Economics and Management
ISSN: 2367-7600; 1312-594X
Volume: XVII, Issue: 1, Year: 2020, pp. 27-42

MACHINE LEARNING ALGORITHMS FOR FORECASTING ASSET PRICES: AN APPLICATION TO THE HOUSING MARKET

Anton A. Gerunov[1]

Received: 12,01.2020 Accepted: 30.01.2020

Abstract

This article investigates the application of advanced machine learning algorithms for forecasting housing prices. To this end we leverage a dataset of 414 observations of housing deals in Taipei and model it with both traditional econometric and novel machine learning algorithms. An exhaustive search among 107 alternative methods is conducted and their forecast accuracy is reported in detail. Using the root mean squared error (RMSE) as a benchmark metric, we find that implementations of the random forest family have superior performance, far surpassing that of more traditional approaches such as the multiple linear regression. The results have utility for both academics and practitioners and can be easily transferred to other forecasting problems in economics and business.

Keywords: *asset prices, real estate, forecasting algorithms, machine learning*

JEL Codes: *C52, C53, R31*

1. Introduction

Modern organizations perform a wide array of complex activities in their regular operations. In addition to the usual business processes of production, delivery and support, many organizations also have activities related to the acquisition and management of tangible fixed assets, including real estate. The pricing of this estate is necessary to estimate for the purposes of accounting, financial planning, and, most importantly, strategic management. Under standard practice, this is often done with the help of a dedicated expert appraiser who combines objective market data with subjective judgment and adjustments to arrive at a final assessment. The main problem with this approach is that it is largely dependent on human judgment, which makes it relatively expensive, slow and difficult to scale. These factors result in infrequent or even one-off property valuations, although a dynamic market environment often implies significant dynamics.

[1] Ph.D. and Associate Professor at the Faculty of Economics and Business Administration, Sofia University “St. Kliment Ohridski”, 125 Tsarigrasko Shosse Blvd., Sofia 1113

In the age of digital transformation this no longer needs to be the case. Modern data storage and processing technology, combined with powerful machine learning algorithms allow organizations to automate the process of real-time valuation of a given property, which can enable its rational management. This article focuses on the best statistical and machine learning methods to forecast housing prices. It aims to review and compare a wide set of popular forecasting algorithms and outline which provide the most accurate results for housing prices. This task is of interest to both organizations involved in the sale and purchase of real estate (construction companies, brokers, etc.), as well as organizations involved in the financing and securitization of transactions (banks, non-bank credit institutions, funds, etc.).

The article is structured as follows: section two presents a compact literature review of popular statistical methods and their application to forecasting housing prices; section three introduces the data set under study and reports its descriptive statistics. Section four applies five of the most common algorithms for modeling the data at and, while section five conducts an extensive search for the best forecasting algorithm among a set of 106 methods. Finally, section six concludes.

2. Literature Review

Prediction of housing prices is a highly relevant business problem that has also attracted significant attention from researchers. A standard model for this valuation is the hedonic pricing approach, whereby prices are determined by both intrinsic characteristics of the real estate, as well as characteristics of its environment (Sirmans et al, 2005). Among the former we can include the size and type of the property, the number of rooms, different features, overall condition, year of construction, location and many others. Among the latter are the characteristics of the neighborhood, transport connectivity, number of shops, crime rates, different amenities and possibly numerous other factors. Essentially, the hedonic pricing approach measures what the consumer is willing to pay given the presence of a set of characteristics (ibid.). All those can be summarized as explanatory (independent) variables, and the resulting housing prices – as a target (dependent) variable.

Alternatively, one can apply a relatively theory-free set of algorithms to estimate precise pricing, given a known structure of the dataset. Many researchers are exploring methods such as artificial neural network and comparing them to the more traditional approaches leveraging hedonic pricing (see e.g. Ghorbani & Afgheh, 2017), or using pure methods from machine learning such as neural networks (Lim et al., 2016) or support vector machines (Chen et al., 2017). While some authors use more than one method for prediction (e.g. Yeh & Hsu, 2018), they still leverage a somewhat limit set of algorithms that are not guaranteed to produce the best results. Therefore, we outline the need for an exercise to create an exhaustive testing of (almost) all relevant regression algorithms and identify the ones with best forecasting performance.

