Price Prediction of Major Rental Platforms in Shanghai via Machine Learning Methods

Xinyao Han & Eden Wu & Yile Xu

Abstract— **Our project aims at predicting rental house prices to give an indication of fake rental house identifications. Initially, we planned to classify real and fake rental houses, while our attempts to get credible and abundant fake labels on rental houses did not succeed. Therefore, we turned to predict rental house prices to use price as a potential index of fake houses. Regression models are applied and the overall performance is relatively satisfying. After feature importance analysis and hyperparameters tuning, we discover Extra Tree gives the best performance. Finally, Ensembling Methods are applied to improve our models. Our final prediction model with an MAE of 587.57 gives us confidence in predicting rental house prices and helping to filter some potentially fake rental houses.**

Index Terms— **Rental Housing, Price Prediction, Machine Learning, Ensemble Methods**

Acknowledgement— **We thank our professor Enric Junqué de Fortuny from NYU Shanghai Computer Science Department, who provided insight and expertise that greatly assisted the project.**

I. Introduction

When people choose to rent houses, most of them will first search through renting websites, and then inspect real houses to save efforts. Problems such as online fake renting are common though convenience it seems. It is widely reported by official China media that around 80% of people encountered fake houses when renting. Currently, there are no existing solutions to distinguish fake rental houses.

As one of the major social problems, we concentrated on the frequent occurrence of rental platforms' fake listings. Due to the lack of credible fake rental housing labels and the opaque historical recordings of fake rental houses on platforms, we turned to examine other potential indexes on fake rental houses to help people make better choices. Rental houses price index could indicate the trustworthiness of platform listings and shield lights on fake house identification.

As indicated by the South China news survey [1], major problems on fake rental houses are excessively beautified pictures, faking detailed information of houses as well as attracting people with false low prices. For individuals to distinguish fake houses, the Chinese government has provided two solutions: compare the average prices of identical houses and pay attention to the textual descriptions.

Therefore, we will focus on rental house price prediction in this paper. Based on the price prediction, we hope to offer rental platform users a reasonable rental house price reference based on given housing information. This price reference would help to filter some of the potentially fake rental houses.

II. The Dataset and Features

'**Shanghai rental houses**' is a dataset containing around 20,000 data with 155 variables crawled and processed from widely used rental platforms in the Shanghai district. Anjuke, Beiker, Lianjia, Ziroom, and 58. Procedures, as follows, are done to get rental prices and other influencing features:

A. *Get Data and Influencing Indicators*

- Crawled data from housing rental web pages through crossing the anti-crawler encryptors and applying Optical Character Recognition [2] to get rental housing dataset. 8 variables are attained

- Integrated dataset through selecting common and crucial features since different rental platforms have different data ranges and features

- Applied Baidu Map open-source API [3] to retrieve longitudes, latitudes and surrounding subway information based on Location variable

- Added geographical analysis using Hexagonal Hierarchical Spatial Index (H3) [4] to include location related influencing factor as dummy variables

