Report of House Price: Advanced Regression Techniques
Date of submission: 23rd August 2020

Number	Name
1.	**Yunhui zhao**

Contents

Part 1:Business Value

- This Kaggle's competition is the housing prediction project, in which buildmodels and analyzes 79 characters related to house prices to predict the final prize of each home based on the house information of Ames, Lowa city dataset. Reasonable and accurate forecasts of house price have profoundimplications for goverments, companies and even consumers. Predicting the house price is beneficial for goverment to regulate markets avoiding gouging price as well as enabling real estate companis have a real knowledge of market. More importantly, it can facilitate consumers to buy house sagaciously.

Part 2：Statistics, Exploration and Visualization

- In order to establish a perfect housing price prediction model, data exploration and visualization are crucial for the reason that such could make us have a deeper understanding of data and pave the way for the subsequent feature engineering.

◆ Basic statistics

- Firstly, this report illustrate that each data record represents the relevant information of each house. There are 1460 training data and test data respectively. Besides, there are79 characteristic columns of data, of which 35 are numerical types and 44 are category types. Besides, statistics for target value - Sale Price as shown in chart 1-1

unit:$

count	1,460.00
mean	180,921.20
min	34,900.00
max	755,000.00
25%	129,975.00
75%	214,000.00

chart 1-1

- The difference between the range of the target values (755000-34900) = 720100 is still very large; 75% of the prices are below 214000, and 50% are between 129965 and 214000; The standard deviation is 79442.5, indicating that the distribution is not very stable.

◆ Exploratory Visualization(correlation)

- Sencondly, to make scatter plot and boxplot using Python's seaborn and matplotliblibrary.

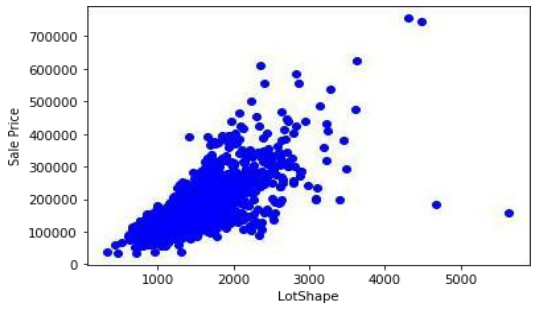


FIG 2-1

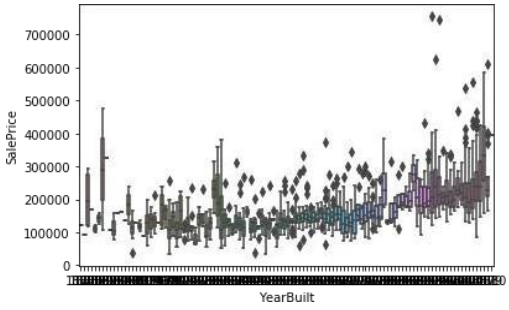


FIG 2-2

- It can be seen that there is a positive correlation between LotShape and Sale Price as shown in figure 2-1. From figure 2-2, the shorter the Year Built, the higher the Sale Price.
- In addition, the next step is to make a heat map using Python's seaborn library. As shown in figure 2-3, variables such as GarageArea and GarageCar have an obvious effect on SalePrice.