The most popular types of algorithms for such a task include both tradition econometric tools such the multiple linear regression as well as a set of more advanced machine learning ones such the neural network, the support vector machines, the k-Nearest Neighbor algorithm, and the decision trees and random forest approaches. Here, we shortly review the less well-known approaches and point the interested reader to Hastie et al. (2009) for a more detailed description.

Neural networks are computational algorithms whose structure is strongly influenced by the way the human brain functions. It consists of neurons that send activating impulses to each other, this system forming a biological neural network. In a statistical neural network, the architecture of the algorithm is similar, with neurons playing different variables and values, and activation performed by a predetermined mathematical function. It calculates and transmits the various values within the model. The explanatory dependent variables form the input layer of the neural network. Each of these variables influences the estimation of the final target variable by a series of weighted functions, called a non-linear weighted sum (sum). In short, the input layer transmits activation pulses, calculated according to a particular activation function K, to the first intermediate layer. It, in turn, uses these impulses as input to its activation functions to the next layer, and so to the last one that defines the final target variable. For more details on the statistical features and characteristics of neural networks, we refer the reader to Ripley & Hjort (1996).

The **k-nearest-neighbors (kNN)** algorithm has a long history of various classification applications, which is due both to its relative simplicity and to the relatively good results it produces. The basic idea behind it is that it is a classification algorithm that uses the already known classes, which are located close enough to a given observation to determine the class of the observation itself. For more details on the calculation and statistical properties of this algorithm, we refer the reader to the work of Peterson (2009) and Hastie et al. (2009). We emphasize that although it is a compact and relatively intuitive algorithm, it can be successful in a number of situations, and its results are particularly good in cases with irregular boundaries between different classes of data or in which each class has a number clearly differentiated prototypes.

Decision trees are an alternative model for modeling tasks related to recognizing different classes. Using an array of test data, the trees select the best classifier among a set of explanatory variables, with the process flowing iteratively. Initially, at the first node, the algorithm selects the variable that best distinguishes the classes from one another and selects its optimal value for classification. The task is then branched to the value of this variable and the new nodes re-selected the optimal variable and its value, resulting in new branching. When a decision is made, the graphical presentation of the results looks very much like an inverted tree, where its name comes from. Their main problem is that they are over-adjusting to the data they are trained on (so-called overfitting) and that the algorithm can be misled by local optimists. As a result, excellent predictive power can be obtained within the data examined and much lower - on another set. To solve the major problems of decision trees such as high variation and over-adjusting, we can combine them into an ensemble model. By collecting a certain number of decision trees, we can combine them into a common ensemble model - the so-called **random forest** (randomized regression and classification forest). When training this model, initially random samples of data and their characteristics are selected and a set of trees is grown based on them. These are then combined into a single model whereby the output value is determined by the weighted values obtained from each tree. For further information and more details on this approach, we direct the reader to the seminal article by Breiman et al. (2001) on the subject.

Support vector machines are classification models that originate in the field of machine learning (Cortes & Vapnik, 1995). For given classes or sets of observations, they seek to find the optimal classification by calculating the optimal hyperlinearity in the middle of the largest distance between the closest points of the different classes. The boundary points in this space are called support vectors and hence the very name of this family of algorithms. Essentially, the parameters of the this algorithm are evaluated by solving quadratic programming problems. More sophisticated machines with support vectors can design data with finite number of dimensions on higher dimensional planes and classify these planes. For more in-depth discussion, statistical features, and other variants of machines with support vectors, we direct the reader to Hastie et al. (2009).

By leveraging a large number of implementations of these popular types of machine learning algorithms to, as well as adding a set of novel one, this paper aims to further elaborate and significantly expand previous work in forecasting housing prices (Park & Bae, 2015), and thus to bring insight on the optimal modeling strategy for both academics and practitioners.

3. Data, Samples, and Descriptive Statistics

The key business problem we solve is the need to determine and regularly update the correct price of a real estate so that the organization can evaluate the effectiveness of potential disposals with it (purchase, sale, letting, etc.), as well as to predict the future price dynamics in order to minimize the risk of unexpected losses due to adverse trends in the property market. For this task, we use data provided by Yeh & Hsu (2018), with which the authors test an evaluation algorithm they propose – the so-called. a comparative quantitative approach. They (ibid.) compare this new approach with four other alternatives - two approaches for hedonic pricing, a multiple linear regression and a neural network, and find that it leads to better predictive results.

