Use Machine Learning Method Predict House Price

Feng Yongxiang yfeng24@buffalo.edu

Ashutosh Anand aanand7@buffalo.edu

Abstract-In general, predict house price is a difficult work, because the price is influenced by many factors. People who want to predict it should first collect many related data like type of house and neighborhood. Next use some methods to calculate the weight of every factor. Last to create the math model to predict the house price. But now we can collect massive data and use some models to let computer deals with data by itself. Usually, we can use different type of machine learning methods to get the ideal result.

Index Term-machine learning, random-forest tree.

1. Introduction

In the recent years, much data appeared in the world. We are busy to deal with massive seeming unrelated data and easily to drop in. Fortunately, many AI scientists create some methods for us to deal with these massive data. We can directly use these models to get connections with data, like price and environment. And these models can be calculated by computer. It can enhance work efficiency.

In this passage, we use some technic methods to solve the problem about prediction of house price. The process of it is clear, get data from website, preprocess data, select the model, train the model and get the result. We will use different methods to process the data to see which one is better to predict the result.

2. Database

There are many websites provide data. We choose Kaggle to get the data because it provides

relatively complete data frame and many methods to use it like download and using API. This dataset includes 80 features like street and house style and 1 label SalePrice. We should use these features to predict the SalePrice. These features also divide into number and text.

80 features: Id, MSSubClass, MSZoning, LotFrontage, LotArea, Street, Alley, LotShape, LandContour, Utilities, LotConfig, LandSlope, Neighborhood, Condition1, Condition2, BldgType, HouseStyle, OverallQual, OverallCond, YearBuilt, YearRemodAdd, RoofStyle, RoofMatl, Exterior1st, Exterior2nd, MasVnrType, MasVnrArea, ExterQual, ExterCond, Foundation, BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinSF1, BsmtFinType2, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, Heating, HeatingQC, CentralAir, Electrical, 1stFlrSF, 2ndFlrSF, LowQualFinSF, GrLivArea, BsmtFullBath, BsmtHalfBath, FullBath, HalfBath, BedroomAbvGr, KitchenAbvGr, KitchenQual, TotRmsAbvGrd, Functional, Fireplaces, FireplaceQu, GarageType, GarageYrBlt, GarageFinish, GarageCars, GarageArea, GarageQual, GarageCond, PavedDrive, WoodDeckSF, OpenPorchSF, EnclosedPorch, 3SsnPorch, ScreenPorch, PoolArea, PoolQC, Fence, MiscFeature, MiscVal, MoSold, YrSold, SaleType, SaleCondition.

1 label: SalePrice.

3. Method

Firstly, we load the data and show some information about it. It has the number data and text data and some of it misses some data. Then we visualize the data, get some information about the distribution of it. Some data are crowd and some data are discrete. Next we use the method to calculate the correlation of the features and the SalePrice. Some data have less correlation may drop off in the later. Next we preprocess the number data, we first get some miss data's medium number, and use the method in the sklearn to fill miss data with the medium number in the column. We preprocess the text data, because the

model only receives number data, so we should transfer text data to number data. After transform the data, we also use medium data to fill in the miss data.

Because there are many features, so we decide to choose ten explanatory features as a compare list. We will train this list and list contains all features and get results separately. These results may show different or same. And we will analysis which method is better to use which list. Because we image some methods are more likely to get perfect result in less features list but some methods will get result in all features list.

We select several models to train the data. These methods are Linear Regression, decision tree, random forest tree, svm, xgboost and lightboost.

3.1. Linear Regression

Linear regression is a method to model the relationship between a response and some variables. We usually use some features to train in the model and get a result.

We use all features and ten outstanding features to train the model, and graph them. As shown in the next two graphics, if predict one and real one are correspond perfect, they will have same number. And the points will more like a linear. So we see if we use all features to train the model, the result is not perfect, if we use these ten features to train the model, the result is better.

We also use line chart to show compatibility between the real data and predict data, it will show the result more Intuitive.

