INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(2), 235-250, April 2018.

Estimating Warehouse Rental Price using Machine Learning Techniques

Y. Ma, Z. Zhang, A. Ihler, B. Pan

Yixuan Ma
1. International Center for Informatics Research
Beijing Jiao Tong University, China
2. Department of Computer Science
University of California Irvine, USA
mayixuan@bjtu.edu.cn

Zhenji Zhang*
International Center for Informatics Research
Beijing Jiao Tong University
100044 No.3 Shangyuancun, Haidian, Beijing, China
*Corresponding author: zhjzhang@bjtu.edu.cn

Alexander Ihler
Department of Computer Science
University of California Irvine, USA
ihler@ics.uci.edu

Baoxiang Pan
Department of Civil and Environmental Engineering
University of California Irvine, USA
baoxianp@uci.edu

Abstract: Boosted by the growing logistics industry and digital transformation, the sharing warehouse market is undergoing a rapid development. Both supply and demand sides in the warehouse rental business are faced with market perturbations brought by unprecedented peer competitions and information transparency. A key question faced by the participants is how to price warehouses in the open market. To understand the pricing mechanism, we built a real world warehouse dataset using data collected from the classified advertisements websites. Based on the dataset, we applied machine learning techniques to relate warehouse price with its relevant features, such as warehouse size, location and nearby real estate price. Four candidate models are used here: Linear Regression, Regression Tree, Random Forest Regression and Gradient Boosting Regression Trees. The case study in the Beijing area shows that warehouse rent is closely related to its location and land price. Models considering multiple factors have better skill in estimating warehouse rent, compared to single-factor estimation. Additionally, tree models have better performance than the linear model, with the best model (Random Forest) achieving correlation coefficient of 0.57 in the test set. Deeper investigation of feature importance illustrates that distance from the city center plays the most important role in determining warehouse price in Beijing, followed by nearby real estate price and warehouse size.
Keywords: sharing warehousing, price estimation, machine learning.

1 Introduction

In today's logistics industry, insufficient capacity of self-owned warehouses and fluctuation in inventory have made sharing warehouses a useful option. However, more time and effort are needed for sharing warehouses to reach their full potential for supporting expanding supply chain businesses. In one 2015 survey [30], more than 75% of the participants were reported to suffer

significant inventory swings, with 31% of them faced with insufficient stock capacity and 26% with vacant storages. The paradox of both storage insufficiency and vacancy suggests a need for a smooth exchange of warehouse supply and demand information.

An ongoing digital marketplace transformation is taking place among various sections of the society. For instance, Airbnb, currently the largest online platform for people to lease or rent short-term lodgings, operates in 65,000 cities and had 31 billion of total valuation as of June 2017 [26]. Uber, the online driving service platform, had an average of 1 million rides of daily trips in 75 countries as of April 2017 [27].

Similar transformation happens in the shared warehousing as well. For example, Flexe, essentially the Airbnb of warehousing, provides a marketplace of spare storage space by gathering and releasing vacant warehouse information [29]. Craigslist, 58, and many other classified advertisements websites offer sections for people to post available warehouse messages.

In the platforms mentioned above, warehouse resources are usually archived into well-defined categories based on information offered by providers. Customers can make optimal decisions by filtering the listings with their demands, while platforms can offer customized integrated solutions when needed. This new business mode offers accessible information to both the supply and demand sides of the industry. It also encourages dedicated warehouses and self-hold vacant spaces to enter the shared warehouse market. A key question faced by the newcomers, as well as original warehouse providers, is how to price their warehouses in the open digital marketplace.

Theoretically, the optimal price is achieved from equilibrium in which the quantities of goods or services provided match the corresponding market's desire and ability to acquire the good or service. In an intransparent market, it's difficult, if not possible, for each participant to get the whole picture of the supply and demand information. This leads to unjustifiable pricing and inefficiency, which hurts the long-term prosperity of the market. Fortunately, such deficiency can be mitigated in the online marketplace, where information is more transparent to both sides of the transaction.

The availability of market information does not directly lead to optimal pricing. While an easy strategy is to take nearby shared warehouses as references to estimate a price, a more refined approach should estimate the price by considering an array of relevant factors, such as warehouse location, size, type, transportation, land price, etc. However, we lack a comprehensive empirical understanding of the pricing mechanism in this new emerging online marketplaces.

To bridge this gap, we apply the tool of Machine Learning to build warehouse rental pricing models. Such models can learn complex relations that are hidden within large amounts of real world data. They are especially helpful when our domain knowledge and understanding are limited. There are many previous works that use machine learning techniques for price estimation and prediction in various economic disciplines, such as stocks [9,17], electricity [4,23], online auctions [6,18], real estate [10,13,14], lodging [1,7,11,12,22], high speed trains [24,25], product lifecycles [19], and parking [16]. However, to our knowledge, this is the first work to gather real world data and analyse the pricing mechanism of the flourishing sharing warehouse market.

Our research develops a database consisting of real world sharing warehouse information from 58.com, one of the most popular classified advertisements website in China. The warehouse characteristics and its location information are selected as potential features in dominating the rental price. Four widely used machine learning algorithms, Linear Regression, Regression Tree, Random Forest Regression and Gradient Boosting Regression Trees, are applied to estimate warehouse pricing.

The rest of the paper is organized as follows. The data and materials are introduced in Section 2. We introduce the machine learning approaches in Section 3, followed by the results and discussion in Section 4. Finally, we summarize our conclusions.