The database itself consists of 414 observations of real estate transactions in Taipei (Taiwan) against seven different characteristics - date, years since the construction of the building, distance to the metro station, number of nearby shops, geographical coordinates (latitude and longitude), unit price area. For convenience of modeling, we divide the date into two components - the year and the order of the transaction within the calendar year (a combination of day and month). The target variable is the price per unit of area and is a continuous numeric variable. For the purposes of estimating the models, we divide the data into two subsets. The training set consists of 80% of the original data and is used for initial estimation of the model, while the testing set (remaining 20%) is used to obtain out-of-sample accuracy metrics as per best practice. This ensures that the models are tested on a different data that they are trained on, and thus mitigates the problem of overfitting and increases the reliability of the reported accuracy numbers.

The main statistics of the data under consideration are presented in Table 1. The average age of traded properties is 17.7 years, and we observe significant differences in this variable. The high standard deviation indicates that there are bot many new and many old buildings. Similarly, the the distance to the nearest metro station is 1084 m. on average, but with a very high standard deviation. There are 4 stores on average around the property, and the average cost per unit area (target variable) is 379,800 new Taiwan dollars per 1 ping. In addition, the data include the exact coordinates of the properties, which give an indication of the neighborhood in which they are located. Additionally, we have information about the time dimension of the transactions - year and sequence (day and month), and these variables allow to take into account the dynamic trend in property prices.

Table no. 1 – Descriptive Statistics for Dataset on Housing Prices, N = 414

Variable	**Name in data base**	**Mean**	**Std. devi.**	**Median**	**Min.**	**Max.**	**Skew ness**	**Kurto-sis**
Years since construction	age	17.71	11.39	16.10	0.00	43.80	0.38	-0.89
Distance to metro station, m	distance	1083.8	1262.1	492.23	23.38	6488.0	1.88	3.13
Number of nearby shops	stores	4.09	2.95	4.00	0.00	10.00	0.15	-1.08
Latitude, coord.	lat	24.97	0.01	24.97	24.93	25.01	-0.44	0.24
Longitude, coord.	long	121.53	0.02	121.54	121.47	121.57	-1.21	1.15
Price per unit area	y	37.98	13.61	38.45	7.60	117.50	0.60	2.11
Year of transaction	year	2012.7	0.46	2013	2012.0	2013	-0.85	-1.29
Transaction order	seq	348.38	275.03	333.00	0.25	667.00	-0.07	-1.55

Source: Author calculations based on data by Yeh & Hsu (2018)

The correlation matrix in Figure 1 shows the relationships between the variables under consideration. The price per unit area is very strongly and negatively related to the distance from the metro station, which is a logical and expected result. In addition, there is a negative correlation between the price and the age of the building in which the property is located. We see a well-expected positive correlation with the number of stores as well as the geographical coordinates of the site. Given the positive association between the coordinates and the number of shops, we can conclude that clusters of preferred properties (neighborhoods) are noticeable, which are very close to shopping malls and high-cost residential buildings. The weak positive correlation between the year of the transaction and the price indicates a certain process of rising property prices over time, which should also be taken into account in their modeling.

Figure no. 1 Correlation matrix of data under study

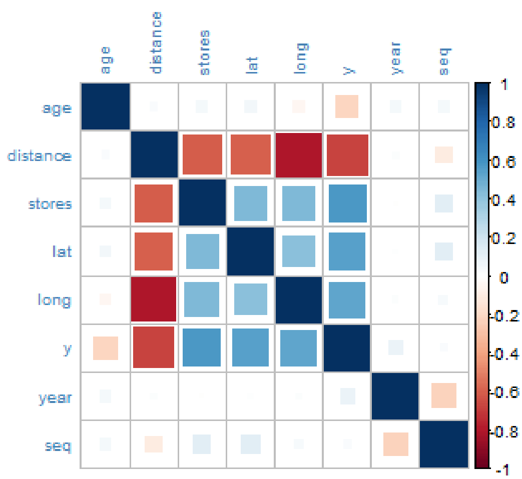
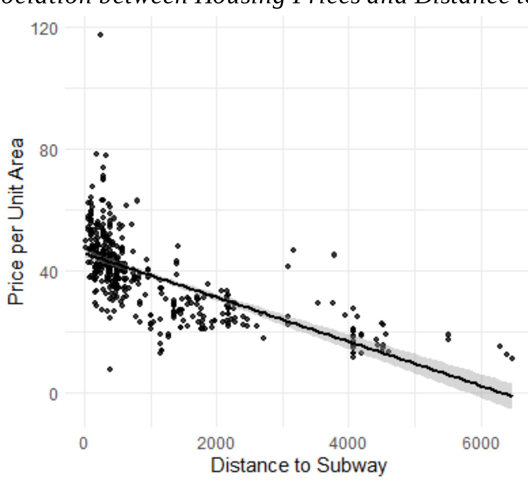