Figure 3.1.1 Linear Regression all features

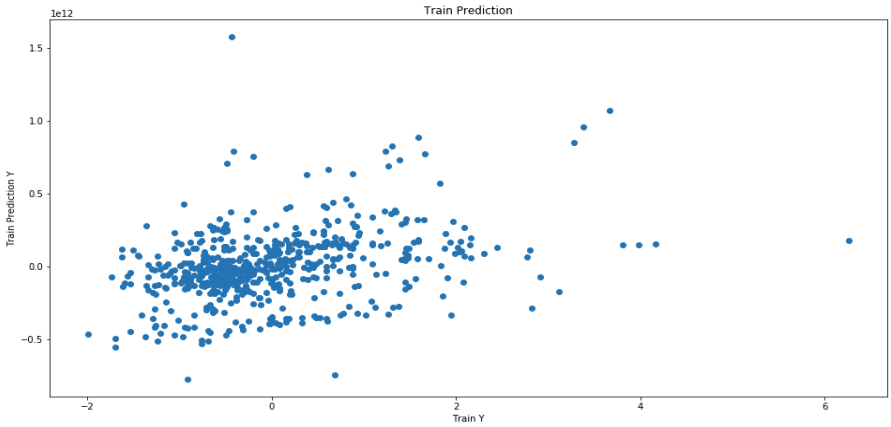


Figure 3.1.2 Linear Regression outstanding features

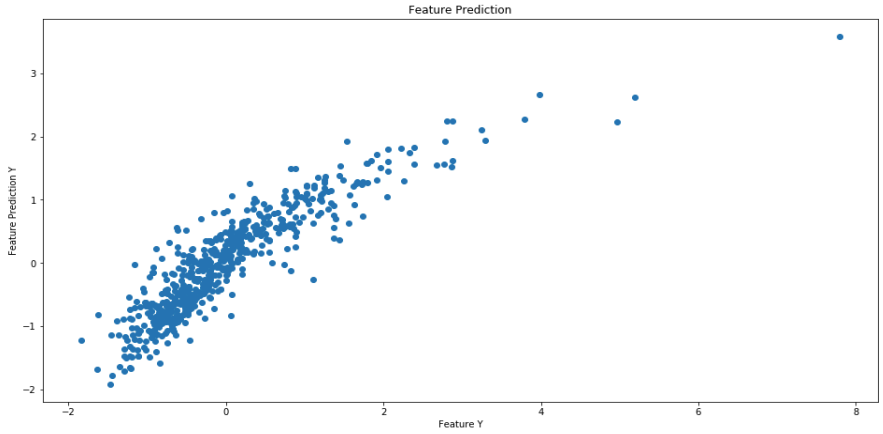
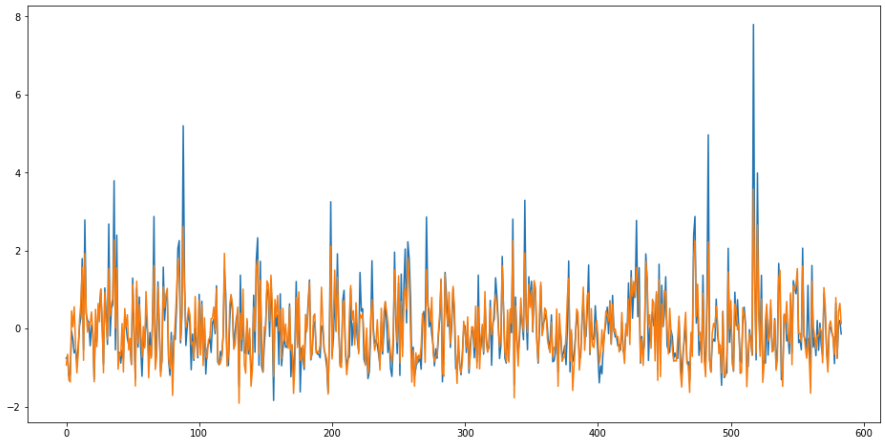


Figure 3.1.3 Linear Regression all features

Figure 3.1.4 Linear Regression outstanding features



3.2. Decision Tree and Random Forest

Decision tree is a method to use a tree-like model to predict every leaf's result and consequence and combine them as the final result.

Random forest is an ensemble method. It combines many decision trees to get their results to get a mean score of them. Use this mean score as a final result. It's better than decision tree. Because decision may occur some accidence like overfitting and underfitting. If we use mean score of them, we can avoid this situation.

