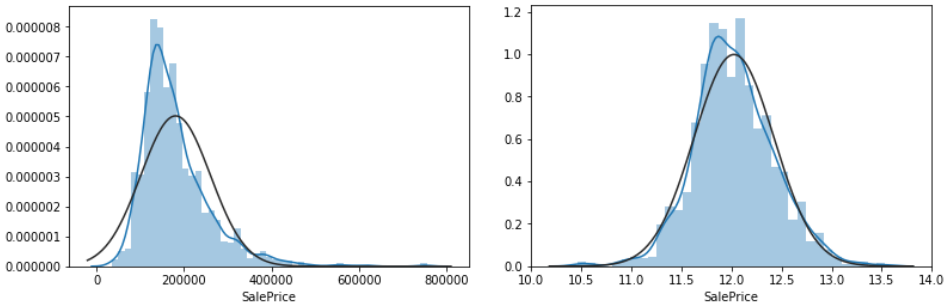
Although high level of negative correlation does not exist, there are some serious positive correlation between two variables (e.g. garageyrblt and yearbuilt, for obvious reason). Clearly, multicollinearity may be a serious issue.

Next step is outliers. It is commonly believe that the most influential variable for housing price is size, I make the following graph.



On the lower right corner, there are two outliers, so I drop them.

Then I deal with normality because almost all regression models assume variables involved are normally distributed. After I find saleprice is not normally distributed, it is logarithm transformed. The following graph describes that variables before/after transformation.



Finally, I have to fill null values and set data types. For some variables (11 in total), null represents zero or none, so I replace nulls with corresponding values. For the 8 variables left, I fill them with most frequent value (mode). Then, I transform 5 numerical variables to string, because these numbers are actually ranking. After that, I create dummy variables and data set is ready.

Part III Modeling and evaluation of 3 other solutions.

Kernels name	Stacked Regressions : Top 4% on LeaderBoard
author	Serigne
features	Droping outliers. Log-transformation of the target variable. Imputing missing values. Box Cox Transformation of (highly) skewed features. Transforming numerical variables are really categorical Label Encoding. Getting dummy variables for categorical features.
modeling approach	Base model: LASSO Regression, Elastic Net Regression, Kernel Ridge Regression, Gradient Boosting Regression, XGBoost, LightGBM. Stacking: ENet, GBoost, KRR and lasso. Meta-model Ensembling model
performance	0.11420, top 4%

Kernels name	Regularized Linear Models
author	Alexandru Papiu
features	Transform the skewed numeric features by taking log(feature + 1) Create Dummy variables Replace the numeric missing values with the mean of their respective columns
modeling approach	Lasso and Ridge regularization. xgboost model Feedforward Neural Nets
performance	0.12096

Kernels name	House prices: Lasso, XGBoost, and a detailed EDA
author	Erik Bruin
features	Correlation graph. Missing data imputation. Data type convert and one-hot encoding. Label encoding Create new features including Total number of Bathrooms, House Age, Remodeled, and IsNew. Binning Neighborhood variable. Dropping highly correlated variables Removing outliers Normalizing skewed variables
modeling approach	Lasso regression model XGBoost Averaging predictions
performance	0.11533

Part IV Modeling

For modeling part, I begin with the naïve model for which I assign the mean price of train data set as the predicted value of test set. As an initial approach, the Kaggle score of this naïve model is 0.41901, far greater than any reasonable result, which is just as expected because prices for different houses may significantly differ, due to their own characteristic including size, quality, location, etc., and pricing them equally is clearly not a good idea.

My next approach is ridge regression. Comparing to simple regression, Ridge regression adds a L2 regularization to loss function, making the model more robust without losing any feature. During application, I create a for loop to select the best value for alphas, which turn out to be 5, and the corresponding rmse_cv score is 0.11889.

Then I try lasso regression. Lasso regression adds a L1 regularization to loss function, decreasing the coefficient. With the optimal alphas is set at 0.00025 and 153 variables left, the rmse_cv score is 0.11646. The fact that the result of lasso is better than that of ridge can be seen commonly across others` kernel, and I think the reason behind is the result of lasso is more generalized, which is a important advantage for prediction.

My last model is xgboost. XGBoost stands for “Extreme Gradient Boosting”, which is used for supervised learning using CART tree ensembles and is used vary common. XGBoost uses Taylor’s expansion to approximately calculate objective function which is only rely on first-order derivative and second-order derivative of each data point on error function. XGBoost want to minimize prediction error and model complexity by greedy strategy and optimization. After parameter tuning using GridSearchCV function, I find the best parameters combination and the GBM.best_score_ is 0.90492.

Then I get the prediction value for test set from every model and use exponential transformation to recover them to their true value.

The final score of my own ensemble model which is the weighted average predicted value of all three models above is 0.12747 while the naïve model gets 0.41901. Based on statistic theory, ensemble is a good way to decrease variance and it successfully improves my final performance.

The table below is a brief summary of the model I made:

Model name	parameters	performance
Naïve	N/A	Kaggle score: 0.41901
Ridge	Alphas = 5	Rmse_cv: 0.11889
Lasso	Alphas = 0.00025	Rmse_cv: 0.11646
Xgboost	learning_rate= 0.1, n_estimators=350, max_depth= 2, min_child_weight=3, seed=0, subsample=0.7, colsample_bytree=0.9, gamma=0, reg_alpha=0.05, reg_lambda=1	GBM.best_score: 0.90492
Ensemble model	= 0.3 rigde + 0.35 lasso + 0.35 xgboost	Kaggle score: 0.12747

Part V Summary

My final score ranks about 1500 among all others in this Kaggle project, which is acceptable although not as good as I expected. Putting the knowledge and application skills learned on the class into regression practice is a good start, and there are many fields I can try to further improve my performance.

For skewed numerical variable, Box Cox Transformation may be a good replacement of log-transformation. Box Cox transformation is a kind of power transform, which can be used to decrease the correlation between error and prediction variable, improve linearity, normality and homoscedasticity. This transformation is based on data itself, do not require any prior information.

For many categorical variables, label encoding is a useful approach because it contains sequence information than dummy transformation.

Filling with mode or mean of available non-null value is a direct and sometimes, rough way to deal nulls. By seeking the hidden relation between variables and applying more advanced technic, these null values can be predicted and better filled.

When the data is set and it comes to modeling, parameter tuning is the most important and time-consuming job to do. Although CV function is available for many models, the determination of best parameters is still complex and require computer performance. One disappointing discover is, while those who perform extraordinarily detailly comment their early stage code, they don`t say anything about how they find their parameter with 5 decimals.

Link:
https://github.com/techfundamentals-fall2018/project-liuzijian1994