Prediction of the Housing Price in Cook County
Data1030 Project — Bohao(Jacob) Wang
https://github.com/bwang98/data1030_project

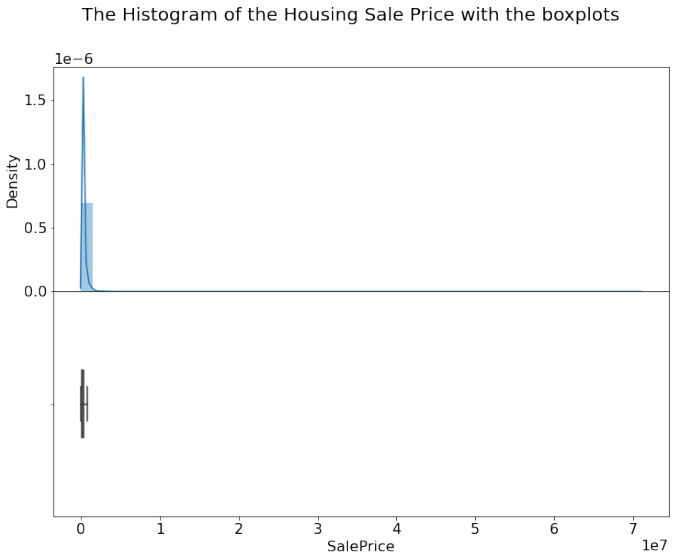
I. Introduction

While the housing price in the United States has exploded exceedingly for the recent decade, as an essential factor for the property tax assessments, the prediction of the housing price is always the priority and the focus of the government and residents. This project will be based on the housing data set in Cook County, Illinois and to the extent possible, will implement the regression model in order to achieve the goal of the housing price prediction. Over the past few decades, people in Cook County have come to realize the traditional way that the government used to estimate the housing price is lagging behind, and with the rise of data science, the Cook County Assessments Office(CCAO) started to consider the feasibility of adopting the machine learning techniques. it is the authenticity and the completeness of this data set that drives my enthusiasm to mine deeply into this topic.

In this data set, there are totally 204,792 data points with 62 features, this large data set provides me an opportunity of building some relatively more accurate models, but the difficulty of manipulating and processing the data would also be elevated in this case. The description of all these 62 features is attached in a separate txt file, named codebook.txt. One may be access and read this under the folder, data, on my Github Repo. The Federal Reserve Bank of Chicago has used this data set to analyze the effects of housing structural characteristics on residential price, the effects of proximity to amenities and of distance from CBD, and other relations[1].

II. Exploratory Data Analysis

This section include some EDA figures and the caption explains what I do in order to do visualizations better.



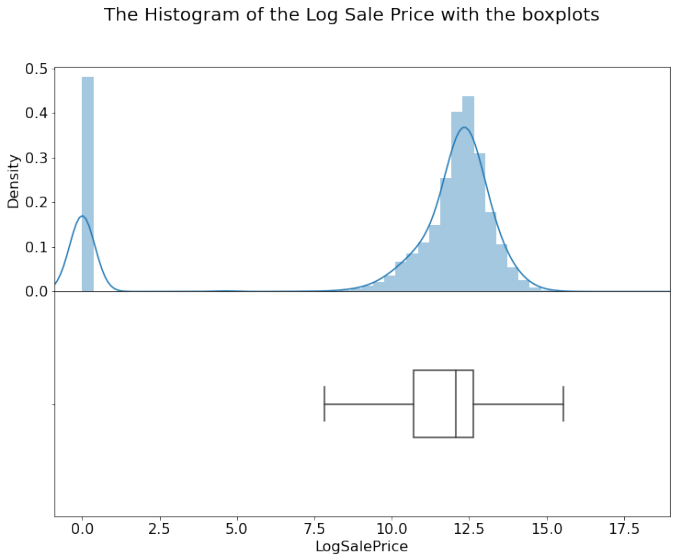


Figure 1 The left figure shows the histogram and the box-plot of the original target variable, sale price. However, as one could see, the plot doesn't reflect any useful information. Also, the plot shows that the distribution of the sale price is highly skewed. Thus, the log transformation will help us reduce the skewness and would center the data like a normal distribution. The right figure shows the plot after Log transformation. One could see that the LogSalePrice of most of the data are crowded between 7.5 to 15.5, but some are below 2.5. This dataset is imbalanced based on the indicator whether the LogSalePrice is below 2.5, above 7.5 or between.