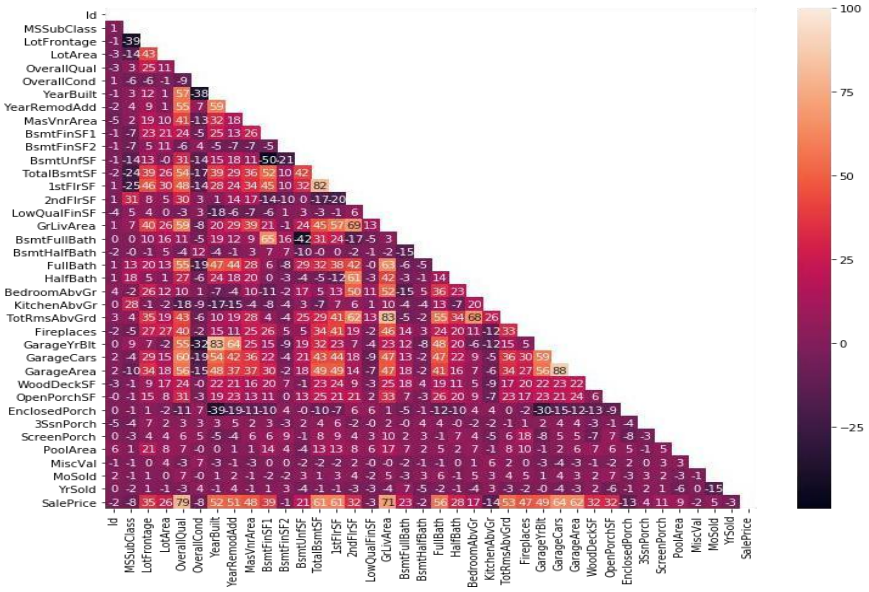


FIG 2-3

- The Data Frame data type function corr() is used to extract all the attributes whose correlation degree is greater than 0.5. In addition to Sale Price itself, there are 10 attributes that are highly related to Sale Price which illustrates on Figure 2-3. We can see that 'OverallQual'.'GrLivArea','TotalBsmtSF''GarageArea''FullBath' and 'YearBuilt' are highly correlated with SalePrice.

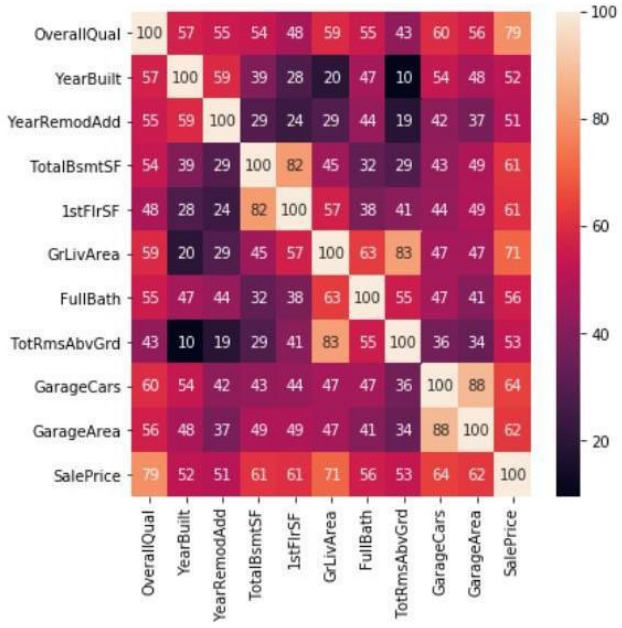


FIG 2-3

Part 3: Data Cleaning

◆ Drop duplication & Fix missing values

- Use ‘Isnull’ function to determine whether the data is null. For missing value of numerical features, inputing them with average value. For missing value of categorical features, we input them using 'missing' / the value with highest frequency.
- Drop duplication means that there are no duplicates of sample.
- From the Figure 3-1, it clearly illustrate that SalePrice did not have a normal diistribution. I use the Numpy function log1p which applies log(1+x) to all elements of the column. After adjustment, the results are shown in the following Figure 3-2.

