Technical Report for Kaggle Competition

Muyang Niu

Table of Contents

1. Introduction.....3

2. Process.....3

2.1 Data exploration & visualization.....3

2.2 Data Cleaning..... 5

2.3 Feature Engineering.....6

2.4 Modeling.....6

3. Conclusion..... 8

References.....9

1.Introduction

The Kaggle Contest is a data science contest which needs various abilities such as Python/R, machine learning, deep learning etc. In this project, I focused on the prediction of each residential home in Ames and Lowa. With 79 explanatory variables which were used describing every aspect of residential homes, we took measure to work it out and got our project improved during the process.

2.Process

The process can be briefly separated into four parts: Data exploration & visualization, Data cleaning, Feature Engineering and Modeling. The modeling part is the most complicated stage in this project which is also directly relevant to the accuracy of our results.

2.1 Data exploration & visualization

Data science as a subject has a longer history than machine learning, however, it still develops so rapidly that human beings have to deal with data problems more frequently than before. According to Lgual and Segui (2017), the usage of Data science are concluded as: Probling reality, Pattern discovery, Predicting future events and Understanding people and the world. Here in this project, it mainly worked for predicting.

The 1st step was data exploration. I used Python for core coding language because of its maturity and its newbies-oriented feature (Lgual & Segui, 2017). Here the core stages contains import libraries, read data and some pre-functions for dataframe. Numpy, Pandas and Matplotlib were mainly used at this stage. Besides, the SQL also has the similar function for processing data.

As for data visualization, I mainly used Python and Tableau. Seaborn was installed in Python for making plots. Commonly, making plots by Python is effective for personal use while Tableau is more suitable for continuous variables. Overall, they both have their pros and cons.

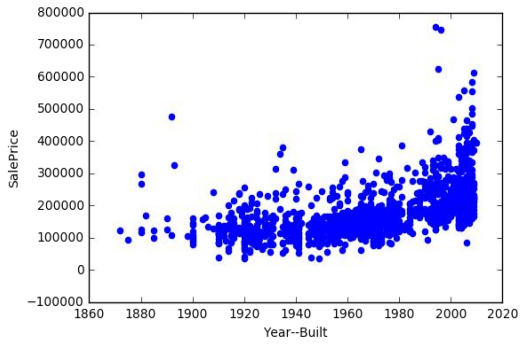

Figure 2.1 Scatter-plot by Python

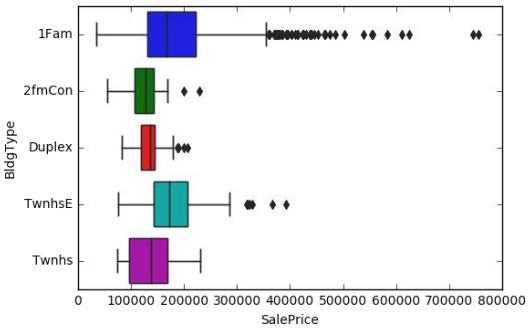

Figure 2.2 Box-plot by Python

In figure 2.1 we can see that the linear relationship between 'SalePrice' and 'Yearbuilt' is not very obvious. From the box-plot (figure 2.2) we can easily compare the means, data distribution and outliers.