We also use two list features to train them to get results. It shows little difference between two methods, so I will show some graphic to identify them. And in the conclusion, I will compare different model's difference.

Figure 3.2.1 Decision Tree all features

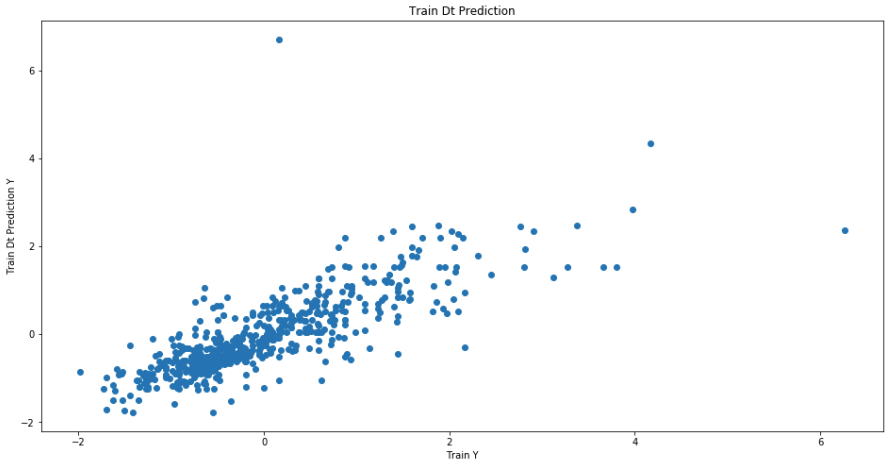


Figure 3.2.2 Decision Tree outstanding features

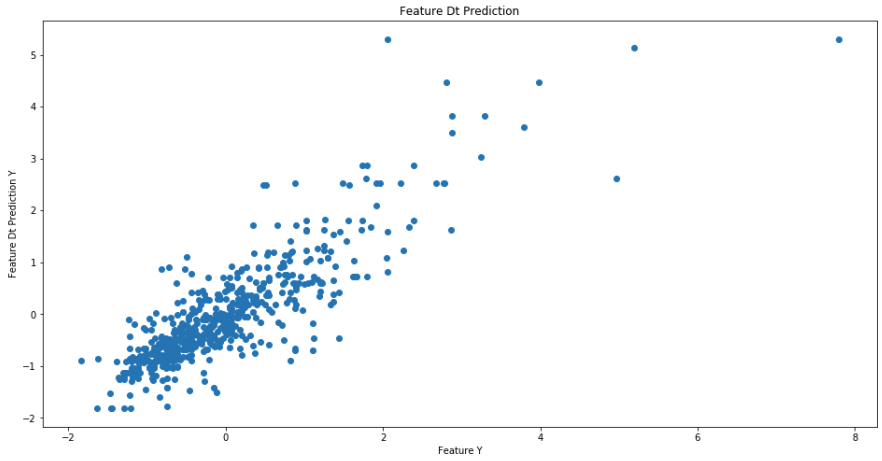


Figure 3.2.3 Random Forest all features