Table 1: Shared warehouse data statistics

Attributes	**Latitude(°)**	**Longitude(°)**	**Size(m^2)**	**Rent (CNY/(m^2day))**
Mean	116.43	39.92	1,394.48	1.36
Std	0.16	0.14	2,311.61	1.26
Min	116.01	39.60	1.00	0.10
25%	116.31	39.84	101.75	0.60
50%	116.42	39.92	500.00	1.00
75%	116.56	40.00	1,500.00	1.50
Max	116.89	40.49	14,000.00	9.11

2 Data

Our major goal of this paper is to estimate sharing warehousing rental price using machine learning techniques. To explore this, we take the shared warehouse market in Beijing, one of the largest economic centers in China, as a case study. In this section, we describe the data set and relevant features for warehouse price estimation.

2.1 Data sets

We first create a dataset of sharing warehouses in the Beijing area. The data are acquired and organized based on warehouse rental posts on 58.com, one of the largest and most active general purpose classified websites in China. Posts during the period of November 2016 to January 2017 are archived using a web scraping framework in Python [21].

Fig. 1 gives an example of one rental listing. After removing incomplete and obviously erroneous data, in total we collected 2,462 rental listings, with each listing including information such as Title, District, Latitude, Longitude, Size and Rent across the Beijing area. Details on the data collection and cleaning processes can be found in Ma et al. [15]. The warehouse data statistics are shown in Table 1.

Figure 1: An example of Beijing Warehouse Rental Information Shown in Detailed Page of 58.com.

In addition to the warehouse pricing dataset, we also obtain 23,438 records of second-hand real estate rental information in the Beijing area in January 2017 from Lianjia.com, which is the website of one of the largest real estate agencies in China. Since a warehouse is a form of real estate, we assume that real estate rental prices are likely related to the shared warehouse rental price. The second hand real estate data statistics are described in Table 2.

Table 2: Second-hand Real Estate Data Statistics

Attributes	House Price (CNY/m^2)	Latitude(°)	Longitude(°)
Mean	59,542.34	116.44	39.94
Std	25,562.39	0.17	0.11
Min	3,949.00	115.9	39.62
25%	40,517.50	116.33	39.87
50%	54,299.50	116.41	39.94
75%	74,014.50	116.51	40.01
Max	149,963.00	116.95	40.47

2.2 Feature selection

To estimate the warehouse rental price, we extracted five features from the data sets, including Distance from City Center, Size, House Price (nearby second hand real estate price), Distance from the Closest House and District. A detailed explanation of these feature is given in the sequel.

We first consider the warehouse size and location, and present a general picture of their connection to rental price in Fig. 2. Each dot represents the location of a sharing warehouse, with the size indicating warehouse size and color indicating rental price. For the location-rent relationship, from Fig. 2, we can see that warehouses closer to the city center tend to have higher prices. For the size-rent relationship, we see that larger warehouses tend to have cheaper prices. Thus, we include Distance from City Center and Size as features in our warehouse rent estimation.

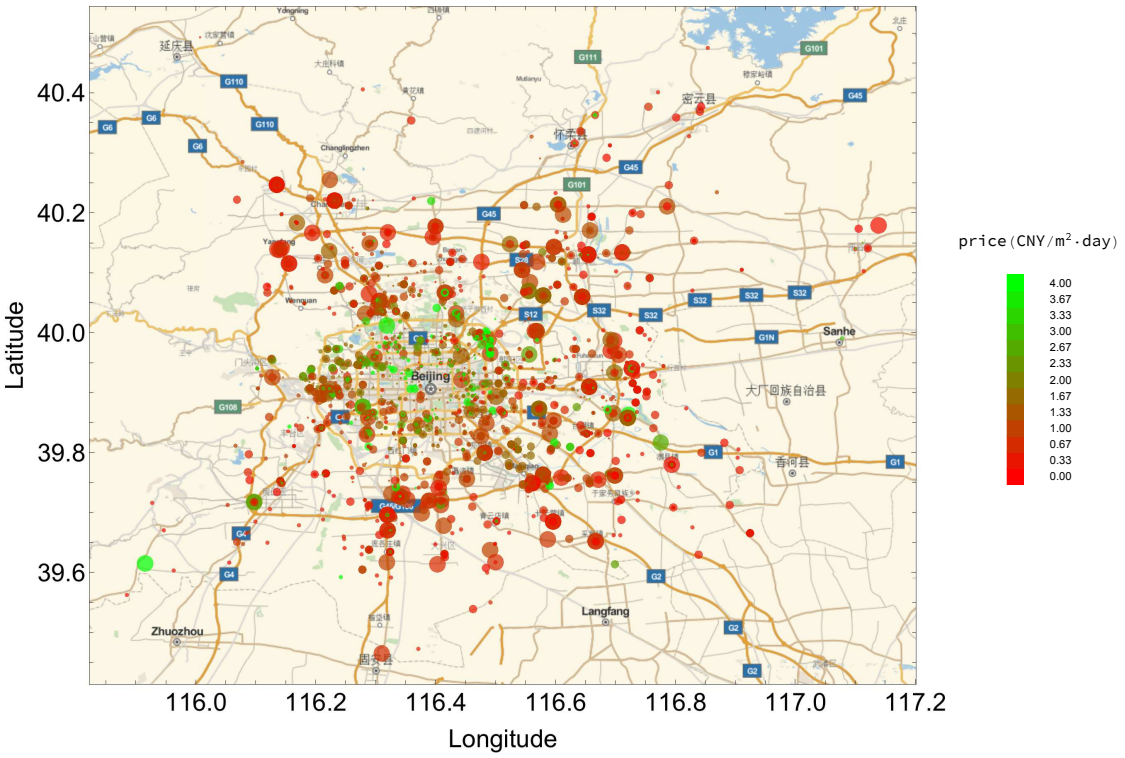

Figure 2: The Distribution of Shared Warehouse Listings in Beijing.

In addition, local real estate price can reflect the market value of the property. Since a warehouse is a kind of real estate, we assume there is a positive relationship between warehouse rent and house prices in the vicinity, with a stronger and more direct relationship when the two locations are closer. Fig. 3 shows the coordinates of warehouses and second-hand real estate. In this work, we computed the market price of the nearest second-hand house to the warehouse (House Price) and the distance between these two locations (Distance from the Closest House).

