CSCI-SHU 360 Machine Learning
Final project paper

Group 18: Jiachen Huang, Yijing Zhou

Abstract

Housing rent plays a remarkable role in city residents' lives, especially in today's Chinese metropolitans. In this project, we focus on Shanghai's rental market and tried to build a model to predict the rent, given apartment attributes and public transportation accessibilities. Our housing dataset contains all the apartment resources available for renting in Shanghai at Ganji.com, which is a leading online rental platform. We also include public transportation data in terms of bus stops and metro stations. With a clean data of 19299 rows and 54 columns after processing data, we applied multiple regression algorithms on the dataset and used RMSE to evaluate the performance. A neural network model was also included for reference. Hyperparameter tuning and ensemble methods were applied to obtain better performance, and generated a final integrated model reaching a RMSE of 0.22 and a RMSPE around 0.26. We also derived an indicator to find undervalued apartments and plotted their distribution.

1 Introduction

The initial interest in this topic arises from ourselves. As seniors, we chose to rent apartments near campus. To find the best apartment, we have lists of factors to consider: price, facility, decoration, location, etc. And this leads to several questions worth discovering further: What are the factors influencing rental prices? Is there a way to assess the fair price of an apartment given location and apartment attributes? How could we find undervalued units and offer reference for potential tenants? Motivated by these questions, we turn to machine learning for solutions.

Apart from our own interest, this is also a society-wide issue. As the economic center of China full of opportunities, Shanghai attracts millions of migrants who wish to settle down. According to statistics, there are 9.6 million migrants in Shanghai, among which 80% rent an apartment. In addition to migrants, a higher proposition of local citizens are forced to rent an apartment under the pressure of rocketing apartment price.

There are already some relevant machine learning work conducted in the field of real estate, but most of them target at the housing price whereas we concentrate on the rental price. Hence, our topic is still novel and worth probing into.

2 Dataset and Features

Mainly composed of housing attributes and public transportation data in Shanghai, the data was obtained from Metrodatateam's city database and Ganji.com by implementing a web crawler. The raw housing dataset contains many columns of text, which were then processed into numeric types after using the regular expression and generating dummies. Another task was to import ArcGis data and transform public transportation sites into attribute of housing units. Combining two datasets together, we have labeled records $(x^{(i)}, y^{(i)})$ for 19299 units, each consists of n=53 features including size, district, decoration style, floor, number of bus stops nearby. First stage data cleaning eliminated outliers and data points with missing attributes, leaving a dataset with

apartments whose size is smaller than 200 and rent is lower than 20000.

2.1 Geometric Data

Specifically, we calculated the number of bus stops within around 500 meters and metro stations in approximately 1 kilometer range to be public transportation accessibility measures for apartments. Let lat_i, lng_i be the latitude and longitude of stop i, and lat_j, lng_j be the latitude and longitude of apartment j.

$$N = \sum_{i,j} \mathbf{1}_{(lat_i - lat_j)^2 + (lng_i - lng_j)^2 < C}$$

The number of transportation sites counted as the"near" a given apartment is calculate with the formula above, setting different thresholds C for bus stops and metro stations respectively. Intuitively, we made the apartment the center and included all stops located in the circle of a certain radius. Figure 1 shows the location of all metro stations in Shanghai.

Figure 1

2.2 Skewness and Logarithm

Calculating the skewness score of all variables, we dropped 7 dummies with extreme skewed distributions to improve the prediction power. In addition, logarithms of price and size are chosen to supplant the original data to make sure errors in predicting expensive apartments and cheap apartments will affect the result equally. Figure 2 shows the distribution pattern and skewness of selected variables.

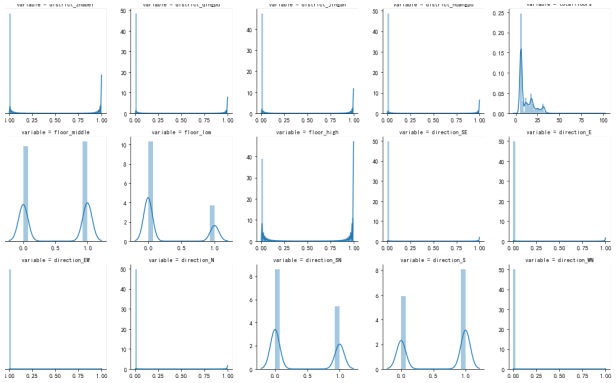

Figure 2

2.3 Feature Crosses

Another data preprocessing before the analysis is to adding interaction terms. Even though we have obtained 53 variables after the first stage data process, it is still meaningful to add few new variables to increase the explanatory power. For example, a counter-intuitive phenomenon was observed that those units with elevators actually have a lower rents in average. This may related to the construction year for apartments in different district, which means an apartment without elevator are likely to be old ones that locates closer to downtown area. Therefore, adding an interaction term from elevator and district variables is reasonable. We generated feature importance scores by LASSO (shown in figure 3) and chose several new variables to generate after several trials, which indeed improved the performance of our model.