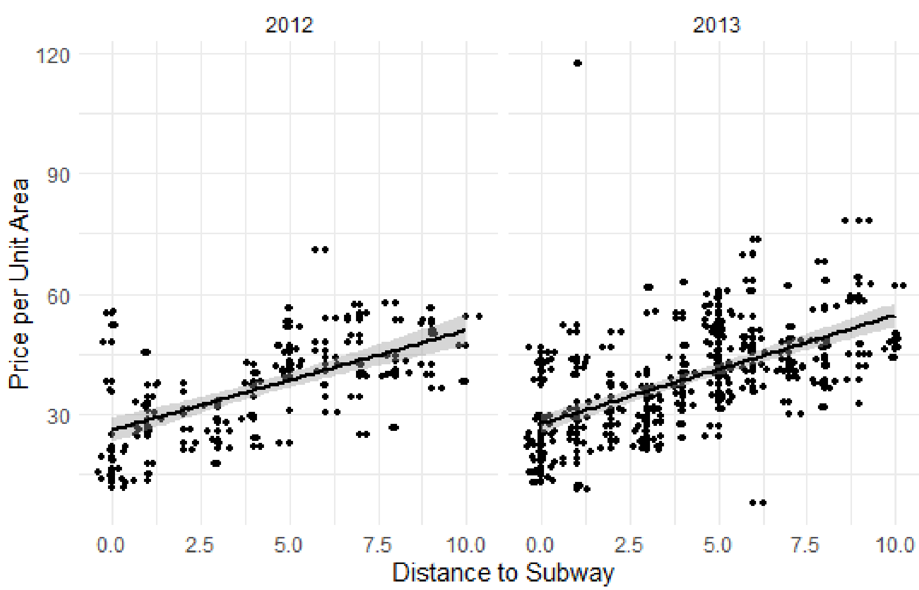
Figure no. 2 Association between Housing Prices and Distance to Subway



The dot plot shown in Figure 2 shows in an even clearer way the relationship between the distance from the transport points (subway) and the price of the property, this relationship being almost linear. However, this should be interpreted with some caution, since the vast majority of the areas in the analyzed database are within one kilometer (1,000 m) of a metro station.

Figure 3 shows the relationship between the price per unit area and the number of shops in the vicinity. We observe a strong positive relationship between the two, and it is valid for the two years for which we have data on transactions. We emphasize that these data allow the training of a forecast model for the price of residential properties, as they include typical features of interest in the formation of prices in this segment - neighborhood (via coordinates), transport connectivity, availability of shops, age of buildings.

Figure no. 3 Association between Housing Prices and Number of Shops in Vicinity across Transaction Years



4. Comparison of Common Approaches

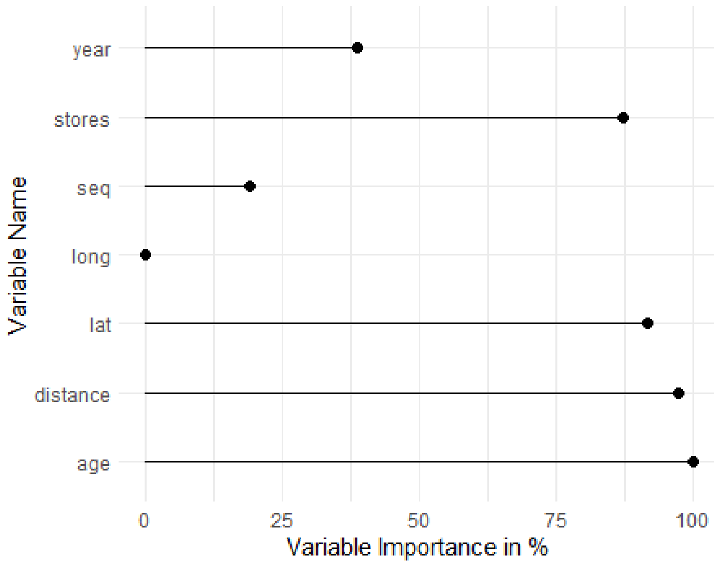
The classical approach for modeling linear relationships in econometrics is by using multiple linear regression. The results of this model are presented in Table 2. All variables considered reach statistical significance at least at the 5% level, with the exception of the longitude and the year of the transaction. The year of construction has a strong negative effect, with each year leading to a reduction in the unit cost of 2,740 new Taiwan dollars. This effect is significant at p <0.005 levels. Surprisingly, although the distance to the metro station reaches statistical significance, the coefficient is negative, which is an unexpected result.