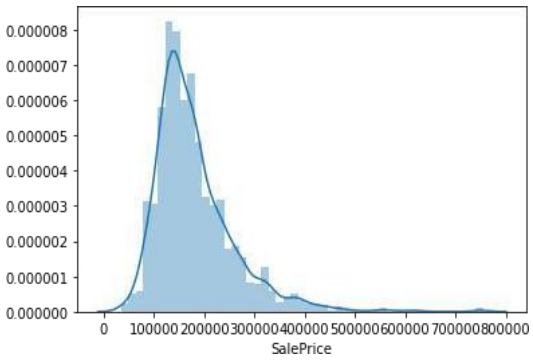


FIG 3-1

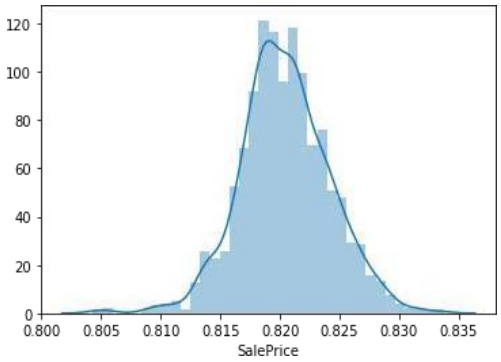


FIG 3-2

◆ Remove unwanted outliers

- Remove lot_size outliers of SalePrice

Part 4: Feature Engineering

- After cleaning the dataset, the next step is to engineer features that can help our predictive models.This is not an exhaustive list of all types of feature engineering. There are limitless possibilities for this step, and it's a skill that will naturally improve as you gain more experience and domain expertise.

◆ Group Sparse classes

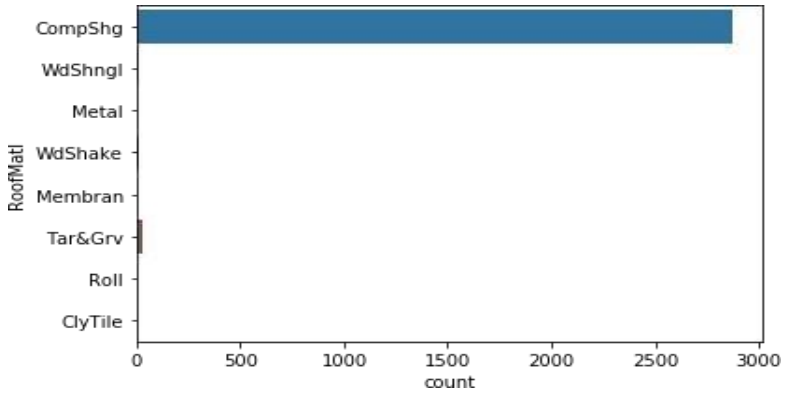


FIG 4-1

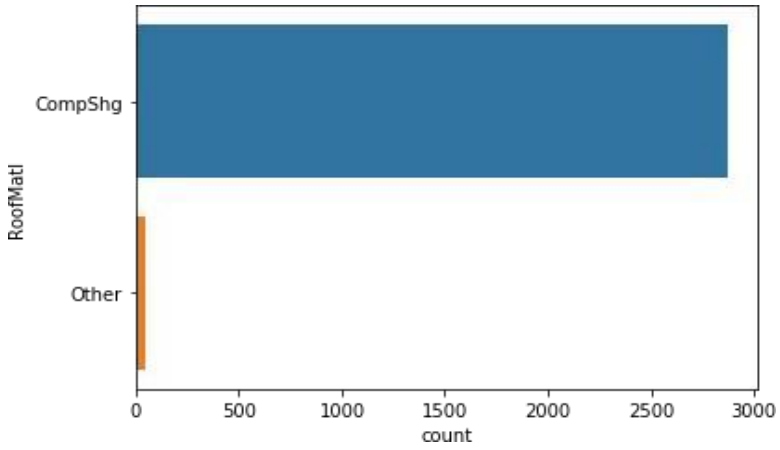


FIG 4-2

- It can reduce overfitting problem when the sample set for a specific class is super small. And then unnecessary data can be merged(as shown in FIG 4-1 and FIG 4-2)

◆ Encode dummy variables

- Secondly, because Python machine learning algorithms cannot handle categorical features directly, and then we could encode dummy variables to create a new dataframe with dummy variables for our categorical features in order to enable categorical feature exchange into the numerical feature.

