Abstract

House pricing is usually determined by a lot of different factors such as location, neighborhood, property size and etc. In this paper, we compete in a Kaggle competition: “House Prices: Advanced Regression Techniques”, and we aimed at utilizing machine learning technique to outperform traditional linear regression model and ranking top 15% on the leaderboard of the Kaggle competition. The dataset that we are using is the “Ames Housing Dataset” which are split into training and testing dataset, and each of which contains 80 features and 1459 observations. Our predictive model is aiming at predicting the sales price which is the independent variables of our model. We used Linear regression as our baseline model, and we used Lasso, ElasticNet and LightGBT to outperform the baseline model.

Business Understanding

Value (Insert as statement)

When looked from a macroeconomic perspective, there are many aspects that significantly affect the behavior of this market, such as demographics, interest rates, government regulation and global economic health. However, looking at housing market at a global scale is too simplistic, and there are a lot of factor that is affecting the market on a local scale such as the neighborhood, property size, local crime rate and even the house orientation.

Apparently, the most important factor that affects real estate asset is the size and location, but some other variables can have an impact on price as well. It is very like a newly renovated property can raise its price by a certain percent, which is higher than the average prices of its neighborhood. Besides, the demand of housing has seasonality, the housing price can vary from time to time because the demand of houses will change due to the economic health and government’s regulation policy.

Our model focusing on predicting the housing price based on the objective factors such as property size, garage size, street condition and other factors. The dataset that we used is the Ames, Iowa housing dataset. It contains 2918 observations and 80 features including the sales price. The whole dataset was divided by half into a training dataset and testing dataset.

Basis (Insert as statement)

Our baseline model is internal. The linear model is a natural choice of baseline model, and we want to outperform the baseline with the advanced regression model. In addition to the test set accuracy, the performance metric that we use to measure the models that we built measured by R^2 and rmse score both of which measure the good-of-fit on the model while R^2 is the relative measure of result on how observation is following the regression line and rmse.

In addition to the internal metric, we competed in the Kaggle competition “House Prices: Advanced Regression Techniques”, and the rank of our model on the leaderboard is our external performance metric. We were aiming at getting to top 15% on the leaderboard.

Value-Add (Insert as performance metrics

Our value add is the spread between the fit of our linear model forecast and the fit of our machine learning model housing prices forecasting. We want to build a model that can pick out the features that have the most impact on the house pricing in the local districts and how our model can outperform the traditional linear models and other regression model with boosting technique.

Data Understanding

Data description

The ‘Ames, Iowa Housing’ data set was created by Dean De Cock for data science education, we found this data set in a featured Kaggle competition, we downloaded both the original data set and the Kaggle version data set for the analysis and the competition. The initial data set contained 113 variables demonstrating 3970 property sales that of personal residential property in Ames, Iowa in 4 years (from 2006 to 2010). Since ordinary house buyers would not like to know about information of some less significant attributes such as the height of basement ceiling, the author removed some irrelevant variables and the dimensionality of the data reduced to 81 variables and 2930 observations. Those 80 explanatory variables explain the quality and quantity of almost every physical characteristics of individual residential homes in Ames(e.g. When was it built? How big is the lot? How many square feet of living space is in the dwelling? Is the basement finished? How many bathrooms are there?).

Specifically, there are 20 continuous variables describing area measurements(e.g. Area measurements on the basement, main living area, and even porches are broken down into individual categories based on quality and type).

Next, 14 discrete variables shown in our data focusing on explaining the number of kitchens, bedrooms, and bathrooms (full and half) located in the basement and above grade (ground) living areas of the home. It is worth mentioning that some variable seems like a numeric value such as garage capacity and construction dates were also recorded as discrete variables.

Also, 23 nominal and 23 ordinal categorical variables are related to environmental conditions and ratings of various items within the house. The nominal variables typically identify various types of dwellings, garages, materials, and environmental conditions while the ordinal variables typically rate various items within the property.