Table no. 2 – Coefficients of Linear Regression on Housing Price Data

Variable	**Variable name in model**	**Coefficient**	**Standard error**	**z-statistic**	**Significance**
Constant	(Intercept)	-5699.739	7337.776	-0.777	0.438
Years since construction	age	-0.274	0.046	-6.005	0.000
Distance to metro station, m	distance	-0.005	0.001	-5.326	0.000
Number of nearby shops	stores	1.208	0.224	5.403	0.000
Latitude, coord.	lat	236.201	51.986	4.544	0.000
Longitude, coord.	long	-32.238	56.126	-0.574	0.566
Year of transaction	year	1.870	1.103	1.695	0.091
Transaction order	seq	-0.004	0.002	-2.193	0.029
		R^2	**0.570**	**Adjusted R^2**	**0.562**

The number of stores within a close radius is clearly one of the most important drives of the unit price. From a statistical point of view, it reaches significance at levels well below 1%, and the high positive coefficient also shows its high practical significance. Increasing the number of nearby stores by 1 increases the price per unit area by 1,208 new Thai dollars. Latitude, but not longitude, is significant, reflecting the direction of development of the city (north-south versus east-west). The last indicator - the order of the transaction - also reaches significance, with an extremely slight fall in prices over the year. The effect is rather small and of little interest in the practical management process. The relative importance of the variables is also shown graphically in Figure 4. Here the importance of the distance to the subway, the number of shops, and the geographical location of the property can be clearly seen. It should be borne in mind that, unlike the regression coefficients, the relative importance of the variables does not allow one to examine the direction of the effect, but only the contribution to the quality of the model.

Figure no. 4 Relative Variable Importance in a Linear Multiple Regression Model



Among the most popular regression algorithms in the field of machine learning that are applicable to this task are the neural network, the kNN algorithm, the random forest, and the support vector machine (for another application see e.g. Gerunov, 2019). Their predictive accuracy is compared with that of the linear regression in Table 3.

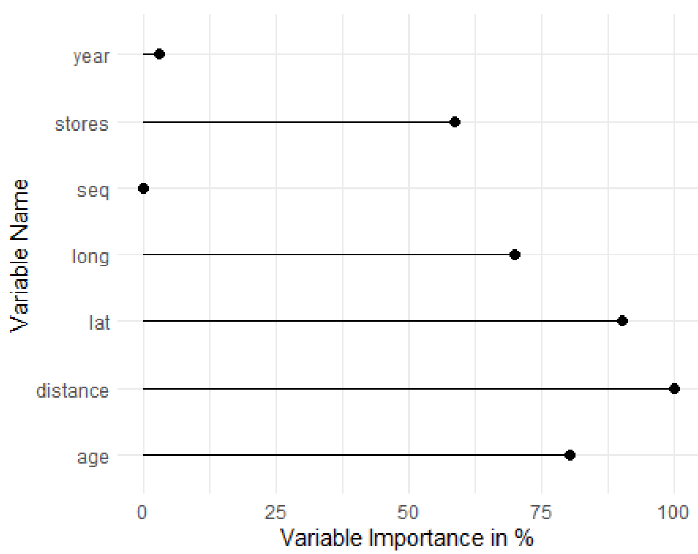
Table no. 3 – Descriptive Statistics for Dataset on Housing Prices, N = 414

Method	**Mean Error, Me**	**Root Mean Squared Error, RMSE**	**Mean Absolute Error**	**Mean Percentage Error, MPE**	**Mean Absolute Percentage Error, MAPE**
Multiple Linear Regression	0.478	8.015	6.065	-2.097	17.004
Neural Network	37.135	39.416	37.135	96.970	96.970
kNN	-0.859	9.521	7.233	-6.509	19.859
Random Forest	-0.768	6.519	4.852	-4.437	13.411
Support Vector Machine	1.864	8.188	6.029	2.008	16.175