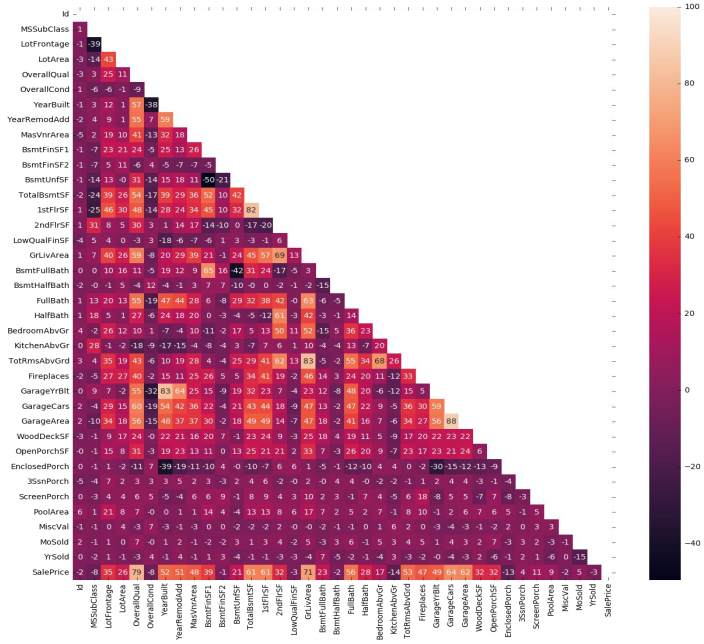

Figure 2.3 Sub-plot by Python

The sub plot shows the correlations by using ' *sns.heatmap(correlations*100, annot=True, fmt='.0f')* ' . It can intuitively show the correlation size. Color can also used for judging the correlation. The lighter the color, the greater the correlation.

Compared with plots in Python, Tableau is easier to operate and more user friendly. For example, In Figure 2.4 we can use the filter to get the numbers of a specific building type rapidly.

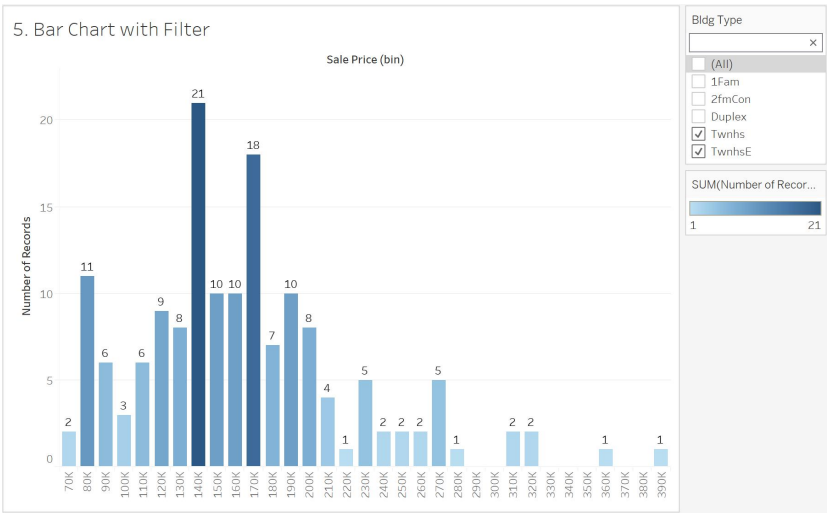

Figure 2.4 Bar Chart with Filter by Tableau

2.2 Data Cleaning

In this part we usually have three main stages: Drop Duplication, Fix missing values and Remove outliers. When fixing missing values, methods are different for different data types. For integers, we usually use means to fill vacancies. For objects, we fill it with 'missing' or the data which exists most frequently.

Under the premise of satisfying the normal distribution, we use '*df['SalePrice'].mean() + 5 * df['SalePrice'].std()*' to define the outliers. After removing the outliers, we can use the box plot to check the result.

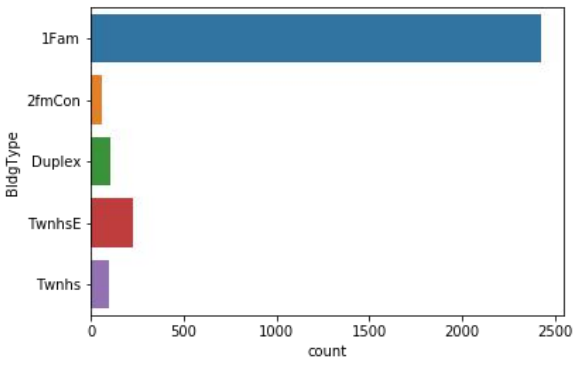

Figure 2.5 Values with the same meaning

Here we found a problem in Figure 2.5, 'TwnhsE' and 'Twnhs' should belong to the same type. So we used '*df.BldgType.replace(['TwnhsE', 'Twnhs'], 'TownHouse', inplace=True)*' to combine them together.