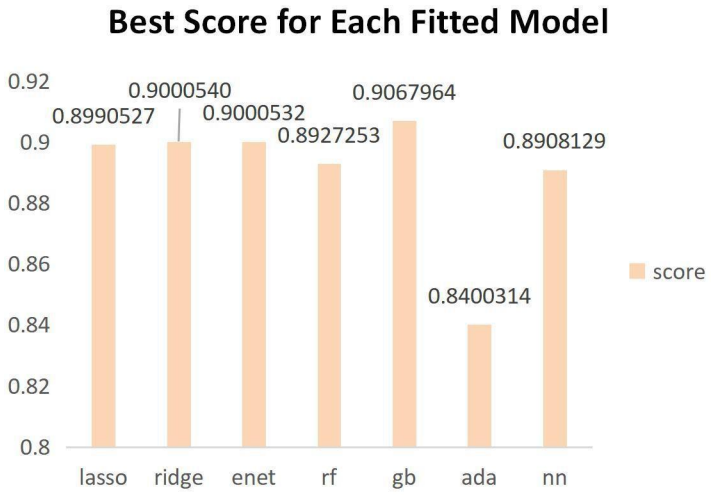
Part 5: Modeling

- When we started to build our model, the first thing that we need to do is to import ML Models creating a pipelines for passing the parameters and test them on the dataset. I mainly use Eight algorithms and cross validation(set cv=10) to evaluate effect. These are ① Lasso②Ridge③Random Forest④Gradient Boosting⑤AdaBoost⑥Neural Network⑦Elastic Net⑧XgBoost
- For instance, I used Gradient Boosting to boost tree hyperparameters and multiple step sizes, tree depth and alpha values were set and parameter adjustment was adjusted perfectly. (as shown in Figure 5-1)

```
]: # Boosted tree hyperparameters
gb_hyperparameters = {
    'gradientboostingregressor__n_estimators': [100, 200],
    'gradientboostingregressor__learning_rate' : [0.01,0.05, 0.1, 0.2],
    'gradientboostingregressor__max_depth': [3, 5, 10],
    'gradientboostingregressor_alpha':[0.9,0.99,0.999],
}
```

FIG 5-1

Part 6：Evaluation and results



- The results are shown in Figure 6-1. The tree model(rf and ada) is not as good as linear model(lasso and ridge).
- Gradient Boosting seems to perform best.

FIG 6-1

- Validation was carried out on the validation dataset, and the R^2,MAE,and RMSE of each model were calculated respectively as shown in Figure 6-2. It seems that Random Forest has the best performance according to the R^2.
- As regard to the stacking model performance, I use 7 algorithms as the basic model and XgBoost as the second layer model and the final evaluation results showed an obvious improvement compared with the single model.(In this diagram, the value of R^2 multiply 10^4 so as to fit the other values of this bar chart.)