Looking at the root mean square error (RMSE) of the forecast, we find that the neural network has by far the lowest performance with RMSE = 39.4, followed by the kNN algorithm (RMSE = 9.5). The linear regression and the support vector machine prove to be good alternatives with almost similar RMSE values. By far the best model is that of a random forest with RMSE = 6.5 and mean absolute error rate MAPE = 13.4. The average random forest error is -0.77, which shows some underestimation of the realized values. From a practical point of view, the ability to generate area unit estimates with an average RMSE error of only $ 6,519 at an average value of $ 39,780 per unit area represents a significant and meaningful improvement.

The relative importance of the variables in the random forest model is presented in Figure 5, again underlining the importance of distance from a metro station, geographical coordinates and the number of shops in the vicinity. The low importance of the year and the sequence of the transaction reflect the limited influence of the time dynamics from year to year, as well as the lack of a clear seasonality trend in prices.

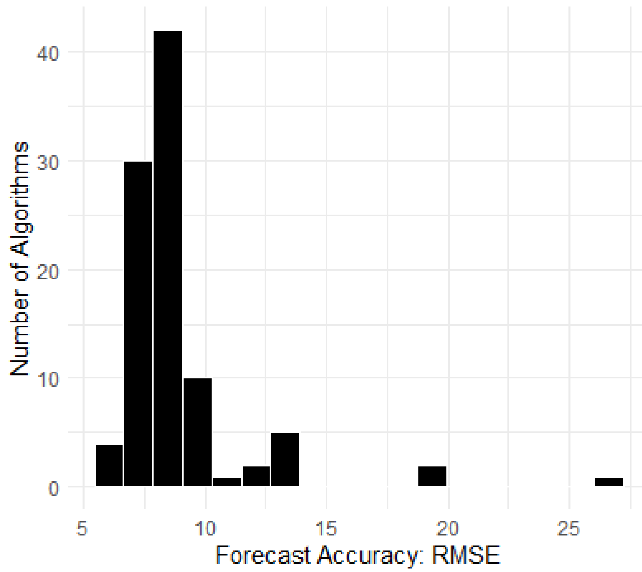
Figure no. 5 Relative Variable Importance in a Random Forest Model



5. Extensive Search for Optimal Algorithms

Using the housing price data, we estimate 106 alternative machine learning models and investigate their predictive accuracy. The full list of algoritym s is presented in the Appendix. A histogram of their predictive accuracy with respect to the root of the root mean square errors is presented in Figure 6. The vast majority of methods have register a RMSE in the range of 7 to about 9. The best algorithms among the tested ones have a predictive accuracy of RMSE < 6.5. and those with the worst results can reach an RMSE value of over 25. It is notable that while most algorithms tend to neatly cluster together, there are some extremely poor performers. On the other hand, a few algorithms have markedly better performance than average, thus showing that there is potential business value in selecting and using the best ones.

Figure no. 6 Forecasting Accuracy of Regression Methods under Study



The ten top performers in terms of lowest root mean square error are presented in Table 4. It is immediately apparent that seven of them are different implementations of the random forest family. They all exhibit extremely good predictive accuracy, with their errors in the range of RMSE = 6.46 to RMSE = 7.10. The other three non-random forest algorithms are 2 based on a kernel function and one based on a Gaussian process, whose predictive accuracy is about RMSE = 7.10. Here we also calculate a complexity measure that takes into account the calculation time needed. The most time-intensive algorithm is standardized to a 100%, and the time spent by other is presented as a fraction of that. The best method - that of a regularized random forest is only 35 p. p. faster than the slowest in the sample. On the other hand, the second best - the quantile random forest - is nearly 20 times faster than the most resource-intensive, and the difference in predictive accuracy between the two is almost imperceptible.

