Final Project Report

Zining Wang, Wenxuan Wang, Wenda Zheng

Motivation and contributions

Our project is House Price Prediction, which is from Kaggle Competition. We are interested in this topic because houses are indispensable things in our lives and house price is therefore a heated topic in our society. People all have their dream houses in their minds, either fulfill some certain criteria. For example, some people prefer houses containing large bedrooms, some people likes car so they want to have large garages, and some people choose large and beautiful gardens since they are in favor of gardening. The house price will be adjusted according to those different kinds of conditions.

Therefore, we would like to exploit some straightforward methods that can predict the house price, i.e. given all the important information as an input, we can derive the output as the predicted house price. The advantages of predicting house price are listed below:

1) The predicted house price should be close to the actual value of the house.

2) You can predict your the actual value of your house without field investigation. Therefore, it is really convenient.

3) Almost all the houses on sale have higher sale price than their actual values. Once you get your predicted house price, it is much easier for you to bargain with sales to lower their sale price.

4) You can compare different houses and their prices simultaneously.

In short, our goal is to predict the house price according to different house features.

In our project, the data come from kaggle, which include a training dataset and a testing dataset. In the training dataset, there are 79 attribute values including numerical attributes and categorical attributes. More specifically, those 79 house attributes contain structural attributes (e.g. the area of first floor), location attributes (e.g. whether the house is adjacent to railroads), and some overall quality ratings.

In our project, Zining Wang mainly works on data preprocessing and testing regularized linear regression. Wenxuan Wang mainly works on testing XGBoost on the dataset. Wenda

Zheng mainly did data preprocessing with Zining Wang, prepare for slides for presentation and test the code.

Related Work and methods

The existing work are mainly a combination of different models. Some of the examples are Elastic Net, Gradient Boosting, and Regressions.

One of the most common models is the regularized linear regression model. This is a linear regression that introduce additional information to avoid overfitting and ill-conditioned problems. More specifically, the regularized linear regression model adds a loss function in its expression as a constraint for the increase in feature numbers. In our dataset, there are 79 attributes in total, including both numeric ones and categorical ones. Since we want to predict accurate results from the house price dataset, a regression model is ideal and will work better than any other data mining and machine learning methods.

Also, gradient boosting (XGBoost), a very popular and widely used method on kaggle, is also a practical method for this dataset. We will also use this in our project because it is a very commonly used and appears among the top solutions of many data competitions. XGBoost is a machine learning algorithm and could be used for regression problems. In more details, it uses an ensemble of relatively less powerful models to generate a more precise model. The less powerful models that are used are mainly decision trees. The gradient boosting method generates the model stage by stage and can eventually provide a more accurate prediction.

Another available method is the Elastic Net method. It is also a regularized regression model that linearly combines the lasso and ridge methods. Compared with the common regularization method, this method is more comprehensive and may give a better prediction.

There are also some other methods, such as Neural Network or clustering. These are less feasible in this dataset so will not work as well as the previous methods do. And usually the combination of the mentioned methods could provide a better estimation.

Approach and methodology

First, we preprocessed the data using different techniques. Some features such as "MSSubClass" (type of dwelling involved in the sale) are in numerical format, but in fact, they are categorical variables, so we converted them into string format. The same situation happened for "MoSold" and "YrSold" (month sold and year sold), so we transformed them into categorical variables.We also check the numbers of missing values for all 79 attributes, and impute those missing values using different methods depending on each feature. For some categorical features such as "MSZoning" (identifies the general zoning classification of the sale), we imputed missing values with mode of the training data, since the mode (most frequency category) dominated the training set. For numerical attributes such as "BsmtUnfSF" (unfinished square feet of basement area), we imputed missing values with mean. In addition, for some categorical variables, "NA" means no such attribute. For example, for the feature "Fence" (fence quality), "NA" indicates "No fence", rather than indicating the value is missing. So to prevent our program from treating it as missing values, we replaced it with a new category, "NoFence". At last, we also dropped several attributes with too many missing values, or do not correlated with the sales price, such as "ID".

We created dummy variables for all of our categorical variables, and used min-max normalization to normalize all our numerical attributes to the new range from 0 to 1. In this way, we can apply our regression models better.