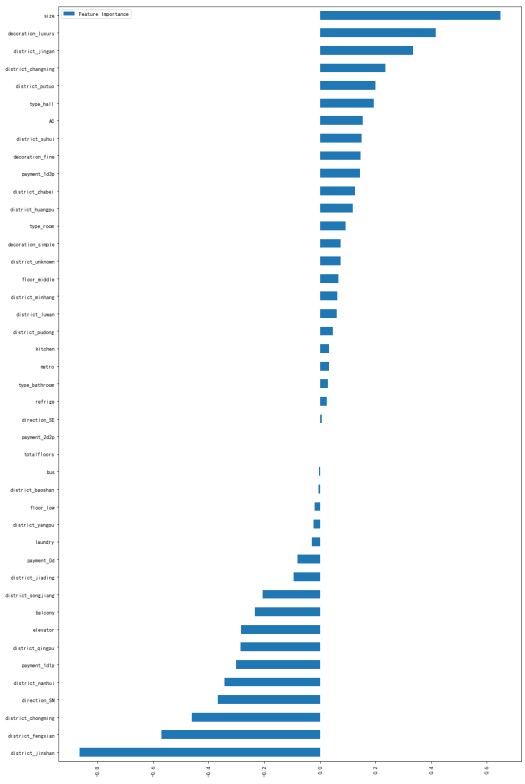

Figure 3

2.4 PCA and Dimension Reduction

PCA and XGB threshold were both applied to the data hoping to limit the number of attributes. However, attempts to reduce the dimension always lead to a worse performance in the later regression. So we decide not to include and dimension reduction technique in the code.

3 Models and Evaluation

For regression models, we try to solve the following problem: given a processed list of features for an apartment, we would like to predict its potential rent.

3.1 Baseline and Evaluation Metric

Linear regression is a natural choice of baseline model for regression problems. A dummy regression which returns the mean is also added as a baseline for reference. The performance was measured by Root Mean Square Error (RMSE) of predicted rents and actual rents. Training error and testing error were both calculated to check overfitting. Our linear regression model generated a rmse of 0.47595, which is based on the difference in the log-transformed sale prices.

3.2 Regressions

After choosing linear regression model as the baseline, we applied multiple algorithms to check their performance. We started our model by picking 13 frequently-used algorithms and plugged in data with arbitrary default hyperparameters. These 13 models are: Linear Regression; Ridge Regression; Lasso Regression; Random Forest; SVR; Linear SVR; Gradient Boosting; Elastic-net; SGD; Bayesian Ridge; Kernel Ridge; ExtraTrees and XGB. Six of them with the lowest training and testing errors (Random Forest; Gradient Boosting; SVR; Kernel Ridge; Extra Trees and XGB) were chosen into hyperparameters tuning. Since all these models are complicated and the tuning process would be extremely time-consuming using grid search, we turned to random search for tuning with a bigger range and faster process. In order to avoid overfitting, we used a 5-folders cross validation and checked the explicit training error with testing error each time.

Lasso generated a RMSE of 0.511728, which was better than our baseline model. Other than lasso, ridge regularizer which generated a rmse of 0.475896 was also applied. The performance was not significantly better than the baseline, indicating regularization may have limited improvement to this problem.

Model	Training error	Testing error
Linear Regression	0.475950	0.4790
Ridge	0.475896	0.4789
Lasso	0.511728	0.5147
Random Forest	0.273322	0.2533
Gradient Boost	0.340178	0.3432
SVR	0.344476	0.3471
Linear SVR	0.529947	0.5378
Elastic Net	0.476835	0.4805
SGD	0.484469	0.4842
Bayesian Ridge	0.475883	0.4789
Kernel Ridge	0.366681	0.3614
Extra	0.248990	0.2491
XGB	0.343567	0.3470

Table 1

Support vector regression (SVR) with Gaussian and linear kernels were also fitted to the features. Hyperparameters of both models were select after multiple trials of tuning to ensure the best performance. SVR with Gaussian kernel model generated a rmse of 0.344476 and that of linear kernel generated a rmse of 0.529947. SVR with Gaussian kernel performed better than our baseline model. Whereas SVR with linear kernel generated a relatively high RMSE due to the kernel's unfit with the dataset in this case.

After running all the 13 models, we found most algorithms with good performances compared to the baseline. The detailed performance of each model could be found in Table 1. The six models selected to conduct ensembling are: Random Forest, Gradient Boost, SVR, Kernel ridge, Extra Trees and XGB, as highlighted in Table 1.

3.2 Hyperparameters Tuning

Tuning hyperparameters provides a significant improvement of each model. The detailed performance of the six models is shown in Table 2. Among them, XGB provided the lowest RMSE error of 0.2225, which was remarkably lower compared to the default model.

After Tuning

Model	Training error	Testing error
Random Forest	0.2350	0.2317
Gradient Boost	0.2395	0.2354
SVR	0.2766	0.2614
Kernel Ridge	0.3174	0.3271
Extra	0.2448	0.2378
XGB	0.2225	0.2256

Table 2

3.2 Ensemble

In addition to single models, we succeeded in applying ensembling methods to improve our result further. We used two approaches of ensembling: weighted average and stacking. By assigning weights to each model based on their performance, we achieved a RMSE error of 0.2228. Moreover, choosing the best two models (random forest and XGB) and assigning them with equal weights also generated good performance, with an even lower RMSE error of 0.2222. Finally, using the more complicated stacking provided a much better result of 0.2188. Figure 4 shows the diagram of our stacking model.