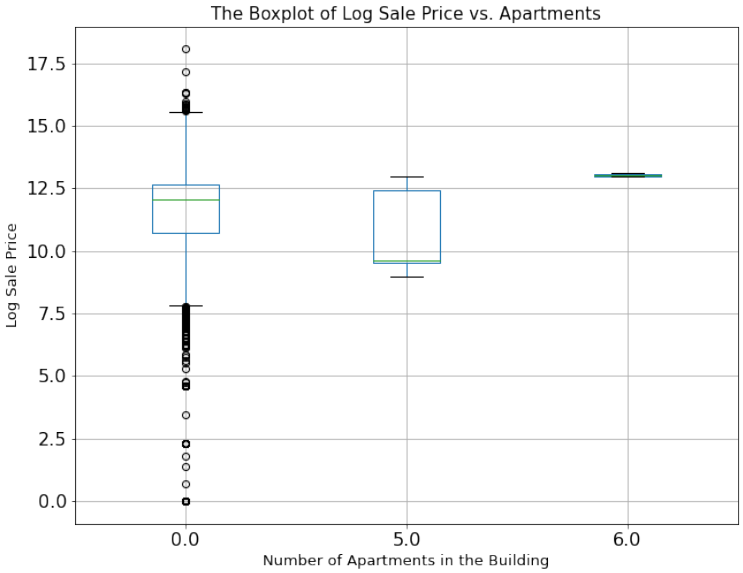


Figure 2 This figure visualizes the box-plot of the LogSalePrice vs. the number of apartments in the building, one could see that the target variable varies a lot based a lot based on the number of apartments in the building. One has evidence to claim that the number of apartments in the building, which is encoded as the variable, Apartments, is a good feature to be included in our regression model as the mean for different number of apartments