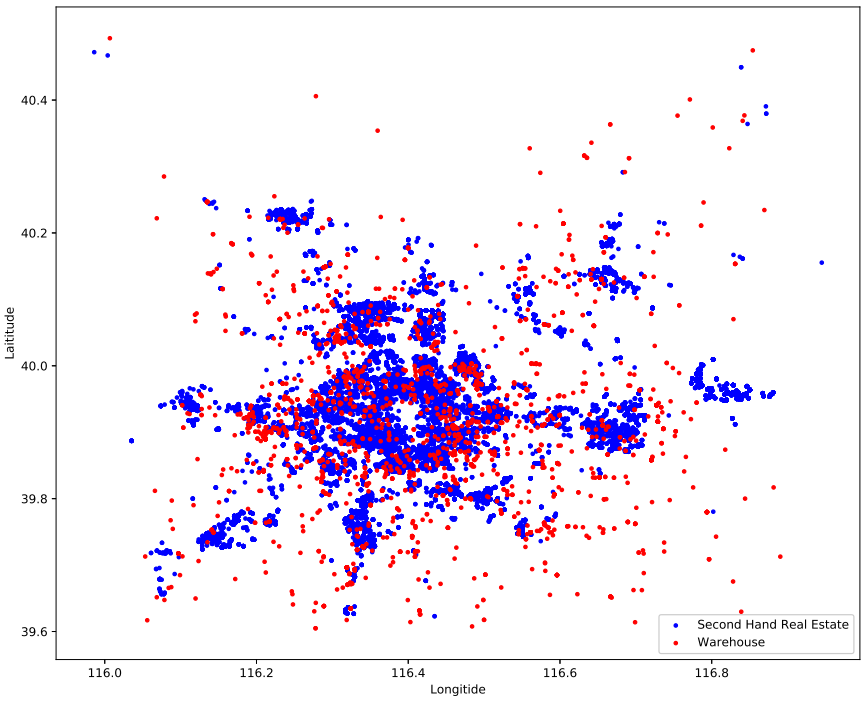

Figure 3: Coordinates of second hand real estate and shared warehouse in Beijing.

Finally, the location's District, a type of administrative division that is managed by local government, plays an important role in warehouse rental pricing. For example, as shown in Fig. 4, the warehouse rent is much more expensive in the Dongcheng and Xicheng districts than in other districts, and the lowest warehouse rent appears in the Miyun district. Thus, we select District as one of the features for estimating warehouse rent. Since it is a categorical feature, we convert it to a vector of fourteen binary indicator variables before modeling. A summary of the features is listed in Table 3, and an analysis of the relationship between several individual features and the warehouse rent is discussed in Section 4.1.

3 Methodology

In this section, we explain the machine learning techniques used in our experiment, which include:

- Linear Regression

Table 3: Feature statistics

Feature Name	**Mean**	**Standard Deviation**	**Definition**
Rent	1.36	1.26	Listed warehouse rent in online warehouse marketplace (Measured in CNY).
House Price	58,014.64	23,791.27	Listed price of nearest second hand house (Measured in CNY).
Distance from the Closest House	0.75	1.27	The distance between the location of a listed rental and the location of the nearest second hand house, computed as spherical distance with latitude and longitude (Measured in km).
Size	1,394.48	2,311.61	Listed warehouse size in online warehouse marketplace (Measured in m^2).
Distance from the City Center	17.17	10.35	The distance between the location of a listed rental and the city center, computed as spherical distance with latitude and longitude (Measured in km).
District Area 1	0.08	0.28	District Area: Changping. (Dummy variable)
District Area 2	0.23	0.42	District Area: Chaoyang. (Dummy variable)
District Area 3	0.13	0.33	District Area: Daxing. (Dummy variable)
District Area 4	0.01	0.11	District Area: Dongcheng. (Dummy variable)
District Area 5	0.02	0.15	District Area: Fangshan. (Dummy variable)
District Area 6	0.09	0.29	District Area: Fengtai. (Dummy variable)
District Area 7	0.15	0.36	District Area: Haidian. (Dummy variable)
District Area 8	0.00	0.06	District Area: Huairou. (Dummy variable)
District Area 9	0.00	0.04	District Area: Mengtougou. (Dummy variable)
District Area 10	0.00	0.05	District Area: Miyun. (Dummy variable)
District Area 11	0.04	0.21	District Area: Shijingshan. (Dummy variable)
District Area 12	0.06	0.23	District Area: Shunyi. (Dummy variable)
District Area 13	0.14	0.35	District Area: Tongzhou. (Dummy variable)
District Area 14	0.08	0.28	District Area: Xicheng. (Dummy variable)