2.3 Feature Engineering

Feature engineering is used to delete useless and irrelevant features. Based on the cleaned data, we can process the data in a deeper level. Here we did five steps to make it perfect:

1. Start with domain language
2. Create Interaction features
3. Group spare classes
4. Encode Dummy Variables
5. Remove unused or redundant features.

In step 2, we created some useful features such as the Property age of building (*df['property_age'] = df.YrSold - df.YearBuilt*). The purpose of step 3 is to avoid the overfitting happens. Step 4 is also very important, given that the data that we have always contain structured data and unstructured data, we have to take measures to change the unstructured data into a operable form. Actually we made it into a matrix.

2.4 Modeling

When modeling is mentioned, we can't ignore a word 'Machine Learning'. According to Zocca et al.(2017), Machine learning is a tool used for large-scaled data processing. Russell and Norvig (2016) highlighted that the importance of machine learning is to adapt to new circumstance and to predict and extrapolate patterns. Similarly, the predicative ability of Machine learning has been widely supported (Zocca et al.,2017, Domingos, 2012). Typically, Machine learning can be mainly separated into two large scales: Supervised learning and Unsupervised learning. In Fig 2.6 we can see the main branches of machine learning models.

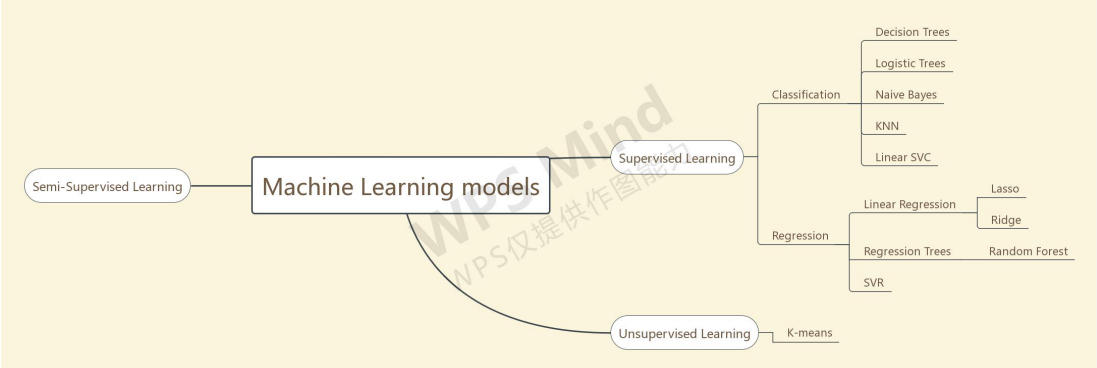

Figure 2.6 Basic Machine Learning Models

2.4.1 Supervised & Unsupervised Learning

Supervised Learning, which is regarded as the first class of machine algorithms, usually uses a set of labeled data in order to classify similar un-labeled data (Zocca et al., 2017). However, in Unsupervised learning, not such a label or other data were provided (Bengio et al., 2015). Depends on the features of data, two main tasks are contained in Supervised Learinig: Classification and Regression. The common algorithms of Classification are: Decision Trees, Logistics Regression, Naive Bayes, KNN, Linear SVC etc. Common Algorithms for Regressions are: Linear Regression, Random Forrest, SVR etc.

Lasso regression and Ridge regression are two famous linear regression models. They can be seen as an improved version of the Linear model. As for a large-scale data, the strength of penalty of Ridge regression is obviously larger than Lasso regression. Different from most of the models in Supervised Learning, each base model in AdaBoost and XGBoost models depends on previous ones.

As deep learning improves, methods which simulate biological organism are welcomed such as neural network (Aggarwal, 2018). It is supported that as sufficient data are given, the accuracy of deep learning method tend to be better than conventional machine learning ones (Aggarwal, 2018, Fig 2.7).