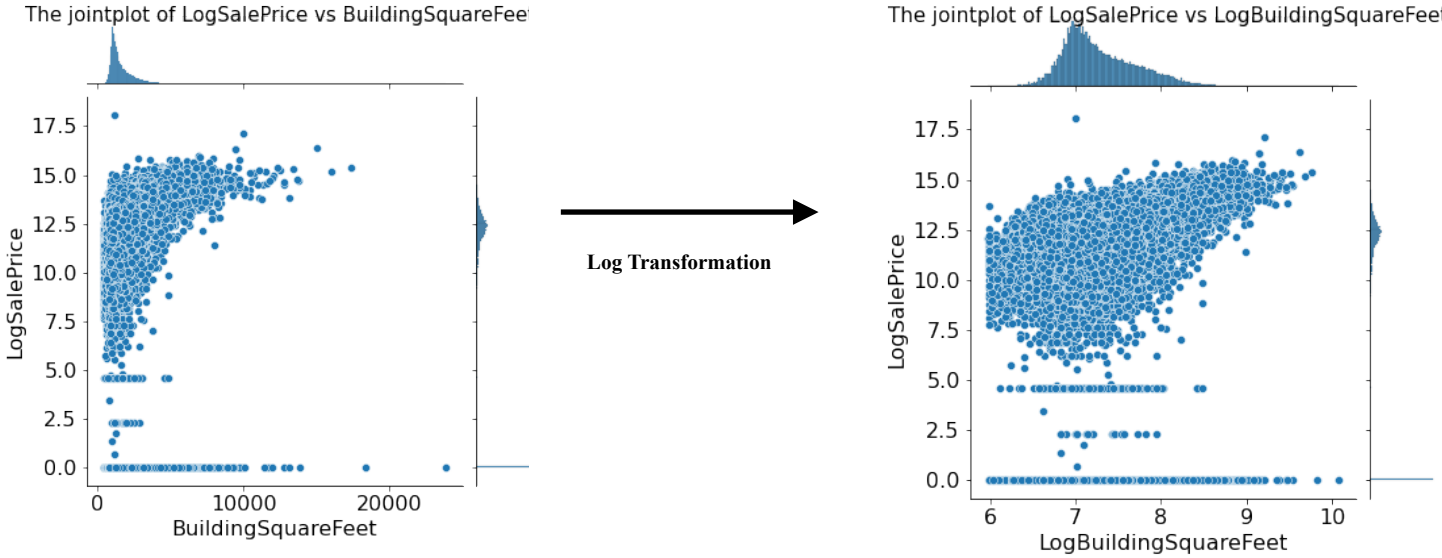


Figure 3 The left plot visualizes the the joint plot of the target variable, LogSalePrice vs BuildingSquareFeet. The left plot seems that there exists a parabola relations between these two variables. Since the value of BuildingSquareFeet is squeezed between 0 and 10000, log transformation is applied on BuildingSquareFeet. The right plot shows the corresponding joint plot after the Log transformation. And the relation is more convinced

III. Methods

A. Data Splitting and Preprocessing

Before doing the data splitting and preprocessing, the above histogram of the housing price after Log transformation inspires me that there might exist some outliers in the dataset. So I use the interquartile rule(IQR) to detect the outliers, and after dropping the outliers, this data set now haas 168757 data points.

As discussed in the EDA section that we could divide the value of the target variable, LogSalePrice into three groups: less than 2.5, between 2.5 and 7.5, and greater than 7.5. By doing so, we could see that this data is imbalanced, so we need to apply the stratified K Fold splitting strategy. Since the size of this dataset is really large, I could split the data with a proportion of 70% of the training set, 10% of the validation set, and 20% of the testing set, and I decide to set the number of folds to 5. Since each observation represents one data, and one data only has one observation in the dataset, so the data is iid. Also, the original target variable has no group structure, but I introduce the Grp_LogSaleprice to the data, which is a group structure. This data set doesn't have the time series data.

Based on the description, it is not necessary to do preprocess on PIN, DeedNo., Description, CensusTract, SiteDesirability, ModelingGroup.BasementFinish,,, AtticType, AtticFinish, ConstructionQuality, Floodplain, are ordinal features since their orders have real meanings. For other categorical data such as Use and O'HareNoise, I should use one-hot encoder because the orders make non-

sense. The histogram of Longitude and Latitude have value boundaries and don't have long tails, so I apply min-max scalar. For the rest of the features, I apply standard scalar. There are 1380 features after doing preprocessing.

B. Models, Hyper-Parameters, and Metrics

After splitting and preprocessing data, four machine learning models were applied to do the data: Lasso Regression, Random Forest Regression, Ridge Regression, and Support Vector Regression. For each of the model adopted, it is tuned with hyper-parameters by using GridSearchCV and tested with 5 different states. Figure 4 shows you the parameters being tuned for each model

Model	Grid Parameters
Lasso Regressor	Alpha: 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3
Random Forest Regressor	n_estimators: 10, 100, 300, 500, 1000 max_depth: 10, 30, 50, 70
Ridge Regressor	Alpha: 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3
Support Vector Machine Regressor	C: 1e-2, 1e-1, 1e0, 1e1, 1e2 Gamma: 1e-2, 1e-1, 1e0, 1e1, 1e2

Figure 4 Parameters tuned for each Regressor Model

After fitting all models with all the tuned hyper-parameters to the training sets and selected from the cross-validation sets, the best model for each regressor is selected. Then it is tested on the testing sets. The metric used to compare is MSE(Mean Squared Error), so the lower this score is, the better the model fits the data. MSE is the best for the regression, it is used to check how close predictions are to actual values. Figure 5 shows you the MSE score for all regressors. The uncertainties are due to the random state. For different random states, MSE score varies. This uncertainty is measured by the standard deviation of the MSE score of five random states each model fitted on, and it is approximately 0.04.

IV. Results

A. Model Selections

The baseline for the regression model is the average of the target variable, the LogSalePrice, which is 12.17236. The baseline score, MSE, is 0.987458, and the standard deviation is 0.6527856. Figure 5 compares the MSE scores of all the models being trained and the baseline score. From the plot, we could see that all the MSE scores of these four models are much lower than the baseline. Indeed, all the models are almost one standard deviation below the MSE scores. Among all these four models, the scores for the Ridge Regression is 0.92 standard deviation below the MSE scores. As the baseline MSE score is extremely low, these four model are fairly acceptable. The ridge model is the best regressor as it has the lowest MSE, which has the parameter of Alpha value, 1.0.