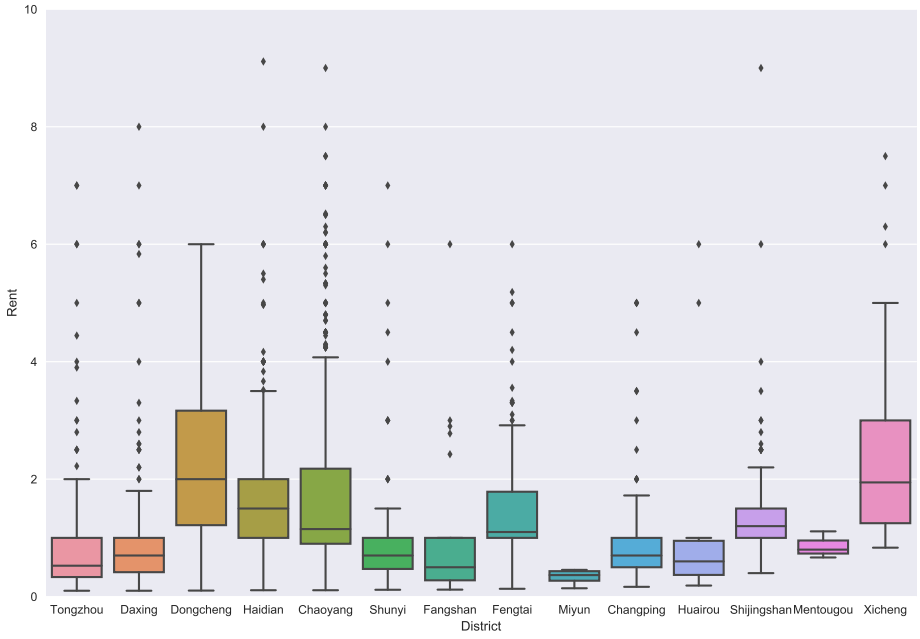

Figure 4: A boxplot showing the price of shared warehousing in Beijing, grouped by District.

- Regression Tree
- Random Forest Regression
- Gradient Boosting Regression Trees

A simple linear regression model constructs an optimized linear combination of the predictors (features). The relatively strict assumption of a linear relationship makes it difficult to fit on many real world problems, in which many factors may work together in a non-linear way to determine the predictand. The latter three models belongs to a broader family of tree-based learning algorithms. Tree algorithms are well suited for problems where features interact in nonlinear manners and no monad global model fits the predictor-predictand relation well. The basic idea behind these algorithms is to recursively partition the feature space until each cell of the partition is well modeled by a simple predictor (such as a constant value) [8]. Next we provide a more detailed explanation of the three tree-based algorithms.

3.1 Regression Tree

The regression tree, a variant of decision trees, is a popular method designed to approximate real-valued functions. Regression trees have a number of advantages. For one, they can be used without requiring feature pre-processing or normalization, since each feature is processed independently. Regression trees also tend to work well with data sets that have a mixture of continuous and categorical variables, and with features that are on very different scales, both of which are present in our research.

Basically, regression trees learn a series of explicit if/then rules based on feature values to predict the target value. Unlike linear regression, which holds a single predictive formula over

the entire feature space, regression trees break the global model into many local models using this recursive partitioning process, and then performs multiple (very simple) local regressions. There are two main parts to building a regression tree: growing the trees and pruning the trees.

The basic Regression Tree growing algorithm is as follows:

1. Start from a root node, containing all data points. Calculate the prediction for the leaf (typically, the mean value of the target values for the training points in that leaf), and the sum of squared errors for the leaf.
2. Search over all possible splits of all variables for the one which results in the smallest sum of square errors for its two leaves. If the sum of square errors is less than some user-defined threshold, or one of the resulting nodes contains less than a user-defined number of points, stop splitting. Otherwise, continue splitting data and creating new nodes.
3. For each new node, go to Step 1.

Once the tree's induction process is finished, pruning can be applied to improve the tree's generalization capacity by reducing its structural complexity. The number of cases in nodes can be used as the pruning criteria.

In the Regression Tree construction process, pre-pruning is used to avoid growing an overly complex tree. Allowing the tree to grow unpruned can result in overfitting, since the model may fit noise in the data. We control the maximum tree structural complexity using a max-depth parameter and a minimum number of samples per leaf. Max depth controls the maximum depth of the tree, limiting the total number of split points any decision can have. The min samples per leaf parameter is a threshold that controls how many data instances must be present in a leaf in order to consider splitting it further. To obtain robust and generalizable models in the experiment, many possible choices of these control parameters were assessed, ranging over depths from 1 to 50 and minimum sample sizes from 1 to 50. Via cross-validation, we found the best results at maximum depth 12 and minimum leaf samples set to 13.

3.2 Random Forest Regression

The Random Forest Regression algorithm, proposed by Breiman [3], is an ensemble method using decision trees as base learners. It combines many different individual trees into an ensemble, and introduces random variation while building each decision tree. The term "random" in random forest has a dual meaning: a subset of the data is randomly chosen to establish each tree, and features are randomly selected in each split test as well. The resulting ensemble of trees is averaged to produce an overall prediction, which reduces overfitting while allowing for complex individual learners.

Random Forests have several good characteristics, including high accuracy among current algorithms, efficiency on large data sets, the ability to handle high dimensional input variables and missing data. Thus, Random Forest Regression is widely used and often yields very good results on a variety of problems [20].

The Random Forest Regression algorithm is as follows:

1. Pick n_{tree} bootstrap samples from the dataset;
2. Build an unpruned regression tree based on each of bootstrap samples. When picking the best split for a node, a random subset of features is selected to be searched over, rather than finding the best split across all possible features;
3. Predict new data using the mean of n_{tree} trees' predictions.

Random Forest models can compute an internal unbiased estimate of the generalization error, called out-of-bag (or OOB). Given enough trees, the OOB estimate of error rate can be quite accurate.

The output of Random Forest Regression depends primarily on two parameters: maximum number of features and number of estimators. The maximum number of features parameter is the number of predictors chosen randomly at each tree node. It greatly influences the diversity of the random trees in the forest. When the parameter is low, the trees become more complex and diverse. However, if the parameter is high (e.g., close to the total number of features), the trees in the forest will tend to be very similar. A typical default setting of max features for regression is the log base two of the total number of features. The number of estimators parameter represents the number of trees in the ensemble. Ensembles reduce overfitting by averaging over more trees, but this increases computation. Increasing the number of trees tends out to be a better solution [2]. After testing, we chose the log base two of the total number of features (which is 4 in the experiment) and 3000 total trees.