Table no. 4 – Forecasting Accuracy of Top 10 Best Performing Methods

Type of Algorithm	**Method**	**Mean Error, Me**	**Root Mean Sqrd. Error, RMSE**	**Mean Abs. Error**	**Complexity Measure**
Regularized Random Forest	RRF	-0.750	6.459	4.831	64.8%
Quantile Random Forest	qrf	0.001	6.470	4.695	5.2%
Regularized Random	RRFglobal	-0.832	6.568	4.890	9.9%

Forest					
Random Forest	ranger	-0.878	6.600	4.884	7.0%
Parallel Random Forest	parRF	-0.936	6.689	4.965	3.8%
Random Forest	ranger	-0.943	6.689	4.908	4.0%
Radial Basis Function Kernel Regularized Least Squares	krlsRadial	-0.435	7.068	5.388	14.8%
Bayesian Additive Regression Trees	bartMachin e	-0.795	7.076	5.353	11.1%
Random Forest by Randomization	extraTrees	-0.948	7.081	5.137	7.8%
Gaussian Process with Polynomial Kernel	gaussprPol y	-0.501	7.082	5.443	2.1%

This underlines that with this type of task, it is possible to find the optimal point between the benefits and the costs of calculating a given algorithm. Moreover, a fast calculation speed of the algorithm also indicates the possibility of switching from asynchronous to synchronous operations, i.e. from model calculation and subsequent use and future updates to real-time analytics, which is used and trained simultaneously. Thus, algorithms that both a high accuracy and a low complexity value are prime candidates for practical applications in the analytic pipeline.

6. Recommendations and Conclusion

This article aims to outline how novel and advanced machine learning methods can be successfully applied to traditional forecasting problems in economics and business. The particular application under study in forecasting the prices in Taipei's housing market, leveraging the data provided by Yeh & Hsu (2018). We applied both traditional econometric methods, as well as machine learning algorithms such as implementations of neural networks, kNN-type algorithms, Bayesian methods, decision trees and random forests, support vector machines and a host of more exotic approaches. Overall, 107 different statistical algorithms are tested. The main results are clear – the forecasting accuracy of machine learning approaches significantly outperforms that of more traditional econometric tools such as the linear regression. Random forests, in particular, display the best forecasting performance, reflected in their low root mean squared errors.

This leads to a number of recommendations that can inform both the forecasting theory and practice. In terms of research, the application of machine learning algorithms to relevant problems in economics and business seems a fruitful venue for further work. These methods can be leveraged to solve a large number of regression and classification type of problems and are characterized by the fact that they scale well to the constantly increasing amount of available data (so-called big data). Application fields include financial and business forecasting, demand planning, risk management, credit scoring, customer segmentation, recommendation engines, and others. From a methodological point of view it would be useful to further investigate the performance and stability of those algorithms in particular tasks of interest.

From a practical standpoint, these results can be applied directly for the benefit of businesses with activities in the real estate, facility management, or financing. Real-time price forecasting and re-evaluation effectively provides valuable information that can be fed in organizational decision loops. Apart from improving the internal decision-making, this can also be used for regulatory purposes, especially by financial institutions with large portfolios in real estate. Whatever the sphere of application, the results obtained aim to improve the understanding of how to implement novel machine learning methods to forecasting and thus enable modern organizations to take a further step along the path to comprehensive digital transformation.

REFERENCES

Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Chen, J. H., Ong, C. F., Zheng, L., & Hsu, S. C. (2017). Forecasting spatial dynamics of the housing market using Support Vector Machine. International Journal of Strategic Property Management, 21(3), 273-283.

Cortes, C. & Vapnik, V. (1995). Support-vector network. Machine Learning, 20, 1-25.

Gerunov, A. (2019). Modelling economic choice under radical uncertainty: machine learning approaches. *International Journal of Business Intelligence and Data Mining*, 14(1/2), 238-253.

Ghorbani, S., & Afgheh, S. M. (2017). Forecasting the House Price for Ahvaz City: the Comparison of the Hedonic and Artificial Neural Network Models. Journal of Urban Economics and Management, 5(19), 29-44.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

Lim, W. T., Wang, L., Wang, Y., & Chang, Q. (2016, August). Housing price prediction using neural networks. In 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD) (pp. 518-522). IEEE.

Park, B., & Bae, J. K. (2015). Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. Expert Systems with Applications, 42(6), 2928-2934.

Peterson, L. E. (2009). K-nearest neighbor. Scholarpedia, 4(2), 1883.

Ripley, B. D., & Hjort, N. L. (1996). Pattern recognition and neural networks. Cambridge university press.

Sirmans, S., Macpherson, D., & Zietz, E. (2005). The composition of hedonic pricing models. Journal of real estate literature, 13(1), 1-44.

Yeh, I. C., & Hsu, T. K. (2018). Building real estate valuation models with comparative approach through case-based reasoning. Applied Soft Computing, 65, 260-271.