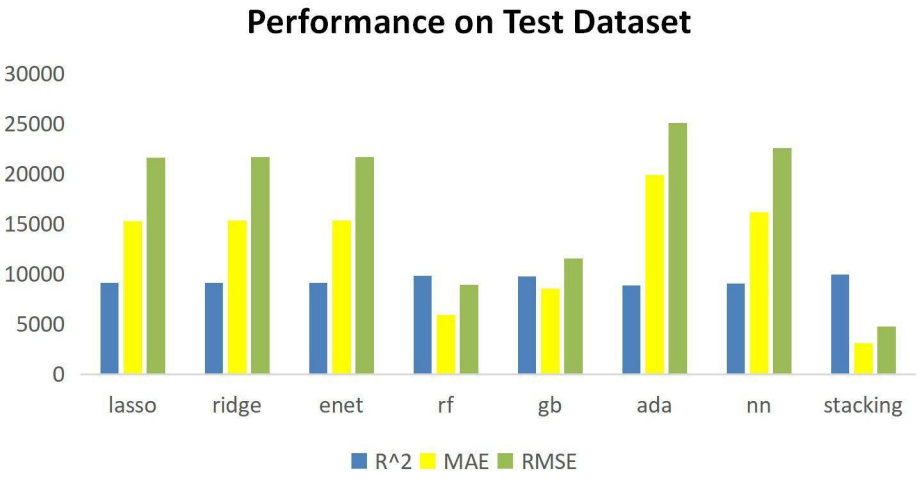
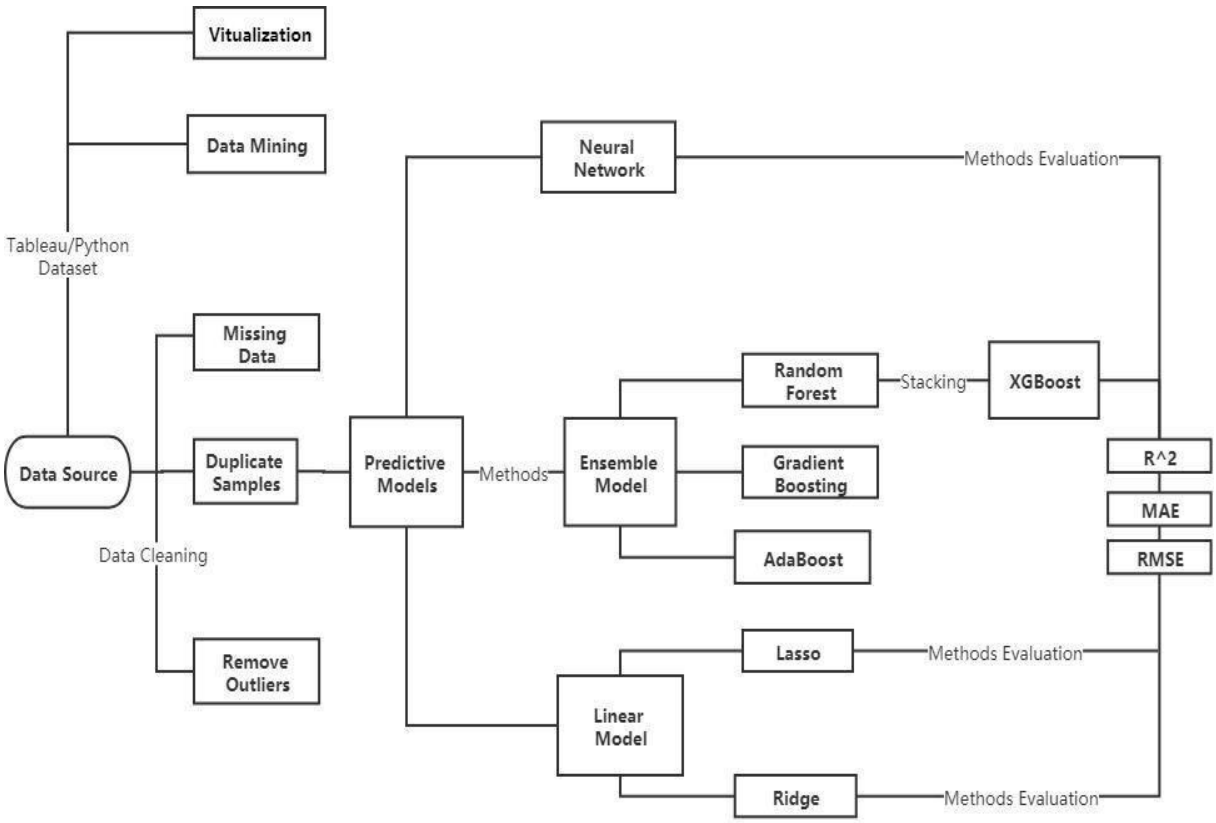


FIG 6-2

	lasso	ridge	enet	rf	gb	ada	nn	Stacking
R^2	9153.5	9145.3	9144.2	9854.8	9757	8855.7	9076.1	9959
MAE	15278	15337	15347	5897	8529	19922	16165	3129.6
RMSE	21604	21708	21722	8948	11574	25117	22570	4754.3

Summarize


Vitualization
Data Mining
Tableau/Python Dataset
Missing Data
Data Source
Duplicate Samples
Data Cleaning
Remove Outliers
Predictive Models
Methods
Neural Network
Ensemble Model
Random Forest
Stacking
XGBoost
Gradient Boosting
AdaBoost
Linear Model
Lasso
Ridge
Methods Evaluation
Methods Evaluation
Methods Evaluation
R^2
MAE
RMSE