Also, to study the correlation between sales price with each individual feature, we made pair-wised plots for sales price with each feature and tried to observe the tendency in the plots. We plotted the top 15 most related plot in our results and were able to observe some strongly related features including OverallQual, GrLivArea, GarageCars… This extra step allowed us to pick the most important features and also helped reduce the unrelated and less related features. To increase the readability, we used the correlation values to plot a heatmap which even better showed the pairwise relationships. The heatmap became a very straightforward way to select the important features.

The first model we tried is the regularized linear regression model. We used multivariable linear regression to predict the house price based on the package of attributes, and used Lasso regularization model for feature selection. Our independent variables Xi are those preprocessed features, including dummy variables. Our dependent variable Y is the house price. Lasso model aims to find the optimal weight for each independent variable that minimize $\frac{1}{2n}||Xw - y||_2^2 + \alpha||w||_1$, where α is the regularization coefficient that controls the complexity of the model. We chose the best α using grid search and a 5-fold cross validation on the training set.

Then, we added the XGBoost approach into our experiment and used both the regularized linear regression model and XGBoost to get a better prediction. The XGBoost method made its prediction from a set of relatively weak prediction models and would give a better estimation based on the weaker models. Also, gradient boosting is widely used and often give quite precise predictions. Even if we only use XGBoost, we still got an ideal prediction according to our submission on kaggle.

According to our observation, regularized linear regression model often got relatively small prediction values while XGBoost often give larger values. Therefore, we also tried to ensemble two models. By doing a linear combination of these two methods, we were able to average out the errors and get better results.

Evaluation and results

To evaluate the results, we made a partition at the very beginning, predicted the test values using our models and then calculated the root-mean-square error(RMSE) by comparing the predicted values with the exact value.

Since we did not have the accurate results for prediction of kaggle test set, we could not perform analysis based on the test dataset. Instead, we made a partition of the original dataset and used 75% for training and another 25% for testing. Since the testing dataset had never been used for training, the model were not biased.

After constructing our predicting models, we used it for predicting house prices of the test dataset and calculated the RMSE values for both regularized linear regression model and the XGBoost model. The RMSE for regularized linear regression model was 696.07 and the RMSE for XGBoost was 2884.46 while the values range was from 100,000 up to 700,000.

After this, we plotted the correlation between predicted values and real values on a single plot and observed that all these values are close enough. On plot plot, it was nearly a perfect fit due to the scale of the axis. But the plot showed that our model were very accurate and could successfully predict the house price and obtain a high accuracy.

Conclusion and future work

a) During working on this project, we learned a lot.

The most important thing is that we learned what is actually data preprocessing and its importance. We think data preprocessing plays a leading role in our project. Without data preprocessing, our results will be pale. There are 79 attributes in our datasets. If we do not preprocess the raw data well, the results would be extremely complex. Actually, some of the attributes do not make any influence on the house price. Moreover, the datasets provided by Kaggle is somewhat incomplete: there are some blanks and NAs. The preprocessing techniques have already discussed above. We can see the procedures are very complicated. However, once we finished our data preprocess, the rest parts are relatively easy to deal with.

Secondly, we learned how to predict the house price using different models and compare which one is better. There are two models in our project, the first one is Lasso linear regression model and the second one is XGBoost model. In the Lasso regression model, we learned what is regularization. Since there are so many attributes (view as dependent variables), we need a regularizer to control the complexity of our model. In short, regularizer control the smoothness of our regression line. Furthermore, we also learned the application of XGBoost. XGBoost is somewhat similar to decision trees that we learned in class as it is also based on a boosting tree model. The most important thing we learned from two models is that we learn how to combine two models to get a more accurate prediction. Our combination is linear, i.e. a*lasso + (1-a)*XGBoost. We tried various values of parameters and finally find one that optimizes our linear combination of two models.

b) future work.

Future works need to be done to find smarter ways to preprocess some features. For example, some location variables such as “Neighborhood” (Physical locations within Ames city limits), can be better represented using other ways to dummy code, which incorporate specific location characteristics within that city. Some areas may be closer to shopping mall or restaurants, so the house price may be higher. In addition, some places may be noisy, so the corresponding house price should be adjusted below average.

What is more, a multi-task model can be used to divide the data into different groups based on different features, and build an individual model for each group, and combine the local effect with the global model.