Stat 542 First Project- Predict the Housing Prices in Ames

Team: RSS

1. General Idea

(1) Data cleaning steps

First, we eliminate the columns including more than 730 NAs (25%) based on the other expert's experiences.

Then, we filled the rest NAs with function mice. Since leaving out available data points deprives the data of some amount of information and also by replacing missing data with mean or median might increase bias, consequently, we decided to apply mice function. It imputes missing values with plausible data values. These plausible values are drawn from a distribution specifically designed for each missing data points. However, it would take more times.

At last, we found that the response and some feature values are right-skewed. Log transformation can turn them into normal distribution very well.

(2) Model selection steps

In the beginning, we tried many models (step BIC, Random Forest, SVM, PCA, Ridge, Lasso, neural network etc.). And then we apply the self-evaluation steps to these models.

We are asked to choose one simple linear model, so one of AIC, BIC, Ridge or Lasso must be chosen since their final models are simple linear models. Among those four models, AIC has the least rmse value.

Also, we choose SVM because it has the best result in Kaggle. We get a 0.12078 rmse and ranked top 1000.

Finally, we applied random forest just because we want to try the model mentioned in the class. The result is not good enough in Kaggle, but it performs pretty well in self-evaluation and it's time-saving compared to other model we applied. So we choose to use this model in our final code.

We also applied some other methods but they all have different problems. Neural network can’t beat SVM if we set the size of hidden-layer or iteration times low. But it would take forever if we want to get a more precise result. Polynomial models are extremely dangerous to cause an overfitting when there are many features. Also, PCA and Lasso won’t reach the accuracy we need.

2. Models

(1). Preprocessing

The reason of time-consuming is the MICE steps. However, the parameters set in the mice step will influence the final result by 0.05, so we just choose to spend more time in the data cleaning part.

Model result:

Run time:

```
user     system     elapsed
671.16     21.13       709.30
```

It will take 10 minutes to clean the data.

(2). Step AIC

We are asked to include a simple linear model, so we just applied step AIC to get a simple linear model. In practice, we could just apply Step AIC in the prediction because the accuracy is already pretty satisfying.

Model Result:

Self-evaluation times: 500 times

Self-evaluation quantiles:

```
       0%                 25%              50%                75%            100%
0.1069510     0.1305383     0.1399910     0.1524692     0.1931223
```

Self-evaluation standard error of rmse: 0.01492717 / 500 times

Kaggle evaluation: 0.13687

Run time:

```
user     system     elapsed
1.53       0.18         1.81
```

(3). Random Forest method

We heard this model from class and want to apply it in this project. It has done a pretty good job when we self-evaluate it, but not so well in Kaggle evaluation.

Model result:

Self-evaluation times: 500 times

Self-evaluation quantiles:

```
      0%          25%          50%          75%         100%
0.1020275    0.1142701    0.1196771    0.1239120    0.1391330
```

Self-evaluation standard error of rmse: 0.00676
Kaggle evaluation: 0.14373
Run time:

```
user    system    elapsed
0.00     0.01       0.01
```

(4). SVM method

SVM method will be the model we choose if there are only one model needed. It has a good self-evaluation result and gets a pretty good score in Kaggle. We could evaluate the cost more precisely or try some different kernel, but it will take forever to run the code 500 times.

Model Result:
Self-evaluation times: 500 times
Self-evaluation quantiles:

```
0%            25%            50%            75%          100%
0.08980102    0.11620165    0.12502560    0.13400159    0.16680586
```

Self-evaluation standard error of rmse: 0.01256764/ 500 times
Kaggle evaluation: 0.12081
Run time:

```
user    system    elapsed
0.02     0.02       0.05
```

Note: The run time did not include the code finding the cost parameter. Cause in practice I may just use the default 1 in RBF kernel and won't influence the accuracy.

3. Conclusion

1. The models we choose all get an average rmse below 0.145 and ranked top 50% in Kaggle.

2. The code will take nearly 15 minutes to run, which is acceptable.

3. In self-evaluate process, we choose to re-sample 360 observations as test data 500 times and calculate the standard error and quantiles of rmse. The standard error could tell how much our models rely on specific data.

4. Simple step AIC is the model we will choose in practice since it's simple and can meet the need. In fact it's already pretty precise.

SVM model will be chosen if we want to reach the highest accuracy.