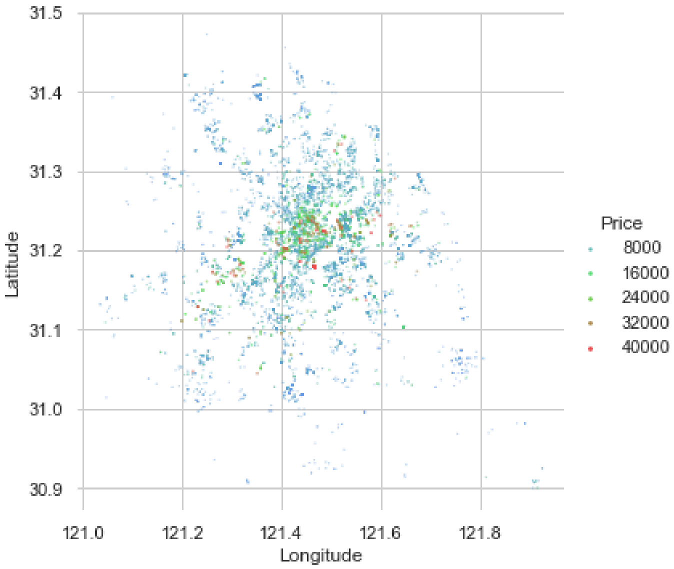

Fig. 1. Y Distribution with Longitude and Latitude

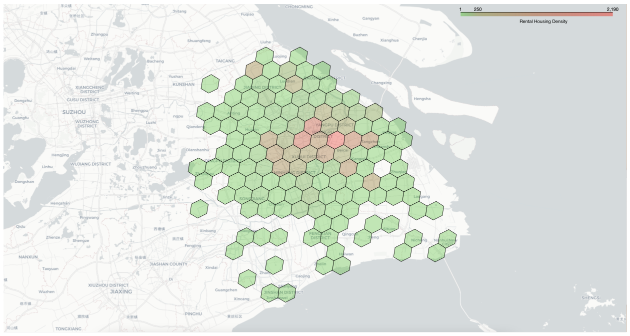

Fig. 2. H3 Index and Rental Density Visualization in Shanghai

With the rental houses dataset, we then split the dataset using 0.2 factor. Our data preprocessing contains:

B. *Data Preprocessing*

- Investigated outliers on the Price variable: we clipped high value over 40,000 and low value under 1000 since the long-tailed distribution of price contains noises and will bias the prediction result

- Handled missing values using mean value, 0 and algorithms for different features: Exquisite and OpenToVisit variables, with NA values, were filled with 0; Bedrooms, Floor, Longitude and Latitude variables were filled with mean values; Bathroom variable was filled with an algorithm by computing Area variable/120 rounded up

- Applied StandardScaler to standardize the datasets. Our features have vastly different scales, which may degrade our model performance. Using StandardScaler can rescale the dataset such that all feature values are mapped to a standard distribution.

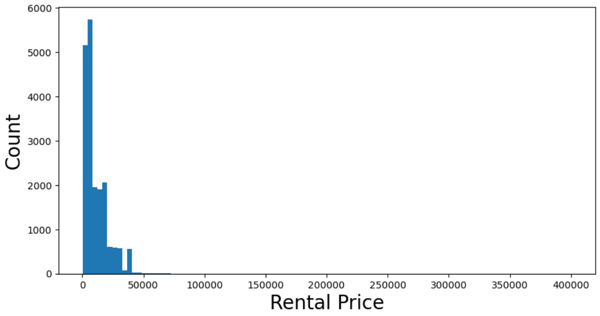

Fig. 3. Y Distribution Before Clipping

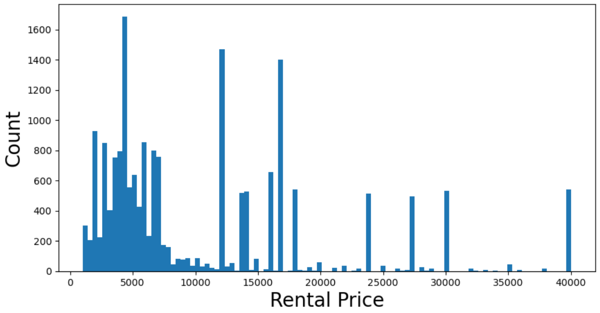

Fig. 4. Y Distribution After Clipping

III. Explanation of the Methods Used

A. *Evaluation Method*

- Evaluation function used for the price prediction model is **Mean Absolute Error** (MAE). Compared with **Mean Square Error** (MSE) and other evaluation functions, MAE is able to show the price differences more directly. It is expressed as follows:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}.$$

Since MSE squared the Error ϵ, we can see that if e is greater than 1, this value will be $>>|e|$. In order to minimize the error caused by the outliers, the MSE model will try to be close to the outliers, that is to give more weight to outliers. This will affect the overall model effect. Compared with MSE, MAE has better performance when the data is not conducive to the prediction of abnormal values. When there are outliers, the mean is better than the square. Therefore, MAE has a higher rolling performance than MSE.