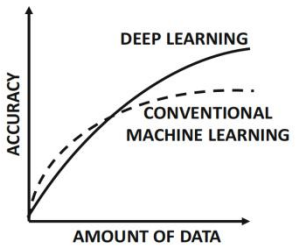

Figure 2.7 Comparison between Deep Learning and Machine Learning Models

2.4.2 Training, Testing and Evaluation

According to Kubat (2017), The simplest scenario will divide the available pre-classified examples into two parts: the training set, from which the classifier is induced, and the testing set, on which it is evaluated.
We usually used MAE (Mean Absolute Error) and RMSE (Root Mean Square Error). Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors.

2.4.3 Ensemble Learning

Ensemble learning can be regarded as a theory that we use the model to make predictions. Usually we separate ensemble learning into three groups: Bagging, Boosting and Stacking. Bagging and Boosting all considers homogeneous models. However, the relationship between base models is different in these two sets. As for Stacking (Fig 2.8), it's a creative method by using two levels models in which the second level model will work out the output of the first level model.

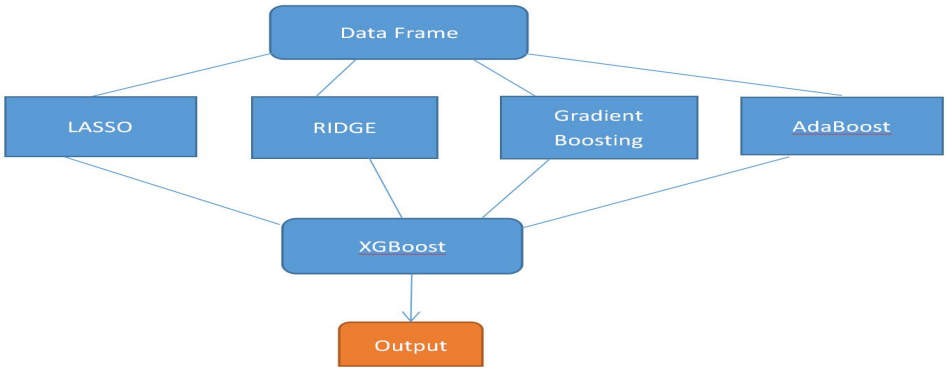

Figure 2.8 Stacking theory in Modeling

2.4.4 Comments and Limitations

In the modeling part, I firstly used Linear regression to check the benchmark and then separate the data into train sets and test sets. In this case I chose the stacking theory for predicting. So in the first level model, I transferred LASSO, RIDGE, ElasticNet, Random forest, Neural Network and AdaBoost algorithms. XGBoost was used as the second level model. The final results were: R square is 0.997, MAE is 3129.596 and RMSE is 4754.261. All parameters proved to be more accurate than each first-level models performance. However, this episode still can be more accurate by tuning the parameters.

3.Conclusion

In this project, we participated in a kaggle competition. The goal is to predict the housing price as accurate as possible which can help buyers and sellers in the market to get fair evaluation of houses. We took several steps including business understanding, data exploring, cleaning, feature engineering and modeling to finally make the final predictions, which help us ranks at top 14% among all teams. The best model we built via stacking combined lasso, ridge, random forest etc. as the first layer and feed into a meta model (XGboost) to have the lowest RMSE on the dataset.

References

Aggarwal, C. C. 2018. *Neural Network and Deep Learning: A Textbook.* Springer: New York.
Bengio, Y., Goodfellow, I.J., Courville, A. 2015. *Deep Learning.*

Domingos, P. 2012. A Few Useful Things to Know about Machine Learning. *Communication of the ACM*. **55**(10). pp.78-87.

Kubat, M. 2017. *An Introduction to Machine Learning. (2ed)* Springer: University of Miami.

Lgual, L and Segui, S. 2017. *Introduction to Data Science:A Python Approach to Concepts, Techniques and Applications.* Springer:Barcelona.

Zocca, V., Spacagna, G., Slater, D., Roelants, P. 2017. *Python Deep Learning.* Packt:Birmingham.