3.3 Gradient Boosting Regression Tree

Gradient Boosting Regression Tree is another tree based ensemble method widely used in real word applications. It often yields excellent off-the-shelf accuracy on many problems. Unlike Random Forest Regression, which builds a forest of different trees in parallel, Gradient Boosting Regression Tree establishes a sequence of trees, each of which is intended to correct the mistakes of the previous trees in the series.

The basic idea of boosting is to convert weak learners to strong ones. Starting from a weak learner, we do iterations, where at each iteration, we change the weight distribution of the data and apply a new weak learner to the weighted data. This builds a sequence of different weak learners, and by combining those weak learners, we get a final, strong learner. When our model makes a large error (high cost) on a given data point, we assign a greater weight to that data point in next iteration, and assign less weight on the points for which the cost is less. If the cost of a weak learner is large, we give it a smaller contribution when combining the learners, and give greater emphasis on the learners for which the cost is less.

The Gradient Boosting Regression Tree algorithm for a loss or cost function $L\left(y^{(i)}, f\left(x\right)\right)$ is as follows [5]:

1. Initialize the model:

$$f_0\left(x\right) = \text{argmin}_c \sum_{i=1}^{M} L\left(y^{(i)}, c\right)$$

2. Train K models iteratively $k = 1, 2, \ldots, \text{M}$:

   (a) Compute the residual: for $i = 1, 2, \ldots, \text{M}$:

$$r_{ki} = -\left[\frac{\partial L\left(y^{(i)}, f\left(x^i\right)\right)}{\partial f\left(x^i\right)}\right] f\left(x\right) = f_{k-1}\left(x\right)$$

   (b) Train a regressor for the residual and obtain the leaf nodes $R_{kj}, j = 1, 2, \ldots, J$

   (c) For $j = 1$ to J, calculate:

$$c_{ki} = \text{argmin}_c \sum_{x_{i \in R_{kj}}} L\left(y^i, f_{k-1}\left(x_i\right) + c\right)$$

(d) Update the model:

$$f_k(x) = f_{k-1}(x) + \sum_{j=1}^{J} c_{kj} I(x \in R_{kj})$$

3. Get the final strong learner:

$$f_s(x) = f_K(x) = \sum_{k=1}^{K} \sum_{j=1}^{J} c_{kj} I(x \in R_{kj})$$

In Gradient Boosting Tree Regression model, the number of estimators and the learning rate are important parameters. The first one controls the number of trees included in the ensemble. The second parameter works as a step size, scaling the contribution of each individual tree in reducing the loss. The effects of these two parameters interact: a smaller learning rate can often give better performance, but at the cost of requiring more trees in the ensemble (and thus more computation as well). Via cross-validation we find the best results by choosing ensemble size 4,000 with learning rate set to 0.01.

3.4 Model evaluation

For all the models, we take 70% of the data as a training set and use rest of the data as a test set. Two common score metrics are used for evaluating model performance: Root Mean Square Error (RMSE) and correlation coefficient (r). The "best" model is the one with the lowest RMSE and the highest r.

$$RMSE = \frac{\sqrt{\sum_{i=1}^{m}(e_i - o_i)}}{m}$$

$$r = \frac{\Sigma(e_i - \bar{e})(o_i - \bar{o})}{\sqrt{\Sigma(e_i - \bar{e})^2 \Sigma(o_i - \bar{o})^2}}$$

where e is the model estimation, m is sample size, and o is the observation.

4 Results and discussion

In this section, we use each of the four modeling strategies to predict warehouse rental price. We demonstrate the performance of each model, and illustrate the importance of the various features.

4.1 Model performance

Before presenting the model performance, we show the correlation between rent and selected features in Fig.5. Rent and Distance from City Center are negatively correlated. Warehouses that are closer to the city center have higher rent variance. Considering the Size feature, 70% warehouses have sizes below $1000m^2$. In this range, size alone has little impact on Rent. For warehouses that are larger than $5000m^2$, rents are relatively small and stable. We also see that House Price is a good indicator for land price. Among all the features, House Price has the greatest impact on Rent($r = 0.34$). All three factors can affect the warehouse rent; however, the combination of them provides more information on predicting warehouse renal price. Below we illustrate model results considering all the features we described in Section 2.2.

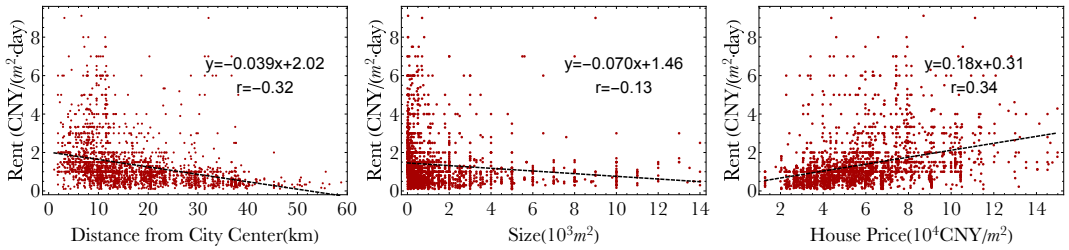

Figure 5: The relationship between rent and each selected feature. The fitted linear relation and its correlation coefficient are indicated in each sub-figure.

Fig. 6 shows the scatter plot of estimated and true warehouse rental prices of the four candidate models. Here we only show the test data performance. Generally, all models show a certain skill in estimating warehouse rental price. The tree based models have obviously higher skill compared to the Linear Regression Model. This reveals the nonlinear relationship between the factors and the rental price. Additionally, models for simulating lower rental price have better performance. When it comes to higher rental price, models tend to underestimate the rents.