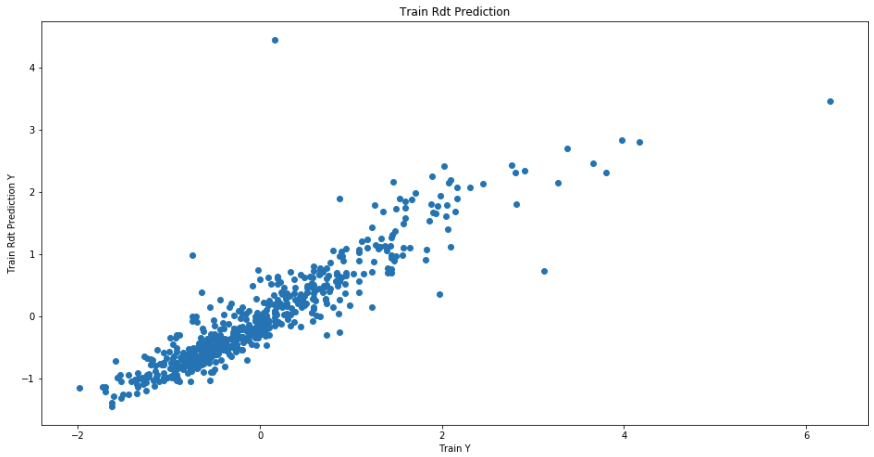
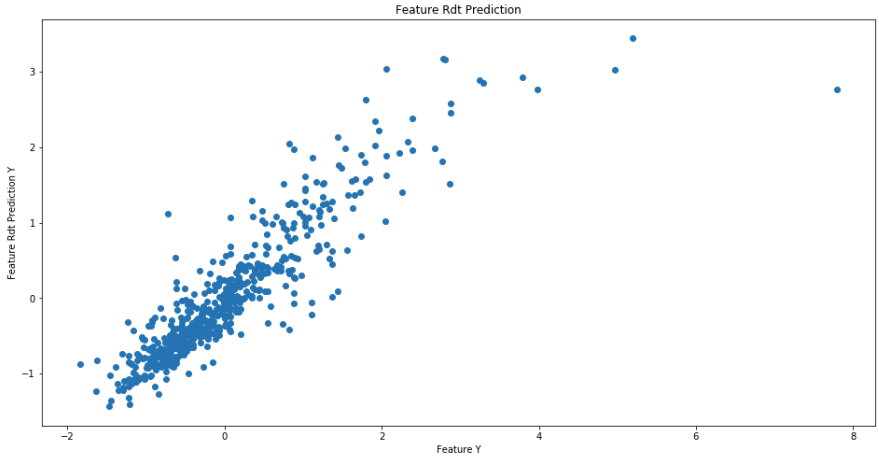


Figure 3.2.4 Random Forest outstanding features



3.3. SVM

SVM can be used by classification and regression. If we use this model, it will divide the all data into two or more separate categories. It tries to find a line to divide these data as far as possible.

Figure 3.3.1 SVM all features

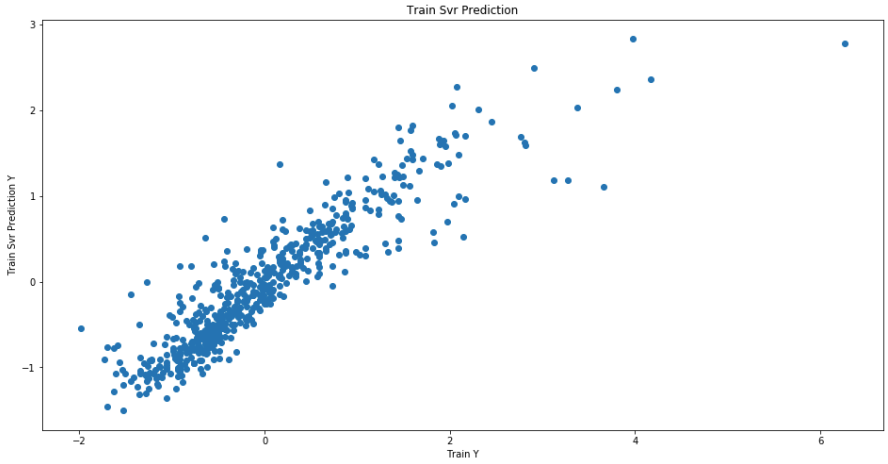
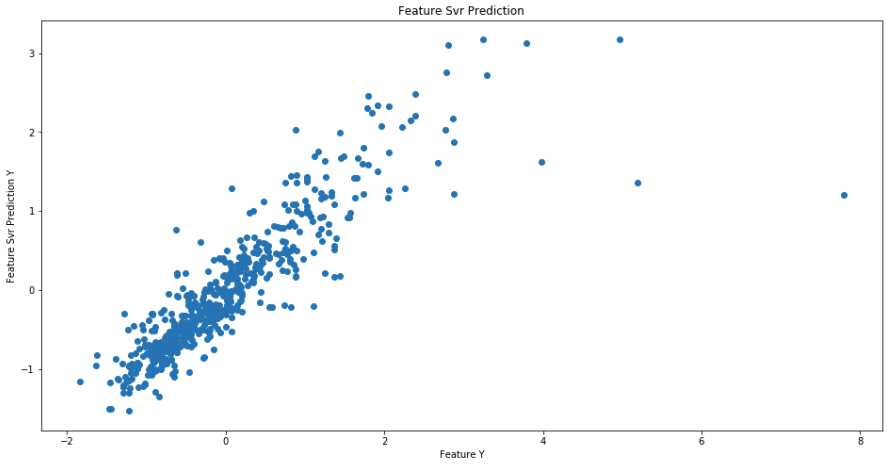