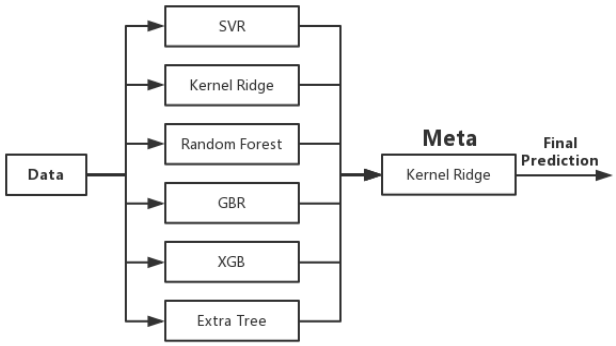

Figure 4

3.3 Neural Network

Initially, we expected that neural network might help with our result. However, after several trials of neural network using Keras dealing with overfitting, the outcome was not satisfying. In the future, we might use convoluted neural network or other advanced models to handle this.

4 Results

4.1 RMSPE on testing set

The final model with stacking achieved a good enough score using cross-validation. As for the final result, we divided the original dataset into training and testing set and apply the stacked model. While the RMSE score is 0.2201 for the test set, we also add a RMSPE function to evaluate the relative deviation of prediction model on the test set.

$$RMSPE = \sqrt{\frac{1}{N}\sum_{i}^{N}\left(\frac{\hat{Y}_i - Y_i}{Y_i}\right)^2}$$

Using rescaled rent prediction (take the exponential), our RMSPE reached the level of 0.26, which indicates our model having a reasonable prediction power.

4.2 Deviation and Undervalued Units

One proposed application of our project is to find undervalued units and offer reference for potential tenants. Therefore, we defined a deviation function that tracks the deviation of the residual between prediction and underlying rents, from the overall RMSPE.

$$De_i = \frac{\frac{\hat{Y}_i - Y_i}{Y_i}}{RMSPE}$$

The distribution of the deviation indicator is close to a Gaussian distribution by CLT. In figure 5 we plot apartments with deviation value larger than 1.96, which means we are 95% confident to say that these units are undervalued.

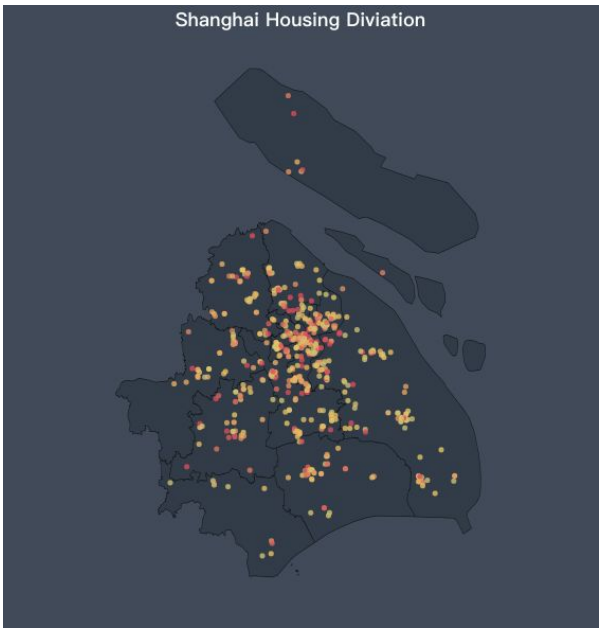

Figure 5

5 Future Work & Conclusion

There are several work in this project that can be improved in the future. First improvement could be conducted by adding more features and data points to expand the entire dataset. This could help with building more precise models and reducing overfitting. A second improvement could be made in the process of tuning hyper parameters. Instead of using random search, grid search should be applied to find out the best parameters more accurately, even though at the cost of enormous computing time. A third improvement might be necessary regarding the stacking process. After building the stacking model, we could also do parameter tuning for this meta model, which might improve its performance further. If time allowed and we have completed all the steps above, we want to apply our model to different cities in China and spot similar and different factors lying behind each city's rental price. Furthermore, our model could be compared with those existing models built upon housing

price in order to better understand the distinction between housing price and rental price.

In short, our original goal to predict rental price in Shanghai based on house attributes and public transportation accessibilities has been well achieved. We sincerely hope that under continuous revising, our model could be successfully applied to the rental market and help with tenants' decisions.

Works Cited

Massquantity. "Massquantity/Kaggle-HousePrices." *GitHub*, github.com/massquantity/Kaggle-HousePrices/blob/master/HousePrices Kernel.ipynb.

Sadayuki, Taisuke. "Measuring the Spatial Effect of Multiple Sites: An Application to Housing Rent and Public Transportation in Tokyo, Japan." *Regional Science and Urban Economics*, vol. 70, 2018, pp. 155–173., doi:10.1016/j.regsciurbeco.2018.03.002.