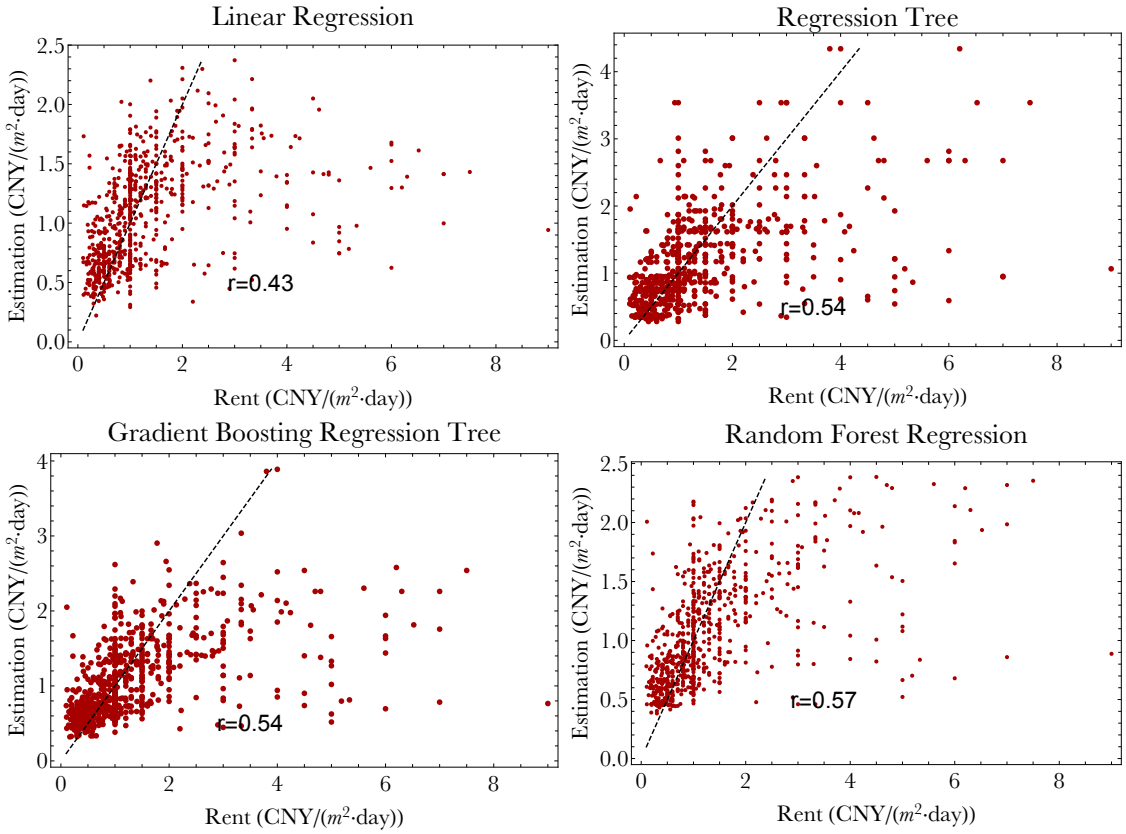

Figure 6: The relationship between the estimation and the target value for different models on the test set.

Table 4 shows results of RMSE and correlation coefficients for different models. Three tree based algorithms have smaller RMSE value and larger r value both on training set and test set, compared to Linear Regression. Within three tree models, the results of RMSE and r are similar. This indicates that the tree-based models are more capable of capturing data interconnection and estimating warehouse price, compared to Linear Regression. Of the four models, the Regression

Table 4: RMSE and correlation coefficient among different models

Model	Training Set		Test Set	
	RMSE (CNY/m^2· day)	r	RMSE (CNY/m^2· day)	r
Linear Regression	1.31	0.43	1.25	0.44
Regression Tree	1.02	0.63	1.05	0.54
Random Forest Regression	1.04	0.60	1.06	0.57
Gradient Boosting Regression Tree	1.05	0.68	1.07	0.54

Table 5: Error among different models on the test set

Model	Min	25%	Medium	75%	Max	Mean	Std
Linear Regression	-1.30	-0.27	-0.04	0.42	8.05	0.32	1.11
Regression Tree	-2.60	-0.28	0.03	0.38	7.93	0.22	1.03
Gradient Boosting Regression Tree	-1.94	-0.23	0.03	0.39	8.23	0.28	1.04
Random Forest Regression	-1.90	-0.24	0.00	0.40	8.11	0.28	1.03

Tree model has the smallest RMSE (1.05 CNY/m^2· day)), and Random Forest Regression has the highest correlation coefficient (0.57) on the test set.

Table 4 shows the error statistics for different models on test set. Linear Regression has the highest error mean and standard deviation and Regression Tree Model has smallest error mean and standard deviation.

To further analyze the error, we draw the error distributions of four models in different observation intervals on the test set in Fig.7. There is high relevance between the observed value and the model error. Large target values tend to have higher error mean and variance, no matter what model is used. These results could be related to data quality or the pricing mechanism. High prices may be boosted to aid subsequent bargaining, or list a fake price rather than representing the true warehouse value. Alternatively, warehouses with higher rent may be priced based on different criteria compared to lower rent warehouses.

4.2 Feature importance

The features predicted to be important in the model help us understand what features are driving the rent and what features are deemed important for each of the model. Feature importance is typically a number between 0 and 1 assigned to an individual feature. A feature importance of zero means the feature is not used at all in the prediction. A feature importance of one means the feature perfectly predicts the target.

For regression tree, the importance of a feature is simply the total reduction of the cost in sum of squares achieved by all splits based on that feature [3]. However, in random forest regression, there are two common measures of feature importance. The first is based on mean squared error (MSE), and the second is prediction accuracy of the out-of-bag portion of the data after permuting each feature. The difference between the two MSEs are then averaged over all trees and normalized by the standard error. The second measure is calculated on the training data used to grow the trees. Here we use the latter method as our measure of importance.