B. *Training Procedures*

- **Feature Selection.** To extract features with high influencing power, we used Lasso Regression to measure the effect of different features to the Price variable, It is expressed as follows:

  $$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

  Model result reflects the correlation level of individual features to price, expressed as $\theta_0, \theta_1, \ldots, \theta n$. Features with relatively low correlations of an are dropped. In this case, features with relatively low correlations, with an absolute θ under 200, were dropped. Total features were reduced from 155 to 103 after feature importance analysis.

- **Rough Estimation.** We figured out the most precise models for rental price prediction through using default scikit-learn model hyper parameters. We roughly compared 10 prediction model performances and selected the top 3 models from Linear Regression, Ridge Regression, Lasso, Random Forest, Gradient Boosting, Linear SVR, ElasticNet, Bayesian Ridge Regression, Kernel Ridge Regression, and Extra Trees. Performances of selected models are examined in detail below.

- **Fine-tuning and Ensemble.** The selected models were fine-tuned by Random Search with Cross-Validation. We finally applied Weighted Average and Stacking to fine-tuned models in order to improve prediction performances. This method further boosted our prediction accuracy.

C. *Models*

- **Random Forest.** Random Forest is a kind of ensemble model that combines different decision trees in order to optimize the result. In this paper, we are using RandomForestRegressor provided by sklearn. We used sklearn's random search CV to run 20 epoch between 1 to 100 estimators in order to get the best results.

- **Gradient Boosting.** Gradient Boosting is a technique for regression problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. The referred algorithms are as follow:

  $$\hat{f}_T(x) = \sum_{t=0}^{T} \alpha^{T-t} \cdot h_t(x)$$

  A strong learner $\hat{f}_T(x)$ is trained from many prediction results $h_t(x)$ from weak learners. T is the total number of decision trees and alpha is the learning rate of this algorithm. In this paper, we are using GradientBoostingRegressor by scikit-learn. We also used scikit-learn's random search to run 20 epochs between 1000 to 5000 estimators, a random subsample rate from 0 to 1, and the learning rate of 0.05 or 0.1.

- **Extra Trees.** Extra Trees is an ensemble machine learning algorithm that combines the predictions from many decision trees. It can often achieve as-good or better performance than the random forest algorithm, although it uses a simpler algorithm to construct the decision trees used as members of the ensemble. N_estimators are randomly set between 1 to 100 for RandomizedSearchCV.

D. *Ensemble Methods*

Ensemble Methods are very important to model training and improvement. In machine learning, we use those methods to combine the fine-tuned models that we trained in order to get one optimized predictive model. Under rental house price prediction, ensemble methods are applied to further boost performances. The ensemble methods we used here are Weighted Average and Stacking [5].

- **Weighted Average.** The simplest ensemble can be done through the Weighted Average method, that is, simply weighted and averaged the performance of each model. The weight of each model can be obtained through exhausting grid search or random search. In our training, we used a random search of 20 iterations to get an optimal result.

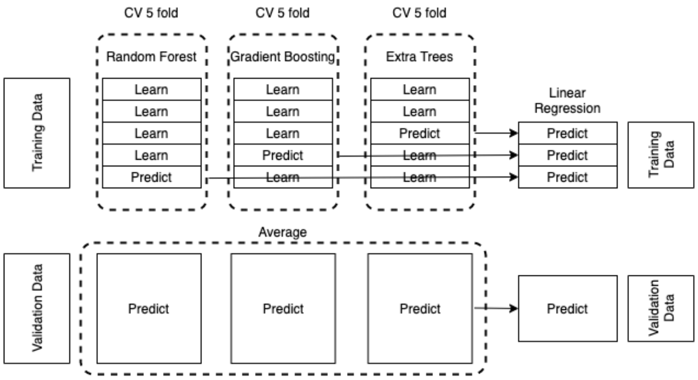

Fig. 5. Stacking Topology

- **Stacking.** Stacking is a powerful method in model ensembles. It uses a meta-learning model to learn the best combination of several machine learning algorithms. We choose the stacking method as an important ensemble method. It is a strong method for result enhancing since we have 3 strong models — Random Forest, Extra Trees, and Gradient Boosting! We choose the models that we fine-tuned previously as the models of our first layer, and then Linear Regression as our meta model to further train our results.