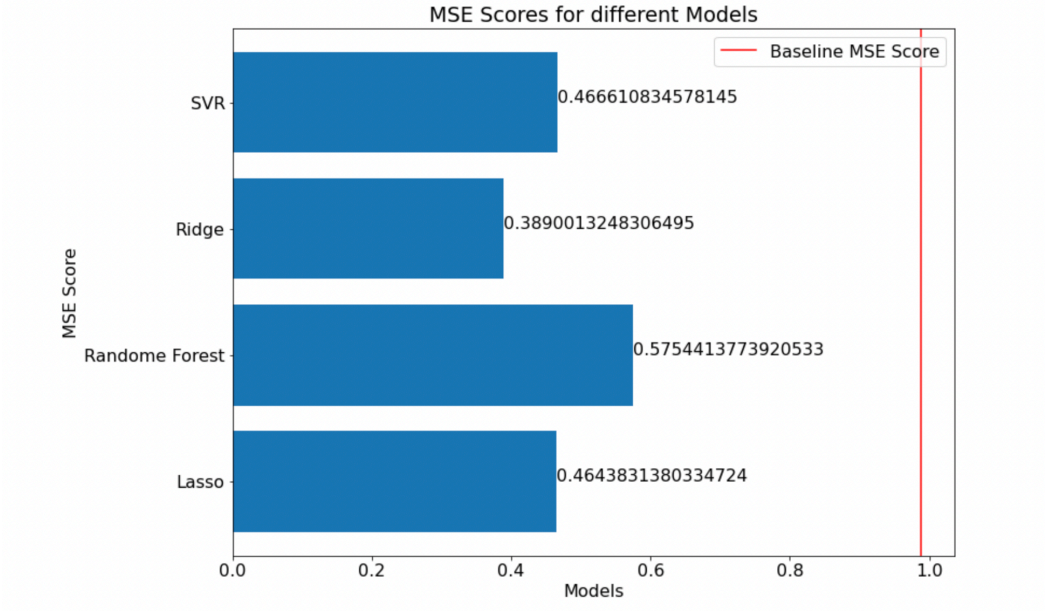


Figure 5 MSE Scores for Lasso, Random Forest, Ridge, and SVR. The red horizontal line indicates the baseline MSE score

B. Feature Importances

Different global feature importances statistics reflects different things, and the three global feature importances I choose are the coefficients of the ridge regression, permutation importances, global shap value. Figure 6 shows the coefficient of my ridge regression model as the coefficients of the linear regression represent the importance of each predictor is to the prediction of the target variable. Among the top 20 most important features, there are 16 of them are the encoders of TownandNeighborhood and two of them is related to TownCode and NeighborhoodCode. In this sense, it implies that the location of the housing has a strong influence on the housing price predictions. This is confirmed by the permutation importances measurement, the permutation importances measurement indicates that TownCode and TownandNeighborhood are the top two most important features to the housing price predictions. In addition, the permutation importances measurement also represents that the Latitude, the LogBuildingSquareFeet, the PureMarketFilter, and the MostRecentSale are the next four essential predictors. This reflects that the area of the building and the market history and evaluation play a decisive role in the predictions of the housing in Cook County. The SHAP values convince me more as they match the findings above, except that the PropertyClass_203 and the CentralHeating_0 are included. For the local feature importances, the SHAP value plot of the TownandNeighborhood_39280 shows that the housing predictions are very dependent on this indicator. Also, the value of 1 increases the housing price.