Table 6 shows the results of feature importance carried out by three tree algorithms. We can capture the consistency of feature importance in three models: Distance from City Center is the most influential feature, followed by Size and House Price. These features were chosen at an early level of the tree, comparing to the others. The District feature has the lowest importance value in all the models, which suggests that District has the smallest impact on Rent prediction.

The three models are slightly different in their feature importance order. Both Regression

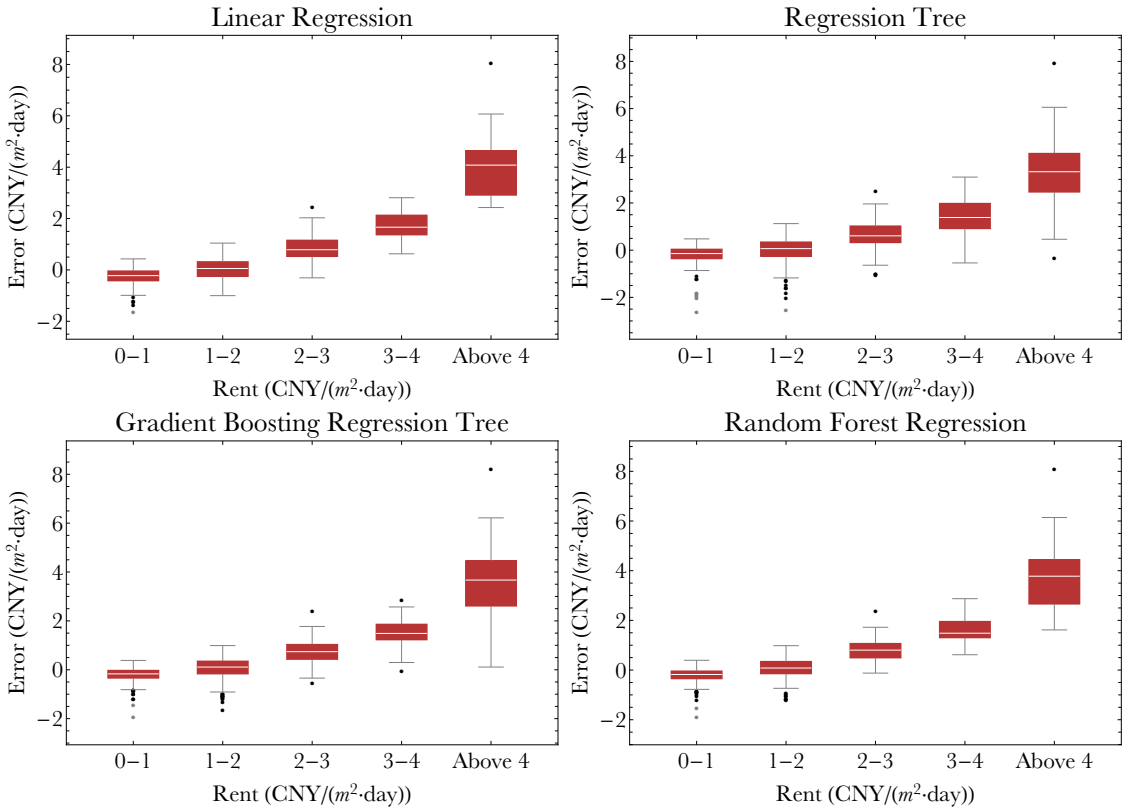

Figure 7: The relationship between error and target value of different models on the test set.

Tree and Random Forest Regression Model take House Price as the second most important feature, while it is Size in the Gradient Boosting Regression Tree.

5 Conclusion

The rental warehouse market is undergoing a transformation boosted by digital platforms. Both the supply and demand sides are faced with information transparency, which creates fluctuations in the pricing. To better understand the pricing mechanism in this evolving market and guide the pricing process for users, we collected and analyzed real-world rental warehouse data from the classified advertisement website.

We estimated the rental price of shared warehouses using five features selected from the data set: Distance from City Center, House Price, Distance from the Closest House, Size and District. Regression Tree, Random Forest Regression and Gradient Boosting Regression Tree along with Linear Regression were compared for predicting price. We used RMSE and correlation coefficient to provide model performance measurements. Results show that single factors exert considerable influence on rent pricing, while models considering multiple factors have more predictive skill. Tree based models provide better performance than Linear Regression, and the Random Forest method gave the best performance.

Feature importance analysis showed that the location, measured by distance from city center, plays the most important role in determining warehouse rental price, followed by local land price or warehouse size.

Estimation error is further analyzed conditioned on the warehouse rent. We found that high rent warehouses often correspond to higher error levels in the models. Thus, high rent warehouses

Table 6: Feature importance in tree based models

Feature Name	**Importance (Regression Tree)**	**Importance (Random Forest Regression)**	**Importance (Gradient Boosting Regression Tree)**
Distance from City Center	**0.434205**	**0.273986**	**0.287716**
Size	0.173837	0.158182	**0.217792**
House Price	**0.207980**	**0.233020**	**0.207745**
Distance from the Closest House	0.112518	0.164432	0.164071
District Area 1	0.002581	0.011886	0.007634
District Area 2	0.009909	0.026253	0.009557
District Area 3	0.005362	0.025396	0.013583
District Area 4	0.000000	0.000730	0.009991
District Area 5	0.011750	0.009544	0.021440
District Area 6	0.000218	0.007067	0.003513
District Area 7	0.011849	0.025661	0.019505
District Area 8	0.003037	0.000000	0.000460
District Area 9	0.000000	0.000000	0.000000
District Area 10	0.000000	0.000000	0.000000
District Area 11	0.016273	0.011826	0.020358
District Area 12	0.002113	0.002895	0.000325
District Area 13	0.006511	0.046599	0.009163
District Area 14	0.001858	0.002523	0.007147

may require more factors to be considered in estimating their price. However, the high prices might also be due to price boosting or fake posts.