Those variables range from 2 to 28 different categories with the smallest one being STREET (gravel or paved) and the largest one is called NEIGHBORHOOD (areas within the Ames city limits). There is an interesting variable called PID in our variables. PID is the Parcel Identification Number related to each property in Ames Assessor’s system. This ID is used only if we want to check the records of some particular observations directly from Assessor’s office or the Beacon website. Hence, we decided to drop this feature at the beginning because we do not expect this variable would contribute to improving our predicting accuracy.

For our purposes, a data set that could be easily understood by users with different levels of background was desirable. As a result, we began our project by removing any variables that required special knowledge or previous calculations for their use. Most of these deleted variables were related to weighting and adjustment factors used in the city’s current modeling system.

Data exploration

Our initial idea that the data was not important came from our misunderstanding of what it was. In the stage of data exploration, we have been able to further deepen in the data, allowing us to have a better sense of it, and thus, improving our feature engineering. some techniques were used to address questions in our data.

● Data completeness: Firstly, we checked the distribution of critical attributes by generating distribution plots. According to those graphs (in our jupyter notebook), there were plenty of missing values among our explanatory variables which we had to input them with some values or remove them. Additionally, some categorical variables were displayed as numeric values which need to be fixed it later as well.

● Analyzing target variable: Next, we tried to focus on our target variable ‘SalePrice’ by generating histograms. From those graphs, we could see that the minimum value is not smaller than 0 which means there is no abnormal observation, and those plots also demonstrated a phenomenon that the data deviates from the center and there is positive skewness since there is more housing price with a large value than a small value.

● Relationships: Another analyze we did is to study the relationship between dependent variable and independent variables. By generating all scatter plots between 'SalePrice' and correlated variables separately and checking the correlation matrix in heatmap style. Consequently, we obtained a quick view of relationships among variable. Although there were some outliers, most numeric values tended to have a positive linear relationship with the target variable. For categorical variables, 'SalePrice' preferred to be larger in higher variable level. What surprised us most was that there was not a strong tendency in the relationship between 'SalePrice' and 'YearBuilt', the money spent on a new building was only a little bit more than an old relic. Moreover, the correlation matrix suggested us that some variables were correlated strongly. For example, 'GarageCars' and 'GarageArea' were some of the most strongly correlated variables with a correlation coefficient higher than 0.6.

Data preparation

To raise the data quality to the level required by our analysis techniques, we decided to use the following data cleaning methods to improve our data.

● Dealing with the normality of the target variable: A simple log transformation can be useful here. However, a lot of observations in our data with zero value could lead to a no significant effect on log transformation. Hence, we decided to do a log transformation to all the non-zero observations by disregarding zeros. After transformation, the QQ-plot demonstrated that almost all data points followed a diagonal line and the distribution of data shifted to the center, which means that the positive skew corrected and 'SalePrice' became pretty much normal.

● Handling missing value: There were 27 variables containing missing values. We have to be very careful to avoid being biased and losing useful information when we are inputting or deleting missing variables. Fortunately, most variables with more than 15 percent missing rate, for example, 'PoolQC' did not seem to be essential. Therefore, we just simply deleted those variables, and we imputed medians for those remaining important variables such as 'Electrical'. We have removed variables, but also have imputed data in order to achieve better accuracy levels.

● Getting rid of outliers: We constructed a function to detect and drop those outliers. In detail, we first separated the numeric variables from the original data, then we considered any data point that fell outside of either 1.5 times the IQR below the first quartile or 1.5 times the IQR above the third quartile to be outliers and remove them.

● Switching data type: Categorical data are variables that contain label values rather than numeric values, many machine learning algorithms cannot operate on label data directly. They require all input variables and output variables to be numeric. It is necessary for us to convert those 78 character variables into categorical variables to reduce additional complexities of our prediction. By simply using one-hot-encoding functions and allowing the model to assume a natural ordering between categories may result in poor performance or unexpected results , we successfully switched the data type and created dummy variables for categorical variables. As a result,

Model Understanding

Model Goal

The goal of our model is not to forecast but to infer from the housing data, since our analysis is not time based.