Figure 3.3.2 SVM outstanding features



3.4. Xgboost and lightboost

These two methods are all depend on decision tree algorithm. And they are also improvements of gradient boosting. Xgboost creates decision tree in every step to get a result to let next decision tree to learn. Lightboost is same as xgboost in some degree. But xgbbost uses level-wise tree growth, it learns every level's result as the final result. Lightboost uses leaf-wise tree growth, it will choose excellent features to learn. So lightboost will run faster than xgboost, but sometimes it will occur overfitting. We should use them wisely.

Figure 3.4.1 xgboost all features

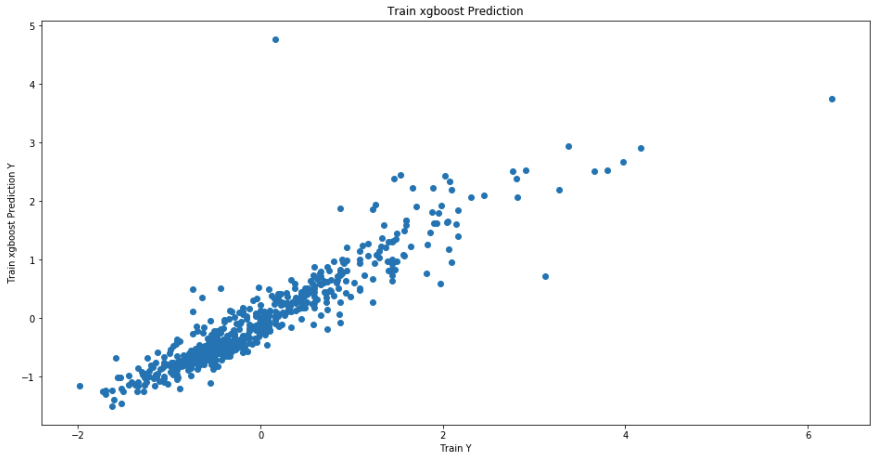


Figure 3.4.2 xgboost outstanding features

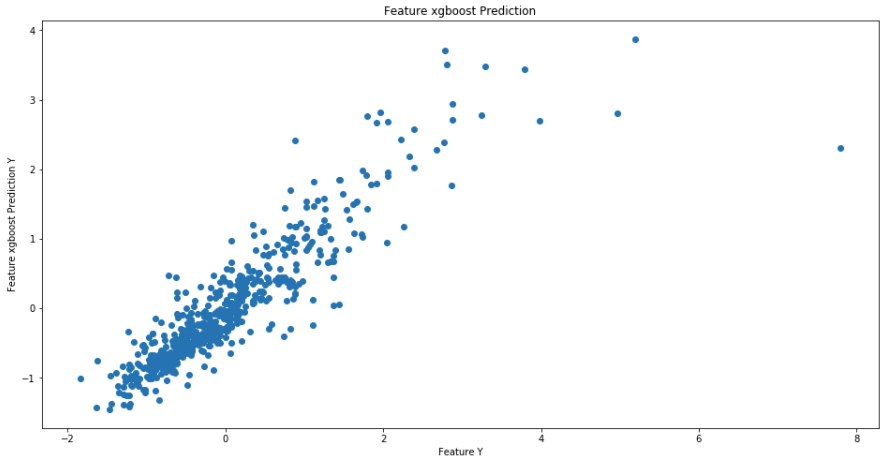


Figure 3.4.3 lightboost all features