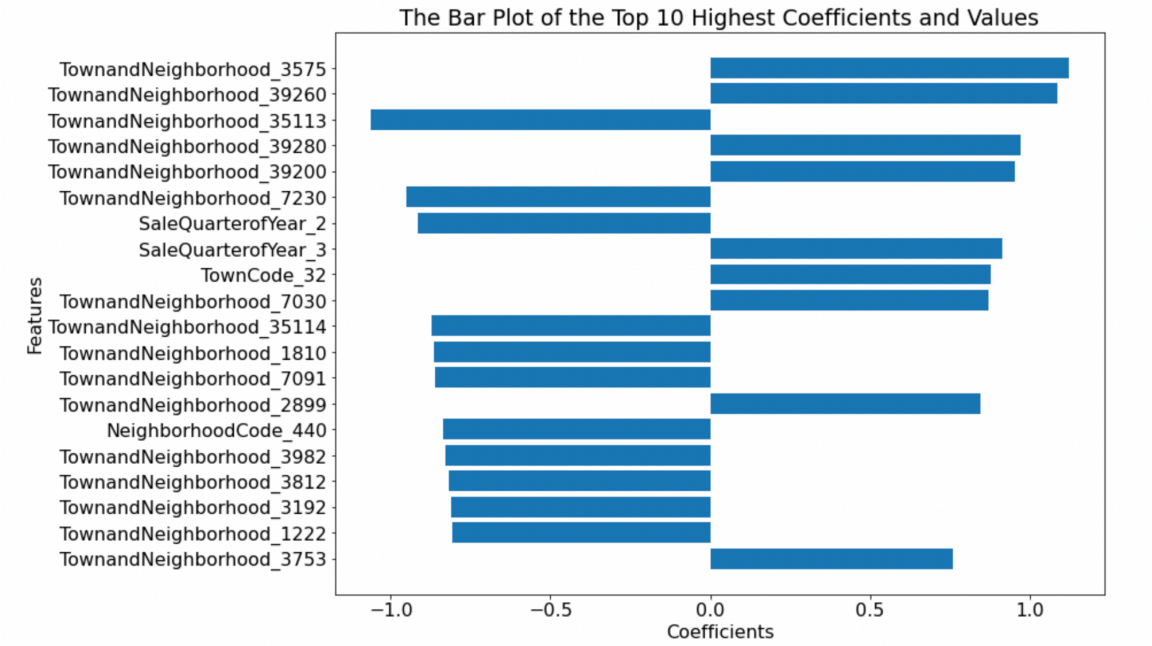


Figure 6 The bar plot of the top 10 highest coefficients and values. The length of each bar represents the values of the coefficients of the Ridge

These observations indeed make sense as people would definitely consider the location, the size, and the space of the houses as well as the market evaluation when they price and quote. However, it surprises me that location is more important in predicting the housing price in Cook County than the size.

V. Outlook

The Ridge Model I adopted perform really well in the housing price predictions. Nevertheless, there are still a couple of perspectives I could focus on in order to improve the model. First of all, I have checked the outliers before preprocessing the data with the IQR scores. This might be a little bit stuff-less and unconvincing though the dataset is large enough. As a matter of fact, the majority of data which are being considered as outliers are the ones that having SalesPrice of 1. People might consider these values as missing. However, they are not, and indeed they are the public housings that are built by the government or related non-profit organizations for the homeless and the low-income families. I would suggest including these data if the project would be a classification problem. In addition to IQR, using Cook Distance test and Hypothesis testing such as Grubbs' test and Chi-square test would be conducive. Moreover, my models are lack of efficiency due to the fact that there are too many features after doing preprocessing. Someone may claim that I should do feature selections, but from my point of view, I don't agree with it. I think the model would be less persuasive if we simply use SelectKBest or SelectPercentile to detect what features should be included. Indeed, one could adjust their models after roughly modeling the data. Doing some dimensionality-reductions would be help improve the efficiency of the modeling processes. Besides these two purely technical issues, adding some other features such as the average price of

the housing in the block, the number of schools in this town code district, the distance to the closest railway station, the number of grocery stores in this district would contribute to a more accurate prediction. In short, though I am satisfied with this ridge model, I should keep in mind that no model is perfect. Also, I should always think about how to improve the models.

VI. References

[1] Maude Toussaint-Comeau. https://www.chicagofed.org/publications/profitwise-news-and-views/2018/determinants-of-housing-values-and-variations-in-home-prices-across-neighborhoods-in-cook-county

[2] Link to the Github: https://github.com/bwang98/data1030_project