Model Selection

We selected three models, Ridge Regression, LASSO, and Elastic Net as baselines and used Gradient Boosting for our value-add. In order to see whether the Gradient Boosting value-adds to our baselines, we calculated R-Squared and MSLE for each model and compared the values. The results of the comparison will be dealt in the Model Performance section.

Model Description Part 1.

The three baselines that we chose were Ridge Regression, LASSO, and Elastic Net.

1. Ridge Regression

We chose the Ridge Regression model as one of our baselines, since it is robust to overfitting and multicollinearity problem compared to a Linear Regression model. Multicollinearity problem occurs when there is a near-linear relationship between two certain features. When features are correlated, it becomes difficult to change one variable without changing the other, thus distorting the fitted estimates.

The Ridge-Regression model solves the problems above by including a penalty to the original Linear Regression. The penalty uses L2 regularization and is calculated as sum of squared coefficients multiplied by a penalty term, usually expressed as l. It is important to choose the penalty term wisely, since it affects the overall quality of the model. Most of the times, the penalty term is chosen heuristically in order to pick the one that performs well with a data.

As the equation suggests, the penalty term penalizes or shrinks the coefficients with large values. Thus, by minimizing large-valued coefficients, minor changes would be made to a single data point, hence, reducing the model’s variability and loosening the dependency relationships among some features.

2. LASSO

LASSO stands for Least Absolute Shrinkage and Selection Operator and is another model which solves the overfitting and the multicollinearity problems. LASSO is very similar to the Ridge Regression except it uses L1 Regularization as penalty. L1 Regularization is calculated as sum of absolute coefficients multiplied by a penalty term, l.

One big difference between the Ridge Regression and LASSO is that the L1 Regularization allows the less important features to minimize towards zero, hence, penalizing such features completely and removing them from the model.

3. Elastic Net

Elastic Net encourages group effect that the Ridge Regression and the LASSO can’t handle and solves a sparsity problem that can possibly arise from the LASSO. By introducing an Elastic Net Regularization which incorporates L1 regularization and L2 regularization at the same time, it tries to take advantages of having L1 regularization and L2 regularization at the same time.

One top of that, the sparsity problem from LASSO, occurring when it minimizes overwhelming numbers of variables to zero, can be dealt with by including the L2 regularization which will make them minimize to near zero.

Model Description Part 2.

Gradient Boosting

Gradient Boosting is one of the boosting methods that uses a gradient descent to minimize errors and find a local optimal point. As with any other boosting method, the advantages of Gradient Boosting are it can minimize the bias-variance trade-off and increase a prediction rate. By applying gradient descent to the boosting, it allows the model to find the optimal point more stable. On top of this, it can handle missing data. However, disadvantages of the Gradient Boosting are it is computationally expensive, requiring time and memory, and it sensitive to hyperparameters.

Hyperparameters

Maximum Depth

3

Maximum Features

70

Alpha

0.85

Three of the important hyperparameters we specified in the model are, first, maximum depth of a tree. This parameter is used to control over-fitting as higher depth will allow model to learn relations very specific to a particular sample.

The second parameter is the number of maximum features to consider while searching for a best split. These will be randomly selected. As a thumb-rule, square root of the total number of features works great but we should check up to 30-40% of the total number of features. Higher values can lead to over-fitting but depends on case to case. The alpha is the alpha-quantile of the loss function.