APPENDIX

Table no. A1 – Set of Algorithms used for Analysis and Testing

#	Algorithm	#	Algorithm
1	Model Averaged Neural Network	55	Multi-Layer Perceptron
2	Bagged MARS	56	Multi-Layer Perceptron, multiple layers
3	Bagged MARS using gCV Pruning	57	Monotone Multi-Layer Perceptron Neural Network
4	Bayesian Additive Regression Trees	58	Multi-Step Adaptive MCP-Net
5	Bayesian Generalized Linear Model	59	Neural Network
6	Boosted Tree	60	Neural Network
7	The Bayesian lasso	61	Non-Negative Least Squares
8	Bayesian Ridge Regression (Model Averaged)	62	Tree-Based Ensembles
9	Bayesian Ridge Regression	63	Non-Informative Model
10	Bayesian Regularized Neural Networks	64	Parallel Random Forest
11	Boosted Linear Model	65	Neural Networks with Feature Extraction
12	Boosted Tree	66	Principal Component Analysis
13	Conditional Inference Random Forest	67	Penalized Linear Regression
14	Conditional Inference Tree	68	Partial Least Squares
15	Conditional Inference Tree	69	Partial Least Squares Generalized Linear Models
16	Cubist	70	Projection Pursuit Regression
17	Stacked AutoEncoder Deep Neural Network	71	Quantile Random Forest
18	Multivariate Adaptive Regression Spline	72	Quantile Regression Neural Network
19	Elasticnet	73	Ensembles of Generalized Linear Models
20	Tree Models from Genetic Algorithms	74	Random Forest
21	Random Forest by Randomization	75	Radial Basis Function Network
22	Ridge Regression with Variable Selection	76	Relaxed Lasso
23	Generalized Additive Model using LOESS	77	Random Forest
24	Generalized Additive Model using Splines	78	Random Forest Rule-Based Model
25	Gaussian Process	79	Ridge Regression
26	Gaussian Process with Polynomial Kernel	80	Robust Linear Model
27	Gaussian Process with Radial Basis Function Kernel	81	Classification and Regression Trees, CART, ver. 1
28	Stochastic Gradient Boosting	82	Classification and Regression Trees, CART, ver. 2
29	Multivariate Adaptive Regression	83	Classification and Regression Trees,

	Splines		CART, ver. 3
30	Fuzzy Rules via MOGUL	84	Quantile Regression with LASSO penalty
31	Generalized Linear Model	85	Non-Convex Penalized Quantile Regression
32	Negative Binomial Generalized Linear Model	86	Regularized Random Forest
33	Boosted Generalized Linear Model	87	Regularized Random Forest
34	glmnet	88	Relevance Vector Machines with Radial Basis Function Kernel
35	Generalized Linear Model with Stepwise Feature Selection	89	Subtractive Clustering and Fuzzy c-Means Rules
36	Hybrid Neural Fuzzy Inference System	90	Partial Least Squares
37	Independent Component Regression	91	Spike and Slab Regression
38	Partial Least Squares	92	Sparse Partial Least Squares
39	k-Nearest Neighbors	93	Supervised Principal Component Analysis
40	k-Nearest Neighbors	94	Support Vector Machines with Linear Kernel
41	Polynomial Kernel Regularized Least Squares	95	Support Vector Machines with Linear Kernel
42	Radial Basis Function Kernel Regularized Least Squares	96	L2 Regularized Support Vector Machine (dual) with Linear Kernel
43	Least Angle Regression	97	Support Vector Machines with Polynomial Kernel
44	Least Angle Regression	98	Support Vector Machines with Radial Basis Function Kernel
45	The lasso	99	Support Vector Machines with Radial Basis Function Kernel
46	Linear Regression with Backwards Selection	100	Support Vector Machines with Radial Basis Function Kernel
47	Linear Regression with Forward Selection	101	Bagged CART
48	Linear Regression with Stepwise Selection	102	Partial Least Squares
49	Linear Regression	103	Wang and Mendel Fuzzy Rules
50	Linear Regression with Stepwise Selection	104	eXtreme Gradient Boosting
51	Model Tree	105	eXtreme Gradient Boosting
52	Model Rules	106	eXtreme Gradient Boosting
53	Multi-Layer Perceptron	107	Self-Organizing Maps
54	Multi-Layer Perceptron, with multiple layers		