IV. Results

In this section, we are using both the MAE of the training set and the validation set to evaluate our models. Both results are showed showing that our model is accurate and not over-fitting. After we applied the measuring standard to various regression models, it appears that Random Forest, Gradient Boosting Regression, and Extra Trees do better than other regression models. As a benchmark, comparing to the 0-dim Linear Regression(mean price) with an MAE of 7271.53, the models we selected all have a good performance of under 1000, which is a great reduction. Thus we modified some key factors of the models and ensemble them into a stacking model to improve their performance.

Using RandomSearchCV for 20 epochs to each of the models we chose, the training MAE of Random Forest with default hyper parameters is 452.88, this value changed to 449.66, which did not decrease greatly. The reason behind this might be that some patterns are followed when calculating rental price thus some cases are very similar in conditions, making decision trees easier and shallower to build; Training MAE Gradient Boosting Regression with default parameter is 912.15, this value changed to 474.60. Since GBR is very powerful while exhausting, I believe if we train it for more estimators, it can improve further way; Training MAE of Extra Trees with default parameter is 442.87, this value changed to 436.77, which is also a trivial improvement. For validation MAE, GBR reducing sharply from 996.28 to 636.98 while the other two stayed the same.

After applying the ensemble methods the result seems to improve greatly especially on the validation set. Validation MAE is reduced to 584.92 after Weighted Average and 587.57 using Stacking Model. Such an increase is significant since the smallest validation MAE before ensemble is 594.99 from Extra Trees. We believe that if we do further training to the Weighted Averaged model and changes the meta model to more complex models such as Polynomial Regression, there are some more space for improving. However, from figure 7 we can see that since the differences between prediction price and actual renting price is already very small, following big improvements will be very hard to make and highly depend on the appending data distributions. The prediction result before and after modification is attached below.

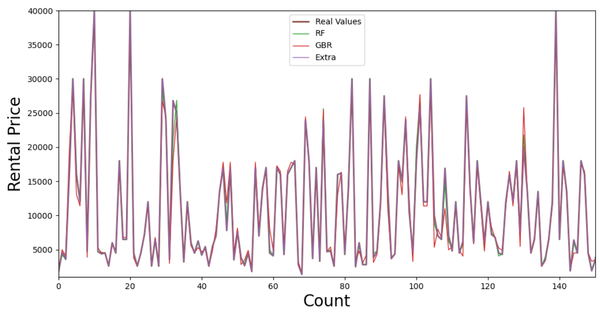

Fig. 6. Prediction Accuracy Before Fine Tuning and Ensemble

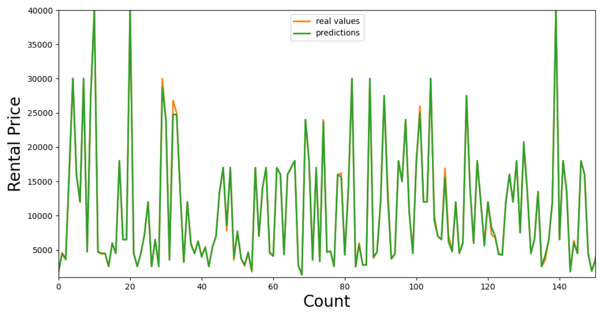

Fig. 7. Prediction Accuracy After Fine Tuning and Ensemble

V. Conclusion and Future Work

In conclusion, our works contribute to online rental platform users in the Shanghai area. Using our rental houses price prediction results as reasonable price references, rental platform users can roughly estimate the rental house prices of their desired houses that meet their requirements and also be able to roughly check whether a piece of housing information on these platforms is potentially real based on the comparisons between its price on the platforms and its estimated prices. This can prevent rental platform users from wasting time on contacting the rental agents to check while often for no reliable information.