Descriptive Statistics

We plotted distribution for each continuous variable in histograms to visualize and understand our features before plugging them into the selected models. The distribution shape varies among the features, where we have normal, left-skewed, and right-skewed data. Among them, 9 are non-skewed, 4 are left-skewed, and 23 are right-skewed. To specify this by feature names, GarageCars, GarageArea, OverallQual, OverallCond, FullBath, TotRmsAbvGrd, BedroomAbvGr, MoSold, YrSold are non-skewed, WoodDeckSF, YearBuilt, YearRemodAdd, GarageYrBlt are left-skewed, and MSSubClass, 1stFirSF, LotFrontage, 2ndFirSF, LotArea, LowQualFinSF, GrLiveArea, BsmtFullBath, OpenPorchSF, BsmtHalfBath, EnclosedPorch, 3SsnPorch, MasVnrArea, HalfBath, ScreenPorch, BmstFinSF1, PoolArea, BmstFinSF2, KitchAbvGr, MiscVal, BsmtUnfSF, TotalBsmtSF, Fireplaces are right-skewed.

Apart from the distributions, we also studied relationship between each categorical independent variable and dependent variable, housing prices, by calculating the R-Squared.

I can’t run the code can u replace these so they are legible? Thanks

Among them, the three categorical variables with highest R-Squared values are, in terms of their feature name, OverallQual, GarageCars, and FullBath.

Variable Relationships

Our prior belief is that the variables which affect the data most are the ones with the highest R-squared across the models. As variables with their highest R-squared values indicate that those are the most important predictor variables in regression models, the top three features, OvearllQual, GarageCars, and FullBath would have the highest coefficients in terms of their absolute values.

Model Performance

It’s reasonable to perform the multiple linear regression at the very beginning. We’ve got 82 variables with housing price as dependent variable and other variables describing the characteristics, qualities and matching facilities of house. All the independent variables may have some impact on the final housing price. The problem of linear regression is the number of independent variables. As I mention above, we have 80 independent variables which indicates that our data is high-dimensional. When fitting linear regression model with high-dimensional data, we may overfit the model and it becomes more difficult to make a relatively accurate prediction, which is our final goal of this project. Thus, we consider Lasso regression and Ridge regression which are more suitable for our data. Lasso and Ridge both reduce the dimension of data by adding “penalty” to the model. The difference between Lasso and Ridge is that Lasso can sift variables and select variables which are more significant, while Ridge simply make compression on the coefficients of variables and all the variables will appear in the final model, that is, no variable selection.

Two parameters are considered when evaluating models, which are R square and MSLE. R square is the ratio of sum of squares of the regression and total sum of squares. The range of R square is [0,1], the closer to 1, the better the model fits. MSLE is mean squared logistic error, the closer to 0, the better the model fits. The results of Multiple Linear Regression, Ridge Regression and Lasso Regression show no much difference, but lasso regression do perform better than other two models which gives 0.61 in R square and 0.221 in MSLE.

The result of Lasso Regression is not ideal enough. Moreover, we can find that many independent variables are quite related to each other which gives us another issue to deal with. Elastic Net Model is then considered. One of the advantage of Elastic Net Model compared to Lasso Regression is that when facing two related characters, Lasso will choose one of them randomly while Elastic Net Model will choose both of them. The result of Elastic Net Model is 0.585 in R square and 0.216 in MSLE. We get a decrease in MSLE as well as a decrease in R square, which indicates that the variance of Elastic Net Model becomes less while less data is explained by this model.

Then, we consider Gradient Boosting Model. Gradient Boosting Model is an ensemble learning process. The model optimizes classification result through iteration. A new “classifier” is put into use after one iteration in order to make up the shortcomings of previous “classifier”. At last, we can get an optimal model by lowering loss function. As we fit Gradient Boosting Model with our data, we get much better evaluating parameters which give R square 0.83 and MSLE 0.134. Comparing this result with that of Multiple Linear Regression which is our baseline model, we optimize the model by increasing R square from 0.5 to 0.83 and decreasing MSLE from 0.235 to 0.134. Although we’ve already got better parameters when fitting Gradient Boosting Model, the disadvantage of this model cannot be ignored. The computation process is costly which may be a problem when we have large volume of data.