Despite the various factors and models used here, there are still considerable errors in estimating the warehouse rents. On the one hand, more features from the real world should be considered, such as text information that reveal social or psychological factors influencing warehouse pricing. On the other hand, it may also be true that in the growing free market, not all prices are justified; the data are intrinsically noisy. Our methodology here provides an attempt to extract useful information from the complicated real world market.

Acknowledgment

This work was supported by the National Natural Science Foundation of China(Grant No.71390334), Beijing Program of Philosophical and Social Science (Grant No.B16HZ00060) and Junior Fellowship for CAST Advanced Innovation Think-tank Program (Grant No.B16M00150).

Bibliography

[1] Antipov, E. A.; Elena, B. P. (2012); Mass appraisal of residential apartments: An application of Random forest for valuation and a CART-based approach for model diagnostics, *Expert Systems with Applications*, 39(2), 1772–1778, 2012.

[2] Breiman, L. (1996); Bagging predictors, *Machine Learning*, 24(2), 123–140, 1996.

[3] Breiman, L. (2001); Random forests, *Machine Learning*, 45(1), 5–32, 2001.

[4] Chen, X.; Dong, Z. Y.; Meng, K.; Xu, Y.; Wong, K. P.; Ngan, H. W. (2012); Electricity price forecasting with extreme learning machine and bootstrapping, *IEEE Transactions on Power Systems*, 27(4), 2055–2062, 2012.

[5] Elith, J.; Leathwick, J. R.; Hastie T. (2008); A working guide to boosted regression trees, *Journal of Animal Ecology*, 77(4), 802–813, 2008.

[6] Ghani, R. (2005); Price prediction and insurance for online auctions, *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 411–418, 2005.

[7] Gutt, D.; Herrmann, P. (2015); Sharing means caring? Hosts' price reaction to rating visibility *ECIS*, 2015.

[8] Hastie, T.; Tibshirani, R.; Friedman, J. (2001); The Elements of Statistical Learning. Springer, 2001.

[9] Kim, K. J. (2003); Financial time series forecasting using support vector machines, *Neurocomputing* 55(1), 307–319, 2003.

[10] Kusan, H.; Osman A.; Ilker O. (2010); The use of fuzzy logic in predicting house selling price, *Expert systems with Applications*, 37(3), 1808–1813, 2010.

[11] Li, J.; Moreno, A.; Zhang, D. J. (2015); Agent behavior in the sharing economy: Evidence from Airbnb, *Ross School of Business Working Paper Series*, 1298, 2015.

[12] Li, Y.; Wang, S.; Yang, Pan, Q.; Tang, J. (2017); Price recommendation on vacation rental websites, *Proceedings of the 2017 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics*, 399–407, 2017.

[13] Limsombunchai, V. (2004); House price prediction: hedonic price model vs. artificial neural network, *New Zealand Agricultural and Resource Economics Society Conference*, 25–26, 2004.

[14] Liu, J. G; Zhang, X. L.; Wu, W. P. (2006); Application of fuzzy neural network for real estate prediction, *International Symposium on Neural Networks*, 1187–1191, 2006.

[15] Ma, Y. X.; Zhang, Z. J.; Pan, B. X. (2017); Reveal status quo of Beijing warehouse in open market, *Logistics, Informatics and Service Sciences (LISS), 2017 International Conference on. IEEE*, 2011–2017, 2017.

[16] Maric, M.; Gracanin, D.; Zogovic, N.; Ruskic, N.; Ivanovic, B. (2017); Parking search optimization in urban area, *International Journal of Simulation Modelling*,16(2), 2017.

[17] Patel, J.; Shah, S.; Thakkar, P.; Kotecha, K. (2015); Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques, *Expert Systems with Applications*, 42(1), 259–268, 2015.

[18] Raykhel, I.; Ventura, D. (2008); Real-time automatic price prediction for eBay online trading, *IAAI*, 2009.

[19] Rehar, T.; Ogrizek, B.; Leber, M.; Pisnik, A.; Buchmeister, B. (2017); Product lifecycle forecasting using system's indicators, *International Journal of Simulation Modelling*,16(1), 2017.

[20] Segal, M. R. (2004); Machine learning benchmarks and random forest regression, *Center for Bioinformatics Molecular Biostatistics*, 2004.

[21] Scrapy Community. (2015); *Scrapy: A Fast and Powerful Scraping and Web Crawling Framework*, http:// scrapy.org/doc/.

[22] Wang, D.; Nicolau, J. L. (2017); Price determinants of sharing economy based accommodation rental: A study of listings from 33 cities on Airbnb. com, *International Journal of Hospitality Management*, 62, 120–131, 2017.

[23] Yamin, H. Y.; Shahidehpour, S. M.; Li, Z. Y. (2004); Adaptive short-term electricity price forecasting using artificial neural networks in the restructured power markets, *International journal of electrical power & energy systems*, 26(8), 571–581, 2004.

[24] Zhang, D. (2017); High-speed Train Control System Big Data Analysis Based on Fuzzy RDF Model and Uncertain Reasoning, *International Journal of Computers, Communications & Control*, 12(4), 2017.

[25] Zhang, D.; Sui, J.; Gong, Y. (2017); Large scale software test data generation based on collective constraint and weighted combination method, *Technical Gazette*, 24(4), 1041–1049, 2017.

[26] [Online]. Available: www.expandedramblings.com/index.php/airbnb-statistics/, Accesed on 10 September 2017.

[27] [Online]. Available: www.expandedramblings.com/index.php/uber-statistics/, Accesed on 10 September 2017.

[28] [Online]. Available: www.dhl.com/, Accesed on 10 September 2017.

[29] [Online]. Available: www.flexe.com/, Accesed on 10 September 2017.

[30] [Online]. Available: www.flexe.com/Flexe-capacity-eco/, Accesed on 10 September 2017.