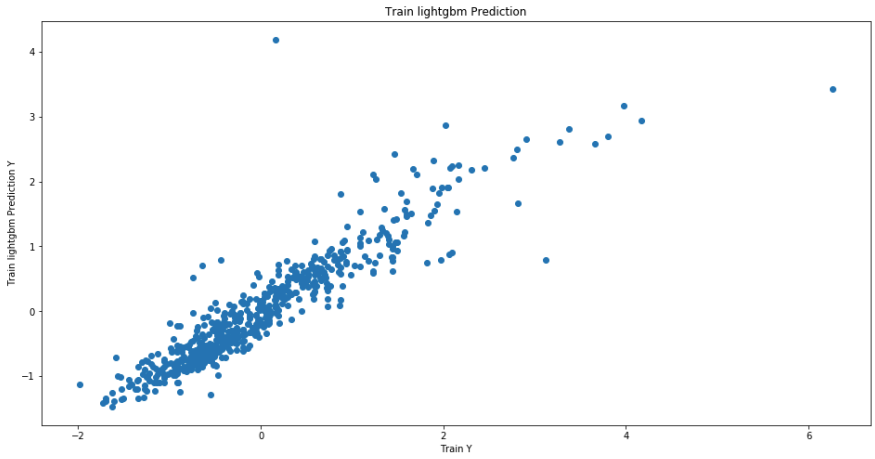
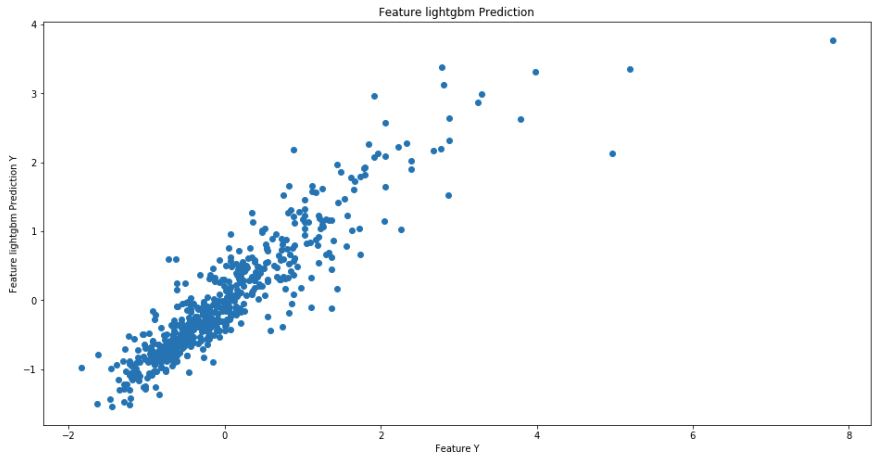


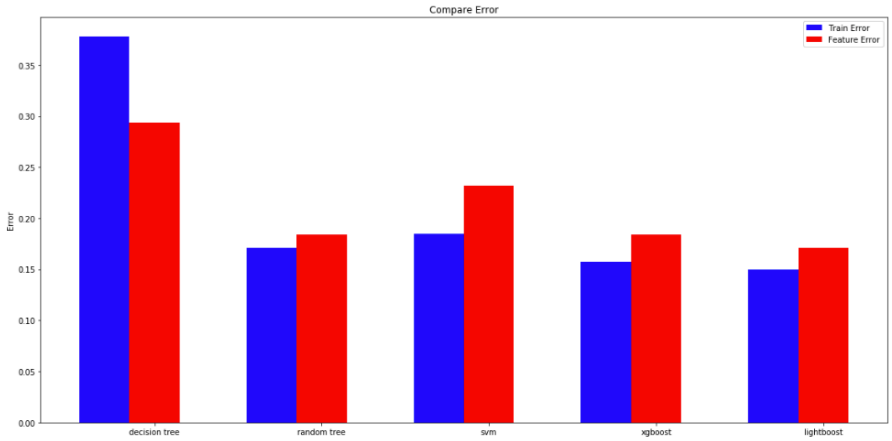
Figure 3.4.4 lightboost outstanding features



4. Conclusion

We now use every model to train data and get results. From our observed, linear regression is not perform well. So we can ignore it. And we compare other methods' results, which one is better.

Figure 4.1. Compare Result



We can see X-axis is different method, and Y-axis is error. The last four methods have less error than the first one. And we know boost methods has better results. Because the lightboost method run less time and has little error. So we can conclude lightboost method is the best one among these machine learning methods.

Youtube: https://www.youtube.com/watch?v=tr956EOTO0M&feature=youtu.be