In the future, for price prediction, we will try to adapt more complex machine learning algorithms into our model, and further investigate these models, especially the combinations of different models.

We plan to put our effort into 5 ways: First, we will try out some more coupling effects of multiple regression models, since sometimes coupling some correlated models will improve the prediction results even better than simply choosing those best models; Second, we should try our model on different data, to check the "relearn" ability of machine learning models on whether it still works well on different datasets; Third, we will try out the combination of Machine Learning and Deep Learning methods; Fourth, we would find out the driven factors of tree-based models for our rental house price prediction and try to optimize those factors for better performances [6][7][8]; Lastly, since our current training processes still take a large amount of time, we wish to find a faster way to fit those complex models.

Besides price prediction, we still want to conduct fake rental house identification. It's one of the main concerns for online housing platform users when online house renting takes more popularity while in urgent need of supervision. We will need more labeled datasets to do the fake-real identification and also need to change our models into clustering models. For rental house labels, we plan to investigate rental houses by ourselves by visiting and calling. Alternatively, this problem could also be handled with Deep Learning methods such as Auto Encoder by studying the features. More features such as images and text descriptions could be included to conduct classification.

VI. Appendix

TABLE I
PREDICTION RESULT

Model	MAE	
	Training Set	*Validation Set*
Mean Price	7271.53	7126.62
Kernel Ridge	882.22	988.19
Random Forest	449.66	613.62
Gradient Boosting	474.60	636.98
Extra Trees	436.77	594.99
Weight Average	447.68	584.92
Stacking Model	429.25	587.57

TABLE II
DATA DESCRIPTION WITHOUT DUMMIES

	count	mean	std	min	25%	50%	75%	max
Price	19137	10824.97	9109.47	1020.00	4300.00	670.00	16000.00	40000.00
Area	19137	102.97	71.58	6.00	51.00	74.00	140.00	902.00
Longitude	17676	121.457	0.115	120.458	121.391	121.455	121.531	121.926
Latitude	17676	31.214	0.122	30.720	31.179	31.218	31.238	31.825
Bedrooms	19137	2.34	1.41	0.0	1.0	2.0	3.0	9.0
Livingrooms	19137	1.43	0.62	0.0	1.0	1.0	2.0	6.0
Bathrooms	362	1.64	0.62	0.0	1.0	1.0	2.0	7.0
Floor	16236	16.35	11.11	1.0	6.0	17.0	27.0	121.0
NextToSubway	19137	0.56	0.50	0.0	0.0	1.0	1.0	1.0
Exquisite	11763	0.69	0.46	0.0	0.0	1.0	1.0	1.0
OpenForVisits	11763	0.19	0.39	0.0	0.0	0.0	0.0	1.0

REFERENCES

[1] Pengyu Zhang. “Internet false housing supply should be cured (Online China)”. In: (). URL: http://paper.people.com.cn/rmrbhwb/html/2019-07/12/content_1935598.htm.

[2] “Tesseract OCR Documentation”. In: (). URL: https://tesseract-ocr.github.io.

[3] “Baidu Map Open Platform”. In: (). URL: https://lbsyun.baidu.com.

[4] “H3: Uber’s Hexagonal Hierarchical Spatial Index”. In: (). URL: https://eng.uber.com/h3/.

[5] massquantity. “Kaggle - House Prices: Advanced Regression Techniques”. In: (). URL: https://www.kaggle.com/massquantity/all-you-need-is-pca-lb-0-11421-top-4.

[6] Jianping Liu. “Overview of scikit-learn Random Forest Class Library”. In: (). URL: https://www.cnblogs.com/pinard/p/6160412.html.

[7] Jianping Liu. “Overview of scikit-learn GBDT Class Library”. In: (). URL: https://www.cnblogs.com/pinard/p/6143927.html.

[8] Jianping Liu. “Overview of Main Parameters of SVM RBF”. In: (). URL: https://www.cnblogs.com/pinard